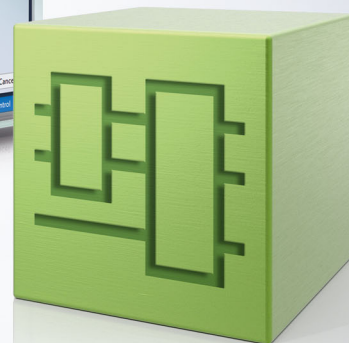
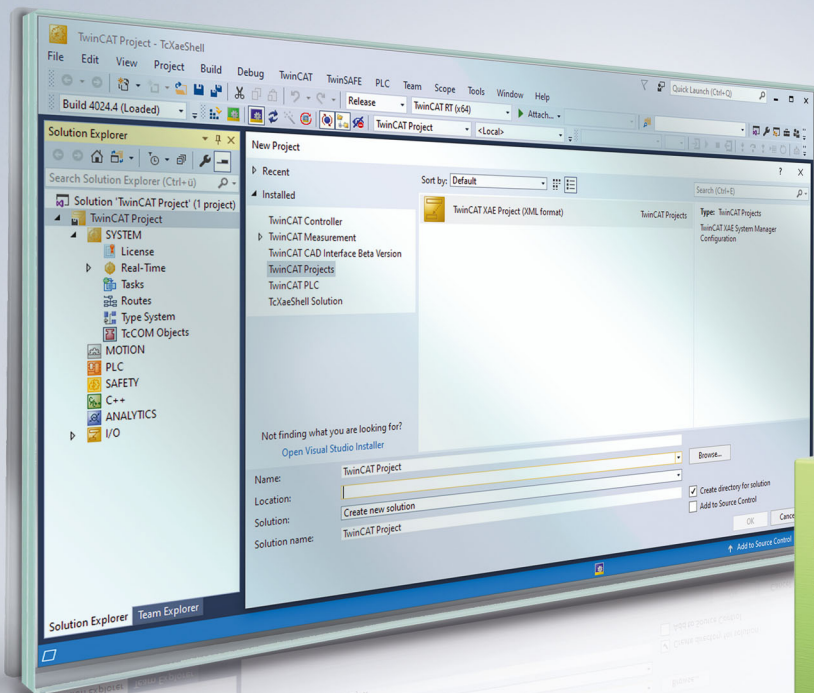


# BECKHOFF New Automation Technology

Manual | EN

# TE1000

TwinCAT 3 | PLC Library: Tc3\_DALI





# Table of contents

<b>1</b>	<b>Foreword</b> .....	<b>5</b>
1.1	Notes on the documentation .....	5
1.2	For your safety .....	5
1.3	Notes on information security.....	7
<b>2</b>	<b>Introduction</b> .....	<b>8</b>
<b>3</b>	<b>DALI</b> .....	<b>9</b>
3.1	IEC 62386 .....	9
3.2	Communication .....	10
3.3	Priorities .....	11
3.4	Bus Timing .....	12
3.4.1	Structure data frame .....	12
3.4.2	Transmission length .....	14
3.4.3	Collision detection .....	14
3.4.4	Cycle times PLC tasks .....	15
3.4.5	Summary.....	16
3.5	Memory banks.....	16
<b>4</b>	<b>Programming</b> .....	<b>21</b>
4.1	POUs.....	21
4.1.1	Applications.....	21
4.1.2	Commands.....	54
4.1.3	Communication .....	578
4.1.4	Devices .....	589
4.1.5	Events .....	841
4.1.6	Helper.....	845
4.1.7	Simulation .....	846
4.2	DUTs .....	852
4.2.1	Structures.....	852
4.2.2	Enumerations .....	854
4.3	GVLs .....	863
4.3.1	Constants .....	863
4.3.2	Parameter .....	864
4.4	Integration into TwinCAT.....	865
4.4.1	EL6821 with CX5120 .....	865
4.4.2	KL6821 with CX5120 .....	868
4.5	Runtime messages .....	873
<b>5</b>	<b>Appendix</b> .....	<b>879</b>
5.1	Commissioning and diagnosis .....	879
5.1.1	Commissioning KL6821 .....	879
5.1.2	Commissioning EL6821 .....	880
5.1.3	DALI PLC Commissioning Tool.....	882
5.1.4	Event Logger.....	885
5.1.5	Error evaluation.....	887
5.2	Use of properties.....	890

---

5.3	Transfer of the reference to the communication block .....	892
5.4	Tc2_DALI and Tc3_DALI in the same project.....	893
5.5	Creating an application function block .....	894
5.6	Creating a device function block .....	897



# 1 Foreword

## 1.1 Notes on the documentation

This description is intended exclusively for trained specialists in control and automation technology who are familiar with the applicable national standards.

For installation and commissioning of the components, it is absolutely necessary to observe the documentation and the following notes and explanations.

The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described satisfies all requirements for safety, including all the relevant laws, regulations, guidelines, and standards.

### Disclaimer

The documentation has been prepared with care. The products described are, however, constantly under development.

We reserve the right to revise and change the documentation at any time and without notice.

No claims to modify products that have already been supplied may be made on the basis of the data, diagrams, and descriptions in this documentation.

### Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered and licensed trademarks of Beckhoff Automation GmbH.

If third parties make use of designations or trademarks used in this publication for their own purposes, this could infringe upon the rights of the owners of the said designations.

### Patents

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:

EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702  
and similar applications and registrations in several other countries.

## EtherCAT®

EtherCAT® is registered trademark and patented technology, licensed by Beckhoff Automation GmbH, Germany

### Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The distribution and reproduction of this document as well as the use and communication of its contents without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event that a patent, utility model, or design are registered.

## 1.2 For your safety

### Safety regulations

Read the following explanations for your safety.

Always observe and follow product-specific safety instructions, which you may find at the appropriate places in this document.

**Exclusion of liability**

All the components are supplied in particular hardware and software configurations which are appropriate for the application. Modifications to hardware or software configurations other than those described in the documentation are not permitted, and nullify the liability of Beckhoff Automation GmbH & Co. KG.

**Personnel qualification**

This description is only intended for trained specialists in control, automation, and drive technology who are familiar with the applicable national standards.

**Signal words**

The signal words used in the documentation are classified below. In order to prevent injury and damage to persons and property, read and follow the safety and warning notices.

**Personal injury warnings****⚠ DANGER**

Hazard with high risk of death or serious injury.

**⚠ WARNING**

Hazard with medium risk of death or serious injury.

**⚠ CAUTION**

There is a low-risk hazard that could result in medium or minor injury.

**Warning of damage to property or environment****NOTICE**

The environment, equipment, or data may be damaged.

**Information on handling the product**

This information includes, for example:  
recommendations for action, assistance or further information on the product.

## 1.3 Notes on information security

The products of Beckhoff Automation GmbH & Co. KG (Beckhoff), insofar as they can be accessed online, are equipped with security functions that support the secure operation of plants, systems, machines and networks. Despite the security functions, the creation, implementation and constant updating of a holistic security concept for the operation are necessary to protect the respective plant, system, machine and networks against cyber threats. The products sold by Beckhoff are only part of the overall security concept. The customer is responsible for preventing unauthorized access by third parties to its equipment, systems, machines and networks. The latter should be connected to the corporate network or the Internet only if appropriate protective measures have been set up.

In addition, the recommendations from Beckhoff regarding appropriate protective measures should be observed. Further information regarding information security and industrial security can be found in our <https://www.beckhoff.com/secguide>.

Beckhoff products and solutions undergo continuous further development. This also applies to security functions. In light of this continuous further development, Beckhoff expressly recommends that the products are kept up to date at all times and that updates are installed for the products once they have been made available. Using outdated or unsupported product versions can increase the risk of cyber threats.

To stay informed about information security for Beckhoff products, subscribe to the RSS feed at <https://www.beckhoff.com/secinfo>.

## 2 Introduction

The user of this library requires basic knowledge of the following:

- TwinCAT XAE
- Design and characteristics of Beckhoff IPCs and their bus terminal system
- DALI technology and mode of operation
- Relevant safety regulations for building technical equipment

This software library is intended for building automation system partners of Beckhoff Automation GmbH & Co. KG. The system partners operate in the field of building automation and are concerned with the installation, commissioning, expansion, maintenance and service of measurement, control and regulating systems for the technical equipment of buildings.

The Tc3\_DALI library is usable on all hardware platforms that support TwinCAT 3.1.4022.20 or higher. The EL6821 is supported from the Tc3\_DALI library V3.16.1.0.

### 3 DALI

DALI (Digital Addressable Lighting Interface) is a definition for the standardization of digital interfaces between control gears (lamps) and control devices (sensors and application controllers). The standard (IEC 62386) allows the manufacturers of lighting components to implement complex lighting tasks easily and conveniently.

The KL6811 (DALI version-1/DSI) and KL6821/EL6821 (DALI-2) Bus Terminals are integrated into the bus terminal system and are therefore fieldbus-independent. The DALI data is forwarded to the DALI devices via the respective bus coupler. Bus controllers also offer the option of running PLC programs locally in IEC 61131-3.

To ensure the interoperability of DALI-2 devices with each other, the DALI Alliance (DiiA) provides a certification program. Products that have successfully completed the DALI-2 certification process may use the DALI-2 logo.



All certified DALI-2 devices are entered in the DiiA product database. The product database can be accessed via the DiiA homepage:

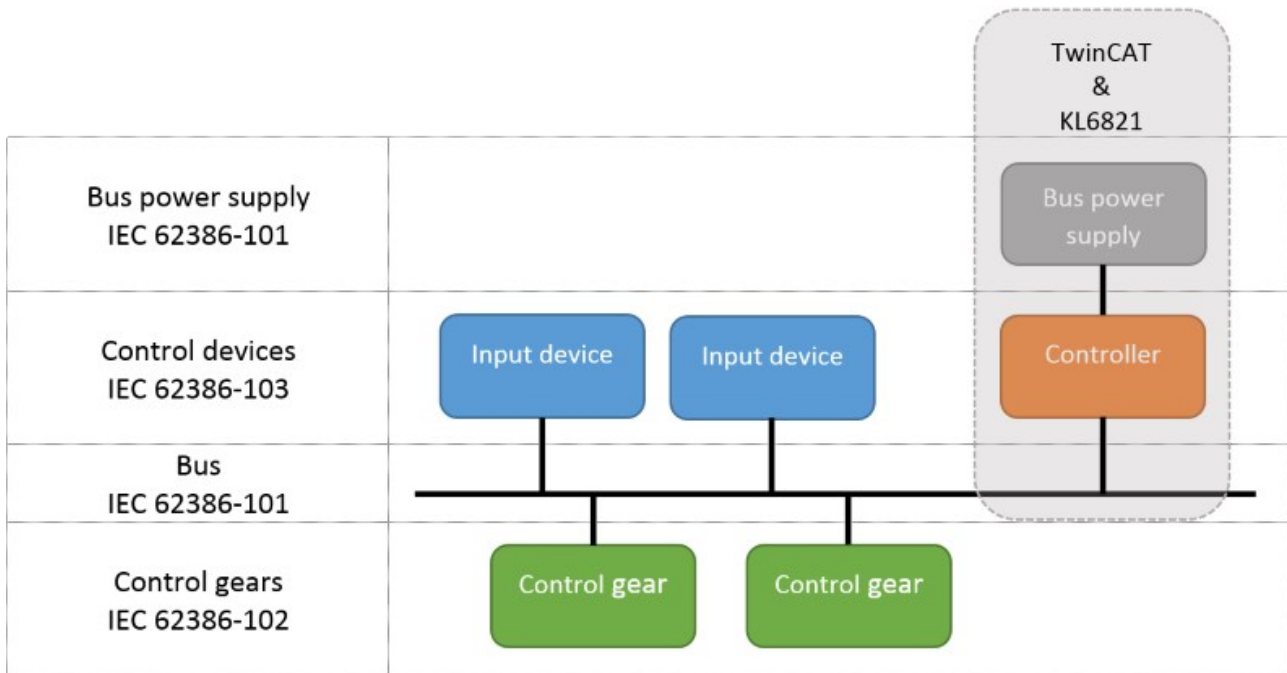
<https://www.dali-alliance.org/products/4844/kl6821-dali-dali-2-multi-master-and-power-supply>

#### 3.1 IEC 62386

DALI is specified in the IEC 62386 standard and offers advantages such as flexibility, simplicity, user friendliness and robustness. IEC 62386 has been revised several times and was extended considerably in November 2014 with the publication of the second revision. While in the first revision only DALI control gears (lamps) were considered, from the second revision onwards DALI control devices (sensors and application controllers) are also included. These are described in the respective section of IEC 62386:

IEC 62386-101	General system properties such as cabling, mains supply and frame structure
IEC 62386-102	General properties of the DALI control gears
	IEC 62386-201: Fluorescent lamps (device type 0) IEC 62386-202: Emergency lighting (device type 1) IEC 62386-203: Discharge lamps (device type 2) IEC 62386-207: LED modules (device type 6) ...
IEC 62386-103	General properties of the DALI control devices
	IEC 62386-301: Push buttons IEC 62386-302: Absolute input devices IEC 62386-303: Occupancy sensor IEC 62386-304: Light sensor ...

The IEC 62386-101, IEC 62386-102 and IEC 62386-103 standards describe general properties, while the IEC 62386-2xx and IEC 62386-3xx standards specify the individual device types. IEC 62386-103 and IEC 62386-3xx were included in Revision 2 of the DALI standard.

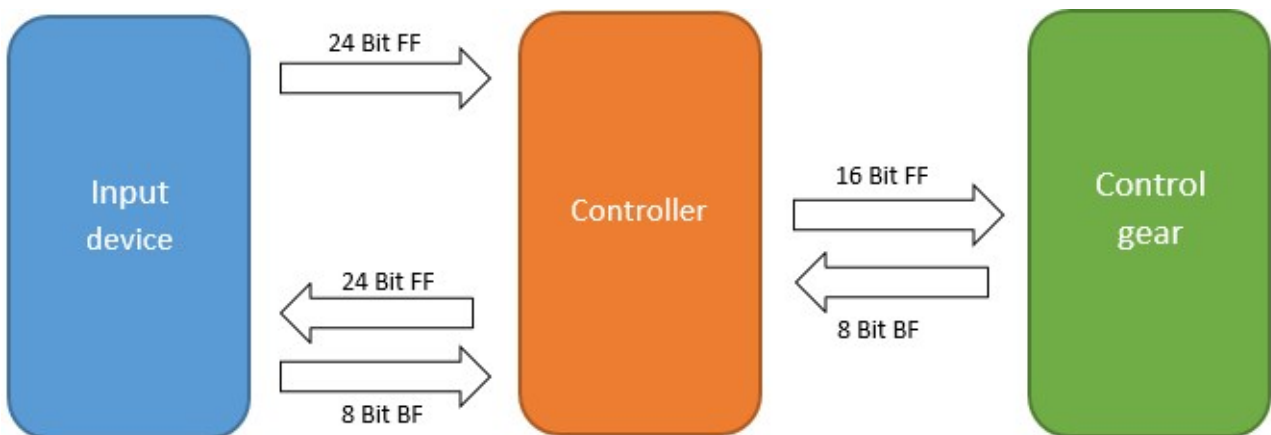


Up to 64 DALI control gears and up to 64 DALI control devices can be connected to the KL6821/EL6821 per DALI line. The KL6821/EL6821 represents the DALI controller. One such device exists for each DALI line. Up to 64 control gears, which have to be DALI/DSI devices, can be connected to the KL6811. Any number of DALI lines (KL6811, KL6821 or EL6821) can be operated with a single TwinCAT controller.

### 3.2 Communication

With regard to the communication, a distinction is made between three frame types:

- 16-bit query, configuration and control frame.
- 24-bit query, configuration and control frame.
- 24-bit event frame.



BF: backward frame  
FF: forward frame

#### 16-bit frames

16-bit frames are always sent from a DALI controller to a DALI control gear. They are used for configuring the devices, querying parameters or sending control commands. For certain DALI commands the DALI control gear sends an 8-bit backward frame. DALI control gears only send an 8-bit frame when requested.

In the Tc3\_DALI library these commands are identified by the PLC function blocks with the prefix FB\_DALI102 or FB\_DALI2xx, e.g. FB\_DALI102QueryActualLevel or FB\_DALI207QueryFeatures.

## 24-bit frames

24-bit frames are always sent from a DALI controller to a DALI control device. They are used for configuring the devices, querying parameters or sending control commands. For certain DALI commands the DALI control device sends an 8-bit backward frame.

In the Tc3\_DALI library these commands are identified by the PLC function blocks with the prefix FB\_DALI103 or FB\_DALI3xx, e.g. FB\_DALI103QueryOperatingMode or FB\_DALI303SetHoldTimer.

## 24-bit events

DALI control devices are able to send events. They are always evaluated by the DALI controller and have a length of 24 bits.

Individual events can be filtered out and further processed with the function blocks FB\_DALIGetInputNotification [► 841] and FB\_DALIGetPowerCycleNotification [► 843].



Further information on DALI can be found on the homepage of the DALI Alliance (<https://www.dali-alliance.org>) and in the IEC 62386 standard.



The KL6811 only supports the first revision of the DALI standard. It is not possible to operate control devices with the KL6811.

---

## 3.3 Priorities

If several DALI control devices are connected to a DALI line, priorities control concurrent access to the DALI bus. According to IEC 62386-103, all DALI-2 devices that can initiate sending of a DALI command (controllers) or sending of an event (input devices) on the DALI bus are referred to as DALI control device.

All DALI-2 devices of a DALI line must share the same data line. To avoid collisions during sending, the sending device checks whether the DALI bus has already been assigned. Sending takes place after a certain settling time, once the DALI bus is free. For high-priority DALI commands the bus access takes place after a short settling time, for low-priority commands the settling time is longer. In other words, high-priority DALI commands are given preference over low-priority DALI commands.

DALI control gears are defined in IEC 62386-102. They are not capable of sending DALI commands or events independently. DALI control gears may only return the 8-bit backward frame to forward frames sent by a DALI controller (see also [Communication \[► 10\]](#)). Since a DALI controller waits for the backward frame, the 8-bit backward frame has the shortest settling time. This settling time is shorter than for DALI commands with the highest priority. This means that DALI forward frames can be processed without interference from other DALI commands.

The priorities used by a DALI controller for sending the DALI commands are referred to as command priorities and are mapped by the data type [E\\_DALICommandPriority \[► 856\]](#). Command priorities can have 5 different values:

- **Low:** DALI priority 5
- **Middle low:** DALI priority 4
- **Middle:** DALI priority 3
- **Middle high:** DALI priority 2
- **High:** DALI priority 1

Most function blocks referred to in chapter [Part 102 \(control gears\) \[► 21\]](#) have the input *eCommandPriority*. This input is used to specify the priority with which the DALI commands are to be sent via the KL6821/EL6821.

Events also have a priority (event priority), which is represented by the data type `E_DALIEventPriority` [► 858]. Event priorities can have 4 values in the range *Low* (DALI priority 5) to *Middle high* (DALI priority 2). The event priority is written as a parameter (see instance variable `eventPriority` [► 233]) into the respective instances of the DALI devices.

Priority *High* (DALI priority 1) is only allowed for DALI-2 commands and cannot be used for events.

Tc3\_DALI uses the following default values for the priorities:

<code>E_DALICommandPriority/ E_DALIEventPriority</code>	Application
Low	-
MiddleLow	Light sensor events (Part 304). All other DALI commands.
Middle	Events of push buttons (Part 301), absolute input devices (Part 302) and occupancy sensors (Part 303).
MiddleHigh	DALI commands for writing parameters and for addressing DALI devices.
High	DALI commands for transactions (from the second DALI command).

When selecting priorities, care should always be taken to ensure that time-critical events that are important for switching the lighting have a higher priority than the DALI commands themselves. Non-system-critical DALI commands, such as the cyclic querying of states for the display in a visualization, should be sent with a lower priority.

Priorities for DALI commands (`E_DALICommandPriority` [► 856]) are supported from Tc3\_DALI V3.11.0.0. If the KL6821 is used, it must contain the firmware BD or newer. Older firmware versions always send DALI commands with the priority *High*.



Neither the KL6811 nor the Tc2\_DALI library support priorities for DALI commands.

---

Priorities are always important in situations where DALI sensors (input devices) send events and DALI commands are sent in parallel via the KL6821/EL6821. If only DALI control gears and a DALI controller (KL6821/EL6821) are connected to a DALI line, the priorities of the DALI commands are of secondary importance. The priorities of the DALI commands can also be neglected if the DALI sensors on the DALI line do not send notifications.

Further details about the DALI priorities can also be found in the following chapter [Bus Timing](#) [► 12].

## 3.4 Bus Timing

The following describes the structure and operation of the DALI protocol. This description focuses on the most important basic principles. For a full explanation, the IEC 62386 standard, in particular Part 101, should be consulted.

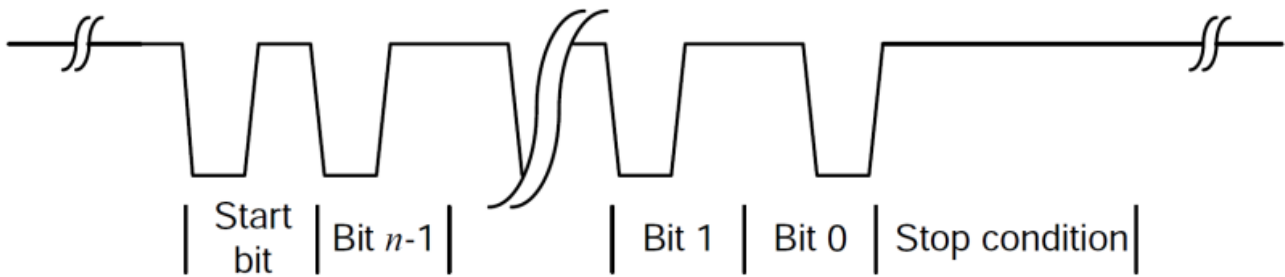
### 3.4.1 Structure data frame

Each Forward Frame (FF) and Backward Frame (BF) basically consists of:

- 1 start bit
- n data bits
- 1 stop condition

The *most significant bit* (MSB) is transmitted at the beginning.

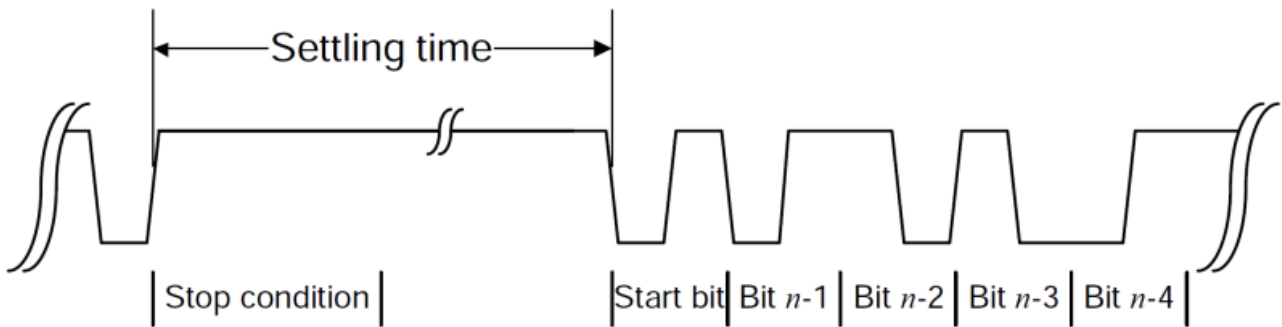




Transmission is via Manchester coding with a data transfer rate of 1200 bits per second. Thus, each bit has a length of 0.833 ms ( $1 / 1200 = 0.000833$ ).

The *Stop condition* has a length of at least 2.45 ms.

A fixed time (*settling time*) must be waited between the sending of two data frames before the sending of the next data frame can be started.



The length of the *settling time* depends on the DALI priority with which a data frame is sent. The higher the DALI priority, the smaller the *settling time*.

Settling time	Minimum	Mean value	Maximum
between FF and BF	5.5 ms	8.0 ms	10.5 ms
before each FF (DALI priority <i>High</i> )	13.5 ms	14.1 ms	14.7 ms
before each FF (DALI priority <i>Middle high</i> )	14.9 ms	15.5 ms	16.1 ms
before each FF (DALI priority <i>Middle</i> )	16.3 ms	17.0 ms	17.7 ms
before each FF (DALI priority <i>Middle low</i> )	17.9 ms	18.6 ms	19.3 ms
before each FF (DALI priority <i>Low</i> )	19.5 ms	20.3 ms	21.2 ms

More information about DALI priorities is also available in the chapter [Priorities](#) [► 11].

Thus, data frames with a higher DALI priority (low *settling time*) occupy the DALI bus earlier, compared to data frames with a lower DALI priority (high *settling time*).

Certain DALI commands must be sent twice within 100 ms (send-twice) without the receiver being allowed to receive another DALI command in the meantime. Only then will the DALI command be recognized as valid by the receiver. This is primarily used with DALI commands that are used to configure DALI devices. So that the two DALI commands are not interrupted by another DALI command, the 2nd DALI command is always sent with DALI priority *High*. The DALI priority *High* is reserved for these DALI commands and must not be used in any other context.

### 3.4.2 Transmission length

The approximate transmission length can be determined from the bit length and the structure of the data frame. For further (simplified) consideration, a mean *settling time* of 17.0 ms is used for the forward frames (FF), and a mean settling time of 8 ms for the backward frames (BF). Between the two DALI commands sent within 100 ms (send-twice), a *settling time* of 14.1 ms is used.

The number of bits results from the number of data bits (8, 16 or 24) plus the start bit. Thus 9, 17 or 25 bits are transmitted with the respective frames.

16-bit frame without backward frame:  
 $17.0 \text{ ms} + (17 \times 0.833 \text{ ms}) = \mathbf{31.2 \text{ ms}}$ .

16-bit frame with backward frame:  
 $17.0 \text{ ms} + (17 \times 0.833 \text{ ms}) + 8.0 \text{ ms} + (9 \times 0.833 \text{ ms}) = \mathbf{46.7 \text{ ms}}$ .

16-bit frame send-twice:  
 $17.0 \text{ ms} + (17 \times 0.833 \text{ ms}) + 14.1 \text{ ms} + (17 \times 0.833 \text{ ms}) = \mathbf{59.4 \text{ ms}}$ .

24-bit frame without backward frame:  
 $17.0 \text{ ms} + (25 \times 0.833 \text{ ms}) = \mathbf{37.8 \text{ ms}}$ .

24-bit frame with backward frame:  
 $17.0 \text{ ms} + (25 \times 0.833 \text{ ms}) + 8.0 \text{ ms} + (9 \times 0.833 \text{ ms}) = \mathbf{53.3 \text{ ms}}$ .

24-bit frame send-twice:  
 $17.0 \text{ ms} + (25 \times 0.833 \text{ ms}) + 14.1 \text{ ms} + (25 \times 0.833 \text{ ms}) = \mathbf{72.8 \text{ ms}}$ .

The following table lists the average transmission lengths of the individual frames. This results in a maximum possible number of frames per second. The PLC program should be designed in such a way that the number of frames always falls below the maximum.

Frame	Transmission length	Frames per second
16-bit frame without backward frame	31.2 ms	approx. 32
16-bit frame with backward frame	46.7 ms	approx. 21
16-bit frame send-twice	59.4 ms	approx. 16
24-bit frame without backward frame	37.8 ms	approx. 26
24-bit frame with backward frame	53.3 ms	approx. 18
24-bit frame send-twice	72.8 ms	approx. 13

### 3.4.3 Collision detection

The generation of the DALI frames is achieved in a DALI device by changing between high and low level within defined times. At a low level, the DALI bus is pulled towards 0 V against a current limitation. With a high level, the DALI connection from the DALI device is high-resistance.

With DALI-2, it may happen that several DALI devices try to send DALI frames independently of each other. For this reason, DALI-2 includes collision avoidance, collision detection and collision resolution.

Collision avoidance is achieved by using the DALI priorities. Before a DALI device sends a DALI frame, it is checked whether the DALI bus is free. Only if the DALI bus is free (high level), a bus access may take place. Correct use of DALI priorities reduces the probability of simultaneous bus access and thus minimizes the number of collisions.

Nevertheless, the DALI priorities cannot completely avoid collisions on the DALI bus, since DALI frames from different DALI devices may have the same DALI priority. For this reason, DALI-2 has collision detection and collision resolution.

The following diagram explains the collision resolution.

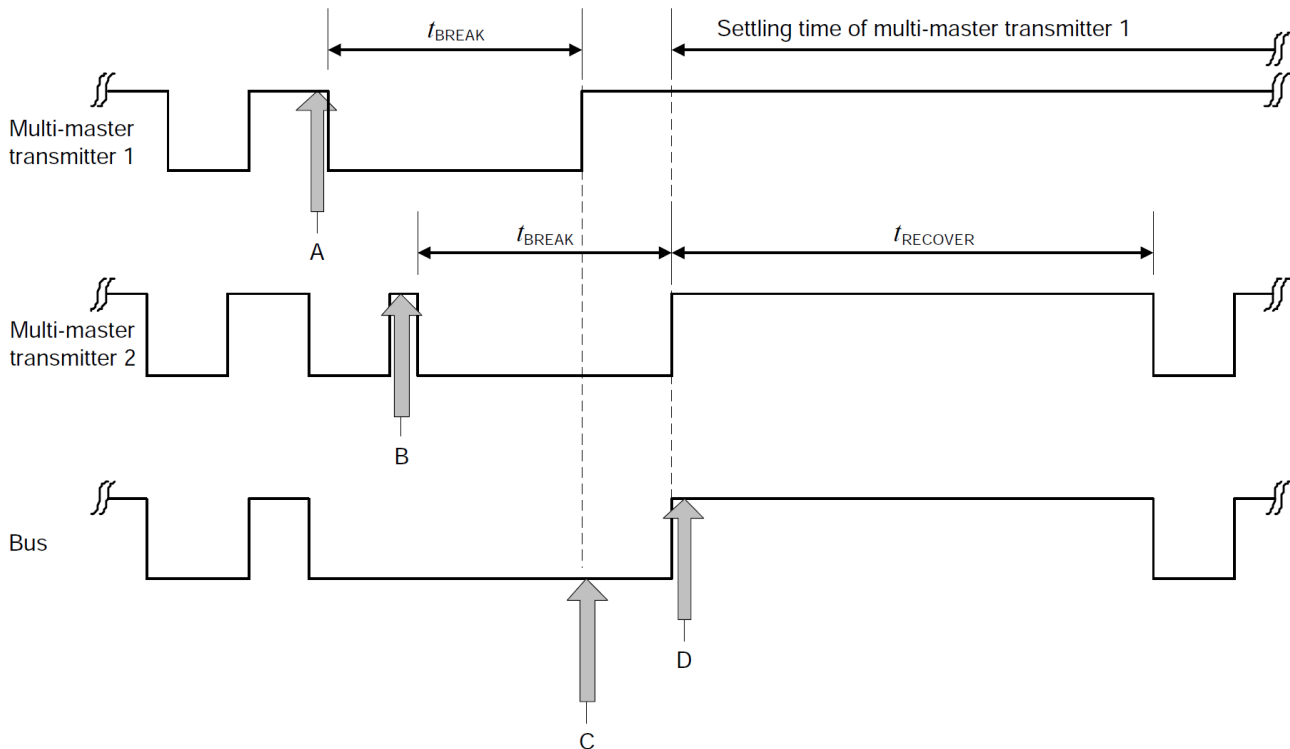
If several DALI devices send a high level, this cannot be detected by the DALI devices. The resulting voltage on the DALI bus is also a high level in this case.

At point A DALI device 1 detects a collision, because an attempt is made to generate a high signal at this point, but the DALI bus is pulled to low level by DALI device 2. DALI device 1 starts the break sequence for this reason. During this time the DALI device pulls the DALI bus to low level.

At point B DALI device 2 tries to generate a high level. However, since the DALI bus is pulled to low level by DALI device 1, DALI device 2 also detects a collision and also starts the break sequence.

At point C the break sequence of DALI device 1 has expired. Subsequently, it is checked whether the DALI bus is still at low level. Since this is the case, the system waits until the DALI bus is free again. DALI device 1 then starts sending the DALI frame again, including the *settling time*.

If the break sequence is finished at DALI device 2, the DALI bus is not occupied by any other DALI device (point D). Therefore the recover sequence is started at DALI device 2 and then the DALI frame is sent again directly (without *settling time*).



The break sequence has a length of 1.2 ms to 1.4 ms, while the recover sequence can be between 4.0 ms and 4.6 ms.

Collisions on the DALI bus interrupt the transmission on the DALI bus for several milliseconds. This further reduces the data throughput. For this reason, a DALI system should be put together and configured in such a way that as few collisions as possible occur.

### 3.4.4 Cycle times PLC tasks

For practical application, the cycle times of the PLC tasks should, if possible, always be set so that the maximum data transfer rate from the DALI bus is achieved.

Tests were performed to determine the number of frames at different cycle times of the PLC tasks. For this purpose, a PLC program was used, which cyclically sends 6 frames independently of each other. Three 16-bit frames (2 x without backward frame, 1 x with backward frame) and three 24-bit frames (2 x without backward frame, 1 x with backward frame) were sent. Since the send-twice frames are only of importance for the configuration of DALI devices, they were not considered further. The total transmission length of the 6 frames was thus 238 ms (2 x 21.2 ms + 46.7 ms + 2 x 37.8 ms + 53.3 ms). This means that the sample program could send a maximum of 25 frames per second (1000 ms / 238 ms x 6). Events that are additionally sent by possible DALI sensors (input devices) are not taken into account.

	80 ms	60 ms	40 ms	30 ms	20 ms	10 ms	8 ms	6 ms	4 ms
30 ms	6	7	7	8	8	8	8	8	8
15 ms	11	11	12	12	13	13	13	13	13
10 ms	15	15	16	17	17	17	17	17	17
8 ms	18	18	19	19	20	20	20	20	20
6 ms	20	21	21	22	22	22	22	22	22
4 ms	23	23	24	25	25	25	25	25	25
2 ms	23	24	25	25	25	25	25	25	25

The times in the top line (4 ms ... 80 ms) specify the cycle time of the PLC task from which the DALI commands are started. The times (2 ms ... 30 ms) in the first column specify the cycle time of the PLC task for background communication.

### 3.4.5 Summary

Even though the test program is only representative, it can be clearly seen that the cycle time of the background communication has a decisive influence on the data throughput. If a maximum data transfer rate is required on the DALI bus, the following points must be observed:

- K-bus and fieldbus should be arranged so that the cycle time for the PLC task accessing the DALI terminal (background communication) does not exceed 6 ms.
- The number of events of the DALI sensors (input devices) should be as low as possible. The more events are sent, the higher the probability of collisions on the DALI bus. The DALI sensors should be configured so that the number of events is minimal.
- To reduce the number of collisions on the DALI bus, the DALI priorities should be used. Recommendations for this can be found in the chapter [DALI priorities \[► 11\]](#).
- A large number of DALI control devices also increases the probability of collisions on the DALI bus. If necessary, the DALI control devices must be divided among different DALI lines. DALI control devices are DALI controllers and DALI sensors (see chapter [Communication \[► 10\]](#)).

## 3.5 Memory banks

Memory banks are freely accessible memory areas in which device-specific information and properties can be stored. The contents of the memory banks can be read with `FB_DALI10xReadMemoryLocation` (see [FB\\_DALI102ReadMemoryLocation \[► 129\]](#) and [FB\\_DALI103ReadMemoryLocation \[► 210\]](#)) and, if enabled, written with `FB_DALI10xWriteMemoryLocationNoReply` (see [FB\\_DALI102WriteMemoryLocationNoReply \[► 146\]](#) and [FB\\_DALI103WriteMemoryLocationNoReply \[► 227\]](#)).

Part of the memory banks can be provided with write protection.

A DALI device can support a maximum of 256 memory banks, each with up to 255 bytes, with memory banks 200 to 255 being reserved. Memory bank 0 and memory bank 1 are predefined by IEC 62386.

#### Structure of memory bank 0:

Memory bank 0 is read only and contains general, vendor-specific information about the DALI control gear or DALI control device. Every certified DALI device must implement memory bank 0. Up to offset 16#1A the fields are defined by IEC 62386 as follows.

Offset	Description	Default values
16#00	Offset of the last memory area inside the memory bank that can be accessed.	Vendor-specific
16#01	Reserved, not implemented	
16#02	Number of the last memory bank that can be accessed.	Vendor-specific
16#03	GTIN byte 0 (MSB)	Vendor-specific
16#04	GTIN byte 1	Vendor-specific
16#05	GTIN byte 2	Vendor-specific
16#06	GTIN byte 3	Vendor-specific
16#07	GTIN byte 4	Vendor-specific
16#08	GTIN byte 5 (LSB)	Vendor-specific
16#09	Firmware Version (major)	Vendor-specific
16#0A	Firmware Version (minor)	Vendor-specific
16#0B	Identification number byte 0 (MSB)	Vendor-specific
16#0C	Identification number byte 1	Vendor-specific
16#0D	Identification number byte 2	Vendor-specific
16#0E	Identification number byte 3	Vendor-specific
16#0F	Identification number byte 4	Vendor-specific
16#10	Identification number byte 5	Vendor-specific
16#11	Identification number byte 6	Vendor-specific
16#12	Identification number byte 7 (LSB)	Vendor-specific
16#13	Hardware version (major)	Vendor-specific
16#14	Hardware version (minor)	Vendor-specific
16#15	101 Version number of the current DALI standard	Vendor-specific
16#16	102 Version numbers of all integrated DALI control gears	Vendor-specific
16#17	103 Version numbers of all integrated DALI control devices	Vendor-specific
16#18	Number of logical control units in the device	Vendor-specific
16#19	Number of logical control gears in the device	Vendor-specific
16#1A	Index number of this logical DALI control gear or DALI control device	Vendor-specific
16#1B... 16#7F	Reserved, not implemented	
16#80... 16#FE	Additional device information	Vendor-specific
16#FF	Reserved, not implemented	

**Structure of memory bank 1:**

Memory bank 1 can be used by the device vendor to store further information in the DALI device. Up to offset 16#10 the fields are defined by IEC 62386 as follows.

Offset	Description	Default values	Memory
16#00	Offset of the last memory area inside the memory bank that can be accessed.	Vendor-specific (16#10... 16#FE)	
16#01	Indicator byte	Vendor-specific	
16#02	Lock byte for memory bank 1. Writeable bytes become changeable through the value 16#55. No other values make writing possible.	16#FF	
16#03	OEM GTIN byte 0 (MSB)	16#FF	Lockable by byte 16#02
16#04	OEM GTIN byte 1	16#FF	Lockable by byte 16#02
16#05	OEM GTIN byte 2	16#FF	Lockable by byte 16#02
16#06	OEM GTIN byte 3	16#FF	Lockable by byte 16#02
16#07	OEM GTIN byte 4	16#FF	Lockable by byte 16#02
16#08	OEM GTIN byte 5 (LSB)	16#FF	Lockable by byte 16#02
16#09	OEM Identification number byte 0 (MSB)	16#FF	Lockable by byte 16#02
16#0A	OEM Identification number byte 1	16#FF	Lockable by byte 16#02
16#0B	OEM Identification number byte 2	16#FF	Lockable by byte 16#02
16#0C	OEM Identification number byte 3	16#FF	Lockable by byte 16#02
16#0D	OEM Identification number byte 4	16#FF	Lockable by byte 16#02
16#0E	OEM Identification number byte 5	16#FF	Lockable by byte 16#02
16#0F	OEM Identification number byte 6	16#FF	Lockable by byte 16#02
16#10	OEM Identification number byte 7 (LSB)	16#FF	Lockable by byte 16#02
16#11... 16#FE	Additional device information	Vendor-specific	
16#FF	Reserved, not implemented		

### Structure of memory banks 2 to 199:

The device vendor can use memory banks 2 to 199 to supply further parameters. The structure of the memory banks is always as shown below. The vendor of the DALI device must be consulted regarding the contents and the possibility to write individual bytes.

Offset	Description	Default values	Memory
16#00	Offset of the last memory area inside the memory bank that can be accessed.	Vendor-specific (16#03... 16#FE)	
16#01	Indicator byte	Vendor-specific	
16#02	Lock byte for the memory bank. Writeable bytes become changeable through the value 16#55. No other values make writing possible.	16#FF	
16#03... 16#FE	Additional device information	Vendor-specific	Vendor-specific; lockable by byte 16#02 if enabled by the vendor
16#FF	Reserved, not implemented		

### Access to memory bank 2 on the EL6821

Since the KL6821/EL6821 represents a DALI control device according to IEC 62386, the Bus Terminal must offer memory bank 0 and memory bank 1. Other DALI control devices can access these memory banks via the corresponding DALI commands.

In addition, the EL6821 offers memory bank 2. In the EL6821 memory bank 2 is mapped by CoE object 16#8002. The first three fields (16#8002:01 ... 16#8002:03) are defined by IEC 62386 (see above). The fields 16#8002:04 to 16#8002:FF can be written and/or read individually.



Note that the offset in memory bank 2 and the subindex are shifted from the CoE. Thus, offset 10 in memory bank 2 corresponds to field 16#8002:0B.

Index	Name	Flags	Value	Unit
8002:0	Memory Bank 2	RO	> 254 <	
8002:01	Offset of last memory area	RO	0xFE (254)	
8002:02	Indicator Byte	RO	0x00 (0)	
8002:03	Lock Byte	RO	0xFF (255)	
8002:04	Additional Device Information #03	RW	0x00 (0)	
8002:05	Additional Device Information #04	RW	0x00 (0)	
8002:06	Additional Device Information #05	RW	0x00 (0)	
8002:07	Additional Device Information #06	RW	0x00 (0)	
8002:08	Additional Device Information #07	RW	0x00 (0)	
8002:09	Additional Device Information #08	RW	0x00 (0)	
8002:0A	Additional Device Information #09	RW	0x00 (0)	
8002:0B	Additional Device Information #10	RW	0x00 (0)	
8002:0C	Additional Device Information #11	RW	0x00 (0)	

The function blocks [FB\\_DALI102ReadMemoryLocation \[▶ 129\]](#)/[FB\\_DALI103ReadMemoryLocation \[▶ 210\]](#) or [FB\\_DALI102WriteMemoryLocationNoReply \[▶ 146\]](#)/[FB\\_DALI103WriteMemoryLocationNoReply \[▶ 227\]](#) are available for accessing the memory banks of a DALI device via the DALI bus.

If memory bank 2 of the own EL6821 is to be accessed from a PLC program, access is via the EtherCAT CoE interface. The TwinCAT library `Tc2_EtherCAT` offers the necessary function blocks for this (see [FB\\_EcCoeSdoRead](#) and [FB\\_EcCoeSdoWrite](#)).

The following example reads offset 3 (subindex 4) and writes offset 4 (subindex 5) of memory bank 2. The EL6821 must be located at the same controller where the PLC program is executed.

```

VAR
  fbCoERead      : FB_EcCoeSdoRead;
  fbCoEWrite     : FB_EcCoeSdoWrite;
  nValue         : USINT;
  bExecuteRead  : BOOL;
  bExecuteWrite  : BOOL;
END_VAR

fbCoERead(sNetId := F_CreateAmsNetId(GVL.stEL6821InData01.stAdsAddr.netId),
  nSlaveAddr := GVL.stEL6821InData01.stAdsAddr.port,
  nIndex := 16#8002,
  nSubIndex := 16#04,
  pDstBuf := ADR(nValue),
  cbBufLen := SIZEOF(nValue),
  bExecute := bExecuteRead);

IF (NOT fbCoERead.bBusy) THEN
  bExecuteRead := FALSE;
END_IF

fbCoEWrite(sNetId := F_CreateAmsNetId(GVL.stEL6821InData01.stAdsAddr.netId),
  nSlaveAddr := GVL.stEL6821InData01.stAdsAddr.port,
  nIndex := 16#8002,
  nSubIndex := 16#05,
  pSrcBuf := ADR(nValue),
  cbBufLen := SIZEOF(nValue),
  bExecute := bExecuteWrite);
  
```

```
IF (NOT fbCoEWrite.bBusy) THEN
  bExecuteWrite := FALSE;
END_IF
```



## 4 Programming

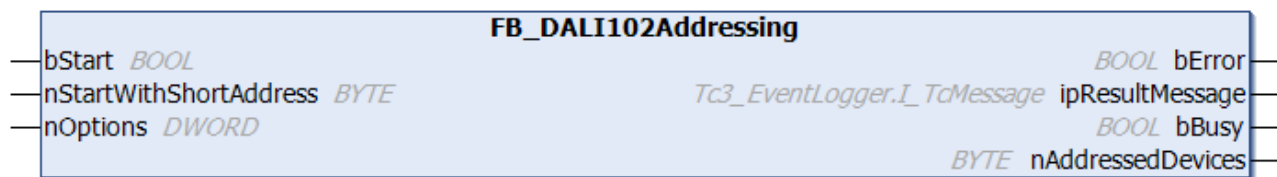
### 4.1 POU's

#### 4.1.1 Applications

##### 4.1.1.1 Part 102 (control gears)

##### 4.1.1.1.1 Addressing

##### 4.1.1.1.1.1 FB\_DALI102Addressing



This function block addresses the DALI control gears at random. The user has no influence on which DALI control gear is assigned which short address. Short addresses are allocated in ascending order.

Applying a positive edge to the *bStart* input starts the function block, and the *bBusy* output goes TRUE. Depending on the selected options (parameter *nOptions*) the group membership and scenes are subsequently deleted. The terminal now addresses all DALI control gears independently. Once all DALI control gears have been addressed, the *bBusy* output goes back to FALSE. The *nAddressedDevices* output variable supplies information about how many DALI control gears have received a short address. Processing this function block can take several minutes, depending on how many DALI control gears are attached.

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nStartWithShortAddress : BYTE := 0;
  nOptions        : DWORD := Tc3_DALI.GVL.cOpticalFeedback;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nStartWithShortAddress	BYTE	Short address, assigned to the first DALI control gear (0...63).
nOptions	DWORD	Options for addressing the DALI control gears (see table below). The individual constants must be linked with OR operators.

Constant	Description
Tc3_DALI.GVL.cCompleteNewInstallation	All DALI control gears are re-addressed, including control gears that already have a short address.
Tc3_DALI.GVL.cDeleteAllGroupAssignments	Before addressing commences, any group assignments are deleted for all DALI control gears, including those that are not actually addressed.
Tc3_DALI.GVL.cDeleteAllSceneAssignments	Before addressing commences, any scene assignments are deleted for all DALI control gears, including those that are not actually addressed.
Tc3_DALI.GVL.cOpticalFeedback	Prior to addressing, all control gears are set to <i>minLevel</i> . Newly addressed DALI control gears are assigned <i>maxLevel</i> brightness after assignment of the short address.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nAddressedDevices : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nAddressedDevices	BYTE	If addressing has been completed ( <i>bBusy</i> is FALSE), then the number of addressed control gears is shown at this output.

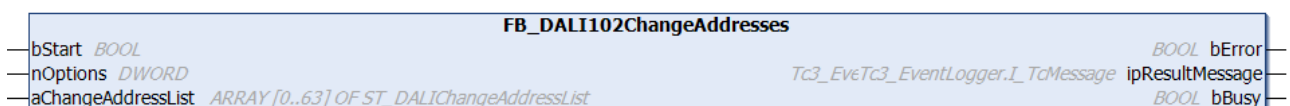
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.1.1.1.2 FB\_DALI102ChangeAddresses**



This function block can be used to change the short addresses of several DALI control gears.

A list of the DALI control gears for which the short address is to be changed is transferred in the array *aChangeAddressList* of type *ST\_DALIChangeAddressList* [▶ 852]. The list has 64 entries from 0 to 63. Each entry contains a variable *nOldAddress* and *nNewAddress* with which the address assignment is parameterized. The end of the list is programmed with a 255 entry at *nOldAddress*, so that the whole list does not necessarily have to be filled in. If this entry is missing, however, then all entries are accepted. When the function block is started (positive edge on *bStart*), the list end is first determined on the basis of the described entry and afterwards the valid list range is examined for the following false entries:

- Address entries > 63
- Double address entry on the source page *nOldAddress*
- Double address entry on the target page *nNewAddress* (leads to double assignment of an address and, hence, to errors)

The function block then uses the short addresses to determine the internal long addresses of the DALI devices and enters them in the *nRandomAddress* parameter of the list.

If an error occurs during queries or during reprogramming, this leads to an error entry for the respective device in the list element *nErrors* (see *ST\_DALIChangeAddressList* [▶ 852]).

The individual bits in the list element *nErrors* have the following meaning:

Bit	Error
0	Error during reading of the long address
1	-
2	-
3	-
4	-
5	Error during programming of the new short address
6	<i>nOldAddress</i> and <i>nNewAddress</i> have the same values

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nOptions        : DWORD := 0;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nOptions	DWORD	Reserved for future extensions.

 **Inputs/outputs**

```
VAR_IN_OUT
  aChangeAddressList : ARRAY [0..63] OF ST_DALIChangeAddressList;
END_VAR
```

Name	Type	Description
aChangeAddressList	<i>ST_DALIChangeAddressList</i> [▶ 852]	List of short addresses to be changed.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.1.1.2 Power control**

**4.1.1.1.2.1 FB\_DALI102ConstantLightControl**



The FB\_DALI102ConstantLightControl function block provides the basic functions for implementing constant light regulation.

The system attempts to regulate to a specified setpoint (*nSetpointValue*) by dimming up and down cyclically. The control dynamics are determined by a dead time (*tDeadTime*). The dead time defines the settling time between the individual DALI commands for changing the output value (*nActualLevel*). The smaller the dead time, the faster the control. A freely definable hysteresis (*nHysteresis*) prevents continuous oscillation around the setpoint. If the actual value is within the hysteresis range around the setpoint, the output value of the DALI control gears is not changed.

## Operation

The function block offers the option to address a single DALI control gear via a short address, several DALI control gears via a group address, or all DALI control gears of a DALI line via a broadcast.

The variables `minLevel` [▶ 150], `maxLevel` [▶ 150], `fadeRate` [▶ 150], `fadeTime` [▶ 151], `extendedFadeTimeBase` [▶ 152] and `extendedFadeTimeMultiplier` [▶ 152] are parameters that are stored separately in each DALI control gear. These variables can be changed in the DALI control gears by writing to the respective properties and with a positive edge at the input `bInitialize`.

## Control

The output value of the DALI control gears can be switched by positive edges at the inputs `bOn`, `bOff` and `bToggle`. If the DALI control gears are switched on and control is active (`bEnable` = TRUE), then they are regulated to the specified setpoint. If the control is not active (`bEnable` = FALSE), the output value of the DALI control gears remains unchanged.

The step-by-step adaption of the output value of the DALI control gears is carried out with the DALI commands STEP UP/STEP DOWN or UP/DOWN.

If the control deviation (`nDeviation`) is greater than `nHysteresis`, the DALI command UP/DOWN is used to adjust the output value. The number of steps by which the output value changes is specified by the `eFadeRate` property.

If the control deviation is between  $nHysteresis/2$  and `nHysteresis`, the DALI command STEP UP/STEP DOWN is used and thus the output value is only adjusted by one step.

If the control deviation is smaller than  $nHysteresis/2$ , the output value of the DALI control gears remains unchanged.

The control deviation is calculated from  $nSetpointValue - nActualBrightness$ .

The dead time (`tDeadtime`) specifies the time after which the output value is adapted. The DALI command STEP UP/STEP DOWN or UP/DOWN is called only once per control cycle.

## Operation via the inputs bOn, bOff, and bToggle

The output value of the DALI control gears can be changed immediately by positive edges at the inputs `bOn`, `bOff`, and `bToggle`. This is independent of whether the control has been enabled or disabled (`bEnable`).

The variables `fadeTime`, `extendedFadeTimeBase`, and `extendedFadeTimeMultiplier` specify the speed at which the output value is changed when the DALI control gears are switched on via `bOn` or `bToggle`. Switching off the DALI control gears by `bOff` or `bToggle` takes place immediately. In this case, `nSwitchOnLevel` is used as the switch-on value. This value must lie in the range of the properties `nMinLevel` and `nMaxLevel`.

## DALI short address reference device (nReferenceDeviceAddress)

If several DALI control gears are addressed, the current output value of the reference DALI control gear is read out via `nReferenceDeviceAddress`. The DALI control gears are set to the desired output value, depending on the state of the reference DALI control gear.

The parameter `nReferenceDeviceAddress` is also used if the output value of the reference DALI control gear is read out cyclically in the background (`tCycleActualLevel` > 0 sec).

No reference DALI control gear is required if a single DALI control gear is to be controlled with the function block (`eAddressType` = `E_DALIAddressType.Short`). The output value of the individual DALI control gear is determined via `nAddress`. The parameter `nReferenceDeviceAddress` has no meaning in this case.

If `nReferenceDeviceAddress` is used, it must always contain the short address of a DALI control gear, which is also contained in the addressed DALI group. For a broadcast a DALI control gear with the corresponding short address must be present on the DALI line.

**Sample**

The following sample shows how a DALI light sensor can be combined with the FB\_DALIConstantLightControl function block. In this sample, the PD11-BMS-FLAT DALI sensor from B.E.G. is used.

Since only the measured brightness is required, the instance for the movement sensor is disabled (property *bEnableOccupancy*).

The brightness is read out from the DALI sensor via the input *bQueryBrightness*. Thus, the instance for the light sensor can also be disabled (property *bEnableBrightness*).

At the end of the dead time, the output *bControlCycleEnding* of the constant light regulation is set to TRUE. This positive edge is connected to the input *bQueryBrightness* of the DALI sensor. This means that the current brightness value is read out immediately before calculating the control deviation (*nDeviation*).

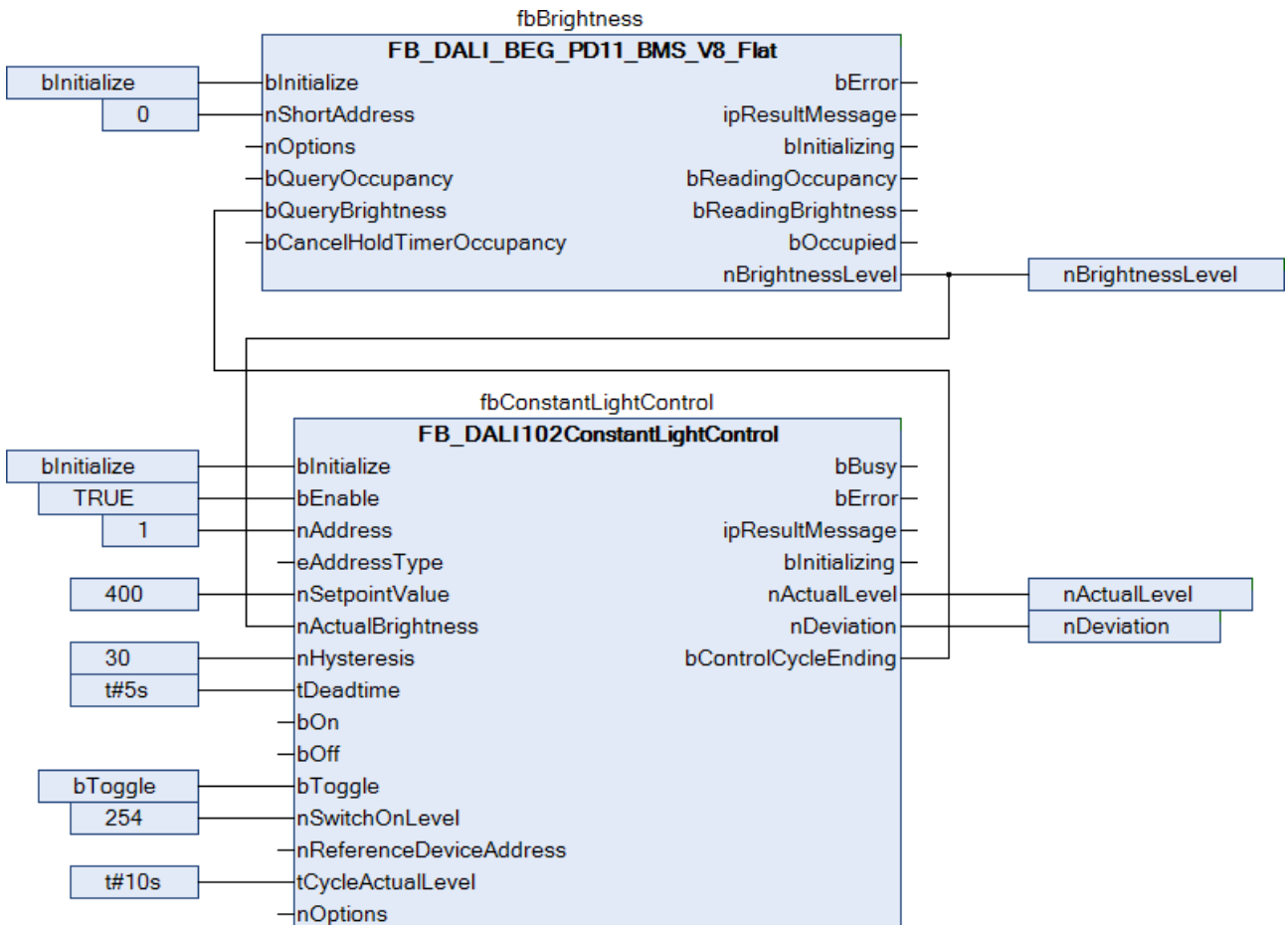


By deactivating both instances, the DALI sensor does not send any events and the DALI bus is not unnecessarily loaded.

The properties are initialized directly when the instance is declared. This means that explicit assignment at runtime is no longer necessary.

```
PROGRAM P_ConstantLightControl
VAR
  fbBrightness      : FB_DALI_BEG_PD11_BMS_V8_Flat(Communication.fbKL6821Communication) :=
                    (bEnableOccupancy := FALSE,
                     bEnableBrightness := FALSE);
  fbConstantLightControl : FB_DALI102ConstantLightControl(Communication.fbKL6821Communication) :=
                    (nMinLevel := 85,
                     nMaxLevel := 254);

  bInitialize       : BOOL;
  bToggle           : BOOL;
  nBrightnessLevel  : UINT;
  nActualLevel      : BYTE;
  nDeviation        : DINT;
END_VAR
```



 **Inputs**

```

VAR_INPUT
  bInitialize          : BOOL := FALSE;
  bEnable             : BOOL := TRUE;
  nAddress            : BYTE;
  eAddressType        : E_DALIAddressType := E_DALIAddressType.Short;
  nSetpointValue      : UINT := 500;
  nActualBrightness   : UINT := 500;
  nHysteresis         : UINT := 30;
  tDeadtime           : TIME := T#5S;
  bOn                 : BOOL;
  bOff                : BOOL;
  bToggle             : BOOL;
  nSwitchOnLevel      : BYTE := 254;
  nLevelMemoryMode    : BYTE := 254;
  nReferenceDeviceAddress : BYTE;
  tCycleActualLevel   : TIME := T#30S;
  nOptions             : DWORD := 0;
END_VAR
    
```

Name	Type	Description
bInitialize	BOOL	A positive edge at this input writes the values of all properties to the DALI control gears. Writing the properties to the DALI control gears is only possible if no other functions are being executed by the function block ( <i>bBusy</i> = FALSE).
bEnable	BOOL	Activates constant light regulation as soon as this input is TRUE. If the input is FALSE, the constant light regulation is disabled.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nSetpointValue	UINT	The setpoint is applied to this input (0...65535).
nActualBrightness	UINT	The actual value is applied at this input.
nHysteresis	UINT	Control hysteresis (1...65535) around the setpoint. If the actual value is within this range, the output values of the DALI control gears are not changed.
tDeadtime	TIME	After the dead time (2...3600 s) has elapsed, the control deviation is recalculated and, if necessary, the output values of the DALI control gears are reduced or increased.
bOn	BOOL	A positive edge at this input sets the DALI control gears to <i>nSwitchOnLevel</i> .
bOff	BOOL	The DALI control gears are switched off via a positive edge at this input.
bToggle	BOOL	A positive edge at this input toggles the DALI control gears between Off and <i>nSwitchOnLevel</i> .
nSwitchOnLevel	BYTE	Output value ( <i>minLevel</i> ... <i>maxLevel</i> ) for switching on the DALI control gears by the bOn and bToggle inputs.
nReferenceDeviceAddress	BYTE	Short address (0...63) of the reference DALI control gear for group call and broadcast. This parameter is not evaluated if <i>eAddressType</i> = <i>E_DALIAddressType.Short</i> . In this case, the reference DALI control gear is read out via <i>nAddress</i> .
tCycleActualLevel	TIME	Cycle time with which the current output value of the reference DALI control gear is read out in the background. Set the cycle time such that as few DALI commands as possible are sent. If the time is set to 0 sec, no reading takes place.
nOptions	DWORD	Reserved for future extensions.



Constant	Description
Tc3_DALI.GVL.cMemoryMode	Activates memory mode.

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bInitializing   : BOOL;
  nActualLevel    : BYTE;
  nDeviation      : DINT;
  bControlCycleEnding : BOOL;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control gears has been started, and remains active until all DALI commands have been executed.
nActualLevel	BYTE	Current output value (0, <i>minLevel</i> ... <i>maxLevel</i> , 255) of the reference DALI control gear that is addressed via <i>nReferenceDeviceAddress</i> . If <i>eAddressType</i> = <i>E_DALIAddressType.Short</i> , the reference DALI control gear is read out via <i>nAddress</i> . A value of 255 (MASK) indicates that an error occurred while reading the output value from the reference DALI control gear. The cause could be, for example, a technical defect or that the DALI control gear is in the start-up phase. In this case <i>bError</i> is not set to TRUE.
nDeviation	DINT	Current control deviation ( <i>nSetpointValue</i> - <i>nActualBrightness</i> )
bControlCycleEnding	BOOL	Before the dead time ( <i>tDeadtime</i> ) expires, this output is set to TRUE for 500 ms. This output can be used, for example, to read the current brightness from a DALI light sensor.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].



Name	Type	Access	Initial value	Description
nMaxLevel	BYTE	Get, Set	254	See variable <a href="#">maxLevel</a> [► 150].
nMinLevel	BYTE	Get, Set	126	See variable <a href="#">minLevel</a> [► 150].
eFadeRate	E_DALIFadeRate	Get, Set	N045StepsPerSec	See variable <a href="#">fadeRate</a> [► 150].
eFadeTime	E_DALIFadeTime	Get, Set	Disabled	See variable <a href="#">fadeTime</a> [► 151].
eExtendedFadeTimeBase	E_DALIExtendedFadeTimeBase	Get, Set	Base01	See variable <a href="#">extendedFadeTimeBase</a> [► 152].
eExtendedFadeTimeMultiplier	E_DALIExtendedFadeTimeMultiplier	Get, Set	Disabled	See variable <a href="#">extendedFadeTimeMultiplier</a> [► 152].
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_DALI from v3.10.5.0

**4.1.1.1.2.2 FB\_DALI102Dimmer1Switch**



The function block FB\_DALI102Dimmer1Switch provides the main functions for implementing a push-dimmer.

**Operation**

The input *bSwitch* is intended for linking with a push button. Pressing the button briefly switches the DALI control gears on or off. Pressing the button longer causes the output value of the DALI control gears to dim cyclically between the minimum and maximum values (see below).

The output value of the DALI control gears can be changed by positive edges at the inputs *bRecallMinLevel*, *bRecallMaxLevel*, *bOn*, *bOff*, *bToggle*, *bGoToScene* and *bSetLevel* (see below).

The function block offers the option to address a single DALI control gear via a short address, several DALI control gears via a group address or all DALI control gears of a DALI line via a broadcast.

The variables `minLevel` [▶ 150], `maxLevel` [▶ 150], `fadeRate` [▶ 150], `fadeTime` [▶ 151], `extendedFadeTimeBase` [▶ 152] and `extendedFadeTimeMultiplier` [▶ 152] are parameters that are stored separately in each DALI control gear. These variables can be changed in the DALI control gears via the properties and via the input `bInitialize`.

If DALI commands are sent for setting the output values (`bBusy = TRUE`), further positive edges at the inputs are ignored. New commands can be executed as soon as `bBusy` has been reset again.

### Operation via the input `bSwitch`

A short signal at the input `bSwitch` switches the DALI control gears on or off, depending on the state of the reference DALI control gear. If the signal is present for longer, the system switches to dimmer mode and the output values of the DALI control gears move cyclically between `minLevel` and `maxLevel` until `bSwitch` is FALSE again. If the output value of the reference DALI control gear reaches the limit value `minLevel` or `maxLevel`, the function block changes its dimming direction.

If `bSwitch` is set to FALSE again, the current output values of the DALI control gears remain unchanged. A new pulse at the input `bSwitch` sets the output values to 0. If the input `bSwitch` is briefly deactivated in dimmer mode, the function block changes its dimming direction.

The speed at which the output values of the DALI control gears change in dimmer mode is determined by the variable `fadeRate` in the individual DALI control gears. The speed at which the DALI control gears are switched on again is specified by the variables `fadeTime`, `extendedFadeTimeBase`, and `extendedFadeTimeMultiplier`.

### Operation via the inputs `bOn`, `bOff`, `bGoToScene` and `bSetLevel`

The output value of the DALI control gears can be changed immediately by positive edges at the inputs `bOn`, `bOff`, `bGoToScene`, and `bSetLevel`.

For `bOn`, `bGoToScene` and `bSetLevel`, the speed at which the specified value is to be reached is specified by the variables `fadeTime`, `extendedFadeTimeBase` and `extendedFadeTimeMultiplier`. `bOff` switches the DALI control gears off immediately.

### Memory mode

When switching on, a distinction must be made as to whether memory mode (see `nOptions`) is active or not. If memory mode is active, the last set value is adopted as the output value for the DALI control gears when the device is switched on. If memory mode is not active, the output value from the parameter `nLevelMemoryMode` is used for the DALI control gears. It is irrelevant whether the DALI control gears are switched via the input `bOn`, `bToggle` or `bSwitch`.

### DALI short address reference device (`nReferenceDeviceAddress`)

If several DALI control gears are addressed, the current output value of the reference DALI control gear is read out via `nReferenceDeviceAddress`. The DALI control gears are set to the desired value, depending on the state of the reference DALI control gear.

The parameter `nReferenceDeviceAddress` is also used if the output value of the reference DALI control gear is read out cyclically in the background (`tCycleActualLevel > 0` sec).

No reference DALI control gear is required if a single DALI control gear is to be controlled with the function block (`eAddressType = E_DALIAddressType.Short`). The output value of the individual DALI control gear is determined via `nAddress`. The parameter `nReferenceDeviceAddress` has no meaning in this case.

If `nReferenceDeviceAddress` is used, it must always contain the short address of a DALI control gear, which is also contained in the addressed DALI group. For a broadcast a DALI control gear with the corresponding short address must be present on the DALI line.

### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nAddress             : BYTE;
  eAddressType        : E_DALIAddressType := E_DALIAddressType.Short;
```

```
bSwitch          : BOOL;  
bRecallMaxLevel  : BOOL;  
bRecallMinLevel  : BOOL;  
bOn              : BOOL;  
bOff            : BOOL;  
bToggle         : BOOL;  
bGoToScene      : BOOL;  
nScene          : BYTE;  
bSetLevel       : BOOL;  
nLevel          : BYTE := 254;  
nLevelMemoryMode : BYTE := 254;  
nReferenceDeviceAddress : BYTE;  
tCycleActualLevel : TIME := T#30S;  
nOptions        : DWORD := 0;  
END_VAR
```

Name	Type	Description
bInitialize	BOOL	A positive edge at this input writes the values of all properties to the DALI control gears. Writing the properties to the DALI control gears is only possible if no other functions are being executed by the function block ( <i>bBusy</i> = FALSE).
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
bSwitch	BOOL	A short signal at these inputs switches the DALI control gears on or off. If the signal is present for a longer period, the system switches to dimmer mode and the output values of the DALI control gears move to <i>minLevel</i> or <i>maxLevel</i> .
bRecallMaxLevel	BOOL	A positive edge at this input sets the DALI control gears to <i>maxLevel</i> .
bRecallMinLevel	BOOL	A positive edge at this input sets the DALI control gears to <i>minLevel</i> .
bOn	BOOL	If memory mode is active (see <i>nOptions</i> ), a positive edge at this input sets the DALI control gears to their output value before the last switching off. If memory mode is not active, the DALI control gears are set to <i>nLevelMemoryMode</i> .
bOff	BOOL	The DALI control gears are switched off via a positive edge at this input. The previous output value is stored internally to be used for switching on if memory mode is active (see <i>nOptions</i> ).
bToggle	BOOL	Each positive edge at this input causes the DALI control gears to switch between off and their output value before the last switch-off (memory mode active) or <i>nLevelMemoryMode</i> (memory mode not active).
bGoToScene	BOOL	A positive edge at this input sets the output value of the DALI control gears to the value stored in <i>nScene</i> .
nScene	BYTE	Scene (0...15) to be called up in the DALI control gears by <i>bGoToScene</i> . Each DALI control gear has its own output value for each scene.
bSetLevel	BOOL	A positive edge at this input sets the output value of the DALI control gears to the value specified by <i>nLevel</i> .
nLevel	BYTE	Output value (0, <i>minLevel</i> ... <i>maxLevel</i> , 255) to be called up in the DALI control gears by <i>bSetLevel</i> . The output value remains unchanged if the value is 255 (MASK).
nLevelMemoryMode	BYTE	Output value ( <i>minLevel</i> ... <i>maxLevel</i> ) for switching on the DALI control gears when memory mode is not active.
nReferenceDeviceAddress	BYTE	Short address (0...63) of the reference DALI control gear for group call and broadcast. This parameter is not evaluated if <i>eAddressType</i> = <i>E_DALIAddressType.Short</i> . In this case, the reference DALI control gear is read out via <i>nAddress</i> .
tCycleActualLevel	TIME	Cycle time with which the current output value of the reference DALI control gear is read out in the background. Set the cycle time such that as few DALI commands as possible are sent. If the time is set to 0 sec, no reading takes place.
nOptions	DWORD	Options that affect the behavior of the function block. The individual constants must be linked with OR operators.

Constant	Description
Tc3_DALI.GVL.cMemoryMode	Activates memory mode.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bInitializing   : BOOL;
  nActualLevel    : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control gears has been started, and remains active until all DALI commands have been executed.
nActualLevel	BYTE	Current output value (0, <i>minLevel</i> ... <i>maxLevel</i> , 255) of the reference DALI control gear that is addressed via <i>nReferenceDeviceAddress</i> . If <i>eAddressType</i> = <i>E_DALIAddressType.Short</i> , the reference DALI control gear is read out via <i>nAddress</i> . A value of 255 (MASK) indicates that an error occurred while reading the output value from the reference DALI control gear. The cause could be, for example, a technical defect or that the DALI control gear is in the start-up phase. In this case <i>bError</i> is not set to TRUE.

 **Properties**

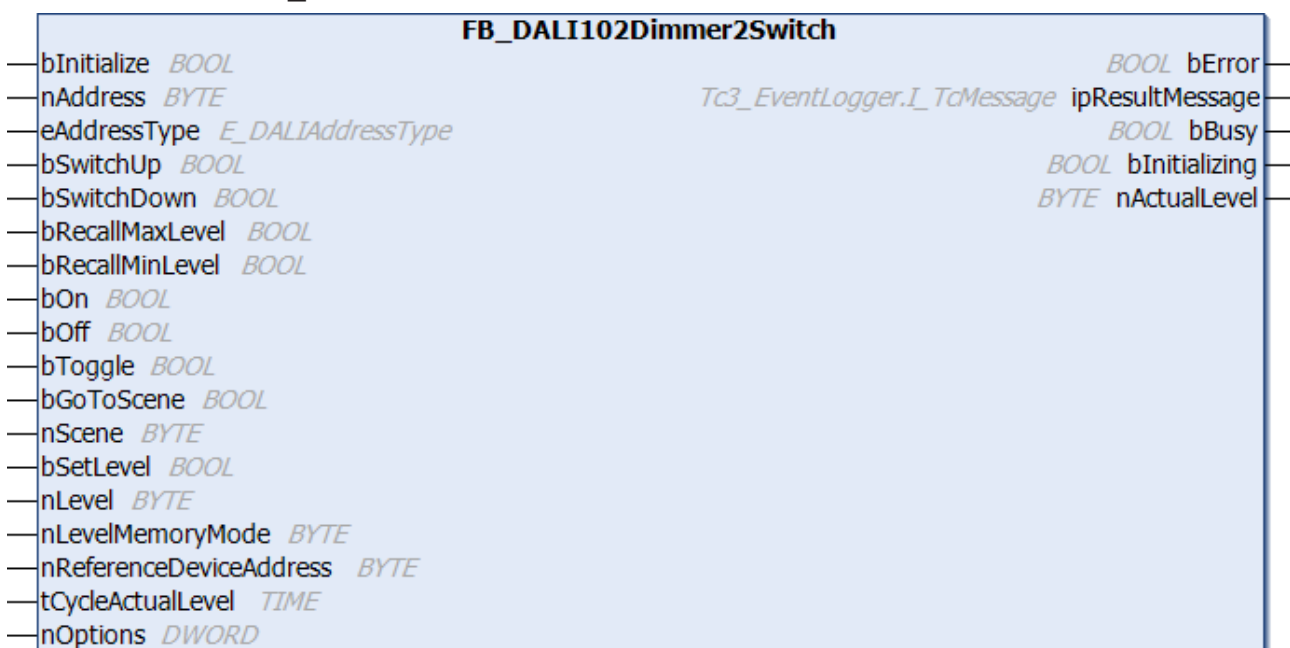
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
nMaxLevel	BYTE	Get, Set	254	See variable <a href="#">maxLevel</a> [► 150].
nMinLevel	BYTE	Get, Set	126	See variable <a href="#">minLevel</a> [► 150].
eFadeRate	E_DALIFadeRate	Get, Set	N045StepsPerSec	See variable <a href="#">fadeRate</a> [► 150].
eFadeTime	E_DALIFadeTime	Get, Set	Disabled	See variable <a href="#">fadeTime</a> [► 151].
eExtendedFadeTimeBase	E_DALIExtendedFadeTimeBase	Get, Set	Base01	See variable <a href="#">extendedFadeTimeBase</a> [► 152].
eExtendedFadeTimeMultiplier	E_DALIExtendedFadeTimeMultiplier	Get, Set	Disabled	See variable <a href="#">extendedFadeTimeMultiplier</a> [► 152].
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

**4.1.1.1.2.3 FB\_DALI102Dimmer2Switch**



The function block FB\_DALI102Dimmer2Switch provides the main functions for implementing a push-dimmer.

**Operation**

The inputs *bSwitchUp* and *bSwitchDown* are intended for linking with two push buttons. Pressing the button briefly switches the DALI control gears on or off. Pressing the button longer causes the output value of the DALI control gears to dim down to the minimum or up to the maximum values, respectively (see below).

The output value of the DALI control gears can be changed by positive edges at the inputs *bRecallMinLevel*, *bRecallMaxLevel*, *bOn*, *bOff*, *bToggle*, *bGoToScene* and *bSetLevel* (see below).

The function block offers the option to address a single DALI control gear via a short address, several DALI control gears via a group address or all DALI control gears of a DALI line via a broadcast.

The variables *minLevel*, *maxLevel*, *fadeRate*, *fadeTime*, *extendedFadeTimeBase* and *extendedFadeTimeMultiplier* are parameters that are stored separately in each DALI control gear. These variables can be changed in the DALI control gears by writing to the properties and with a positive edge at the input *bInitialize*.

If DALI commands are sent for setting the output values (*bBusy* = TRUE), further positive edges at the inputs are ignored. New commands can be executed as soon as *bBusy* has been reset again.

### Operation via the *bSwitchUp* and *bSwitchDown* inputs

A short signal at the inputs *bSwitchUp* or *bSwitchDown* switches the DALI control gears on or off, depending on the state of the reference DALI control gear. If the signal is present for a longer time, the system switches to dimmer mode and the output values of the DALI control gears move up to *minLevel* or *maxLevel* or until *bSwitchUp* or *bSwitchDown* are FALSE again.

If *bSwitchUp* or *bSwitchDown* is set to FALSE again, the current output values of the DALI control gears remain unchanged. A new pulse at the input *bSwitchUp* or *bSwitchDown* sets the output values to 0.

The speed at which the output values of the DALI control gears change in dimmer mode is determined by the variable *fadeRate* in the individual DALI control gears. The speed at which the DALI control gears are switched on again is specified by the variables *fadeTime*, *extendedFadeTimeBase*, and *extendedFadeTimeMultiplier*.

### Operation via the inputs *bOn*, *bOff*, *bGoToScene*, and *bSetLevel*

The output value of the DALI control gears can be changed immediately by positive edges at the inputs *bOn*, *bOff*, *bGoToScene*, and *bSetLevel*.

For *bOn*, *bGoToScene* and *bSetLevel*, the speed at which the specified value is to be reached is specified by the variables *fadeTime*, *extendedFadeTimeBase* and *extendedFadeTimeMultiplier*. *bOff* switches the DALI control gears off immediately.

### Memory mode

When switching on, a distinction must be made as to whether memory mode (see *nOptions*) is active or not. If memory mode is active, the last set value is adopted as the output value for the DALI control gears when the device is switched on. If memory mode is not active, the output value from the parameter *nLevelMemoryMode* is used for the DALI control gears. It is irrelevant whether the DALI control gears are switched via the input *bOn*, *bToggle*, *bSwitchUp*, or *bSwitchDown*.

### DALI short address reference device (*nReferenceDeviceAddress*)

If several DALI control gears are addressed, the current output value of the reference DALI control gear is read out via *nReferenceDeviceAddress*. The DALI control gears are set to the desired value, depending on the state of the reference DALI control gear.

The parameter *nReferenceDeviceAddress* is also used if the output value of the reference DALI control gear is read out cyclically in the background (*tCycleActualLevel* > 0 sec).

No reference DALI control gear is required if a single DALI control gear is to be controlled with the function block (*eAddressType* = *E\_DALIAddressType.Short*). The output value of the individual DALI control gear is determined via *nAddress*. The parameter *nReferenceDeviceAddress* has no meaning in this case.

If *nReferenceDeviceAddress* is used, it must always contain the short address of a DALI control gear, which is also contained in the addressed DALI group. For a broadcast a DALI control gear with the corresponding short address must be present on the DALI line.

### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  bSwitchUp      : BOOL;
  bSwitchDown    : BOOL;
```

```
bRecallMaxLevel      : BOOL;  
bRecallMinLevel      : BOOL;  
bOn                   : BOOL;  
bOff                  : BOOL;  
bToggle              : BOOL;  
bGoToScene           : BOOL;  
nScene               : BYTE;  
bSetLevel            : BOOL;  
nLevel               : BYTE := 254;  
nLevelMemoryMode     : BYTE := 254;  
nReferenceDeviceAddress : BYTE;  
tCycleActualLevel    : TIME := T#30S;  
nOptions             : DWORD := 0;  
END_VAR
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
bSwitchUp/ bSwitchDown	BOOL	A short signal at these inputs switches the DALI control gears on or off. If the signal is present for a longer period, the system switches to dimmer mode and the output values of the DALI control gears move to <i>minLevel</i> or <i>maxLevel</i> .
bRecallMaxLevel	BOOL	A positive edge at this input sets the DALI control gears to <i>maxLevel</i> .
bRecallMinLevel	BOOL	A positive edge at this input sets the DALI control gears to <i>minLevel</i> .
bOn	BOOL	If memory mode is active (see <i>nOptions</i> ), a positive edge at this input sets the DALI control gears to their output value before the last switching off. If memory mode is not active, the DALI control gears are set to <i>nLevelMemoryMode</i> .
bOff	BOOL	The DALI control gears are switched off via a positive edge at this input. The previous output value is stored internally to be used for switching on if memory mode is active (see <i>nOptions</i> ).
bToggle	BOOL	Each positive edge at this input causes the DALI control gears to switch between off and their output value before the last switch-off (memory mode active) or <i>nLevelMemoryMode</i> (memory mode not active).
bGoToScene	BOOL	A positive edge at this input sets the output value of the DALI control gears to the value stored in <i>nScene</i> .
nScene	BYTE	Scene (0...15) to be called up in the DALI control gears by <i>bGoToScene</i> . Each DALI control gear has its own output value for each scene.
bSetLevel	BOOL	A positive edge at this input sets the output value of the DALI control gears to the value specified by <i>nLevel</i> .
nLevel	BYTE	Output value (0, <i>minLevel</i> ... <i>maxLevel</i> , 255) to be called up in the DALI control gears by <i>bSetLevel</i> . The output value remains unchanged if the value is 255 (MASK).
nLevelMemoryMode	BYTE	Output value ( <i>minLevel</i> ... <i>maxLevel</i> ) for switching on the DALI control gears when memory mode is not active.
nReferenceDeviceAddress	BYTE	Short address (0...63) of the reference DALI control gear for group call and broadcast. This parameter is not evaluated if <i>eAddressType</i> = <i>E_DALIAddressType.Short</i> . In this case, the reference DALI control gear is read out via <i>nAddress</i> .
tCycleActualLevel	TIME	Cycle time with which the current output value of the reference DALI control gear is read out in the background. Set the cycle time such that as few DALI commands as possible are sent. If the time is set to 0 sec, no reading takes place.
nOptions	DWORD	Options that affect the behavior of the function block. The individual constants must be linked with OR operators.

Constant	Description
Tc3_DALI.GVL.cMemoryMode	Activates memory mode.

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  bInitializing  : BOOL;
  nActualLevel   : BYTE;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control gears has been started, and remains active until all DALI commands have been executed.
nActualLevel	BYTE	Current output value (0, <i>minLevel</i> ... <i>maxLevel</i> , 255) of the reference DALI control gear that is addressed via <i>nReferenceDeviceAddress</i> . If <i>eAddressType</i> = <i>E_DALIAddressType.Short</i> , the reference DALI control gear is read out via <i>nAddress</i> . A value of 255 (MASK) indicates that an error occurred while reading the output value from the reference DALI control gear. The cause could be, for example, a technical defect or that the DALI control gear is in the start-up phase. In this case <i>bError</i> is not set to TRUE.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
nMaxLevel	BYTE	Get, Set	254	See variable <a href="#">maxLevel</a> [► 150].
nMinLevel	BYTE	Get, Set	126	See variable <a href="#">minLevel</a> [► 150].
eFadeRate	E_DALIFadeRate	Get, Set	N045StepsPerSec	See variable <a href="#">fadeRate</a> [► 150].
eFadeTime	E_DALIFadeTime	Get, Set	Disabled	See variable <a href="#">fadeTime</a> [► 151].
eExtendedFadeTimeBase	E_DALIExtendedFadeTimeBase	Get, Set	Base01	See variable <a href="#">extendedFadeTimeBase</a> [► 152].
eExtendedFadeTimeMultiplier	E_DALIExtendedFadeTimeMultiplier	Get, Set	Disabled	See variable <a href="#">extendedFadeTimeMultiplier</a> [► 152].
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_DALI from v3.10.5.0

**4.1.1.1.3 Settings**

**4.1.1.1.3.1 FB\_DALI102GetSettings**



This function block reads several variables (see table [below](#) [► 40]) from selected DALI control gears and saves them in the structure [ST\\_DALIControlGearSettings](#) [► 852].

The DALI control gears that are to be read can be specified by the input variable *nDevices*. Each bit of this variable corresponds to a short address.

Applying a positive edge to the *bStart* input starts the function block, and the *bBusy* output goes TRUE. A check is first made as to whether a DALI control gear is present at all. If this is the case, then the *bPresent* bit is set in the respective structure (see [ST\\_DALIControlGearSettings](#) [► 852]), after which the settings of the control gear are read one by one and written in the associated variables in the structure. If it is found that a device is not available, the reading is skipped and work continues with the next device. The structure index here reflects the address of the device. In other words, data for the device with short address 0 is located at *arrDALIDeviceSettings[0]*, and so on through to the device with short address 63 having its data at *arrDALIDeviceSettings[63]*. If a read error occurs when reading from a device, the corresponding bit in *nErrors* is set for the respective structure without the function block itself switching to error mode. The following table shows which bit is set in the *nErrors* variable when an error occurs during the reading of a variable from a control gear.

Bit	Error
0	An error occurred while attempting to seek the DALI control gear.
1	Error while reading the variable <a href="#">actualLevel</a> [ <a href="#">▶ 150</a> ].
2	Error while reading the variable <a href="#">powerOnLevel</a> [ <a href="#">▶ 150</a> ].
3	Error while reading the variable <a href="#">systemFailureLevel</a> [ <a href="#">▶ 150</a> ].
4	Error while reading the variable <a href="#">minLevel</a> [ <a href="#">▶ 150</a> ].
5	Error while reading the variable <a href="#">maxLevel</a> [ <a href="#">▶ 150</a> ].
6	Error while reading the variable <a href="#">fadeRate</a> [ <a href="#">▶ 150</a> ].
7	Error while reading the variable <a href="#">fadeTime</a> [ <a href="#">▶ 151</a> ].
8	Error while reading the variables <a href="#">extendedFadeTimeBase</a> [ <a href="#">▶ 152</a> ] and <a href="#">extendedFadeTimeMultiplier</a> [ <a href="#">▶ 152</a> ].
9	Error while reading the variable <a href="#">randomAddress</a> [ <a href="#">▶ 153</a> ].
10	Error while reading the variable <a href="#">gearGroups</a> [ <a href="#">▶ 154</a> ].
11	Error while reading the variable <a href="#">scene0-scene15</a> [ <a href="#">▶ 154</a> ].
12	Error while reading the variable <a href="#">statusInformation</a> [ <a href="#">▶ 126</a> ].
13	Error while reading the version number from the <a href="#">MemoryBank</a> [ <a href="#">▶ 16</a> ].
14	Error while reading the supported device types.
15	Error while reading the variable <a href="#">physicalMinLevel</a> [ <a href="#">▶ 154</a> ].

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nDevices        : LWORD;
  nOptions        : DWORD;
  bCancel         : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nDevices	LWORD	Variable for selecting the DALI control gears to be read. Each bit of the variable represents a DALI control gear (0...63). All short addresses are queried with a setting of 16#FFFF_FFFF_FFFF_FFFF.
nOptions	DWORD	Options for reading the variables (see table below). The individual constants must be linked with OR operators.
bCancel	BOOL	A positive edge at this input will disable the function block and hence abort the reading of the variables.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

Options for reading the variables:

Constant	Description
Tc3_DALI.GVL.cOptionActualDimLevel	Reads the current <u>output value</u> [▶ 150].
Tc3_DALI.GVL.cOptionPowerOnLevel	Reads the current <u>switch-on value</u> [▶ 150].
Tc3_DALI.GVL.cOptionSystemFailureLevel	Reads the current <u>setting value</u> [▶ 150] in case of a DALI bus error.
Tc3_DALI.GVL.cOptionMinLevel	Reads the <u>maximum</u> [▶ 150] output value.
Tc3_DALI.GVL.cOptionMaxLevel	Reads the <u>minimum</u> [▶ 150] output value.
Tc3_DALI.GVL.cOptionFadeRateFadeTime	Reads the <u>FadeRate</u> [▶ 150] and the <u>FadeTime</u> [▶ 151].
Tc3_DALI.GVL.cOptionExtendedFadeTime	Reads the extended <u>FadeTimeBase</u> [▶ 152] and the <u>FadeTimeMultiplier</u> [▶ 152].
Tc3_DALI.GVL.cOptionRandomAddress	Reads the <u>random address</u> [▶ 153].
Tc3_DALI.GVL.cOptionGroups	Reads the <u>group allocations</u> [▶ 154].
Tc3_DALI.GVL.cOptionSceneLevels	Reads the <u>scene settings</u> [▶ 154].
Tc3_DALI.GVL.cOptionStatusInformation	Reads the <u>status information</u> [▶ 126].
Tc3_DALI.GVL.cOptionVersionNumber	Reads the <u>version number</u> [▶ 16] of the DALI control gear.
Tc3_DALI.GVL.cOptionDeviceType	Reads the <u>device type</u> [▶ 107].
Tc3_DALI.GVL.cOptionPhysicalMinLevel	Reads the <u>PhysicalMinLevel</u> [▶ 154].
Tc3_DALI.GVL.cOptionAll	Reads all variables.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Inputs/outputs

```
VAR_IN_OUT
  arrDALIDeviceSettings : ARRAY [0..63] of ST_DALIControlGearSettings;
END_VAR
```

Name	Type	Description
arrDALIDeviceSettings	ST_DALIControlGearSettings [▶ 852]	Array containing the read parameters of the individual DALI control gears.

Outputs

```
VAR_OUTPUT
  bError : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy : BOOL;
  nCurrentShortAddress : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nCurrentShortAddress	BYTE	Indicates which short address is currently being queried.

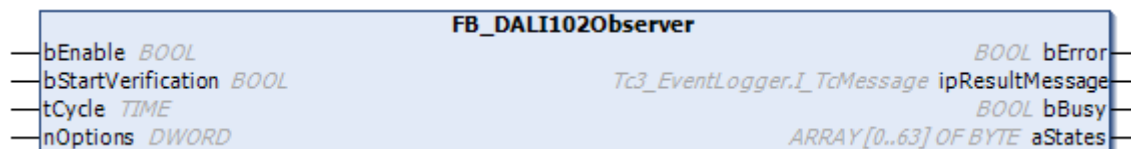
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

**4.1.1.1.3.2 FB\_DALI102Observer**



The function block checks all DALI control gears on a DALI line for errors. Furthermore, it is checked whether there are unaddressed DALI control gears on the DALI line.

For checking the DALI line, the function blocks [FB\\_DALI102QueryControlGearFailure](#) [▶ 104], [FB\\_DALI102QueryLampFailure](#) [▶ 114] and [FB\\_DALI102QueryMissingShortAddress](#) [▶ 118] are called cyclically (see *tCycle*). Here, all DALI commands are executed as a broadcast. This allows all DALI control gears of a DALI line to be checked with one DALI command.

*nOptions* specifies which checks are executed at the DALI line. The verification of the DALI line can also be started immediately by a positive edge at the input *bStartVerification*.

If the error state of a DALI control gear changes, the current state is read out from all DALI control gears. The output *aStates* contains the returned status information of the entire DALI line. Each element of *aStates* contains the two error information of the respective DALI control gear. The meaning of the bits is defined as follows:

Bit	Meaning
0	Status of the DALI control gear (see <a href="#">controlGearFailure</a> [▶ 153])
1	Lamp failure (see <a href="#">lampFailure</a> [▶ 153])

Some error states can only be detected when the lamp is switched on.

If an unaddressed DALI control gear is detected, it is automatically addressed with the first free short address (starting at 0). Addressing is not executed if several DALI control gears without short address are detected.

If several defective DALI control gears are to be replaced or additional DALI control gears are to be added, this can only be done device by device. Add an unaddressed DALI control gear and wait until it is addressed (by using the option `Tc3_DALI.GVL.cOpticalFeedback` the newly addressed DALI control gear goes to the brightness `maxLevel`). Only then add the next unaddressed DALI control gear.



The function block should only be used with the DALI-2 certified KL6821/EL6821.

**Inputs**

```
VAR_INPUT
  bEnable          : BOOL := FALSE;
  bStartVerification : BOOL;
  tCycle           : TIME := T#1M;
  nOptions         : DWORD := GVL.cOpticalFeedback OR GVL.cCheckControlGearFailure OR GVL.cCheckLampFailure OR GVL.cCheckUnaddressedControlGears;
END_VAR
```

Name	Type	Description
bEnable	BOOL	If the input is TRUE, the verification of the DALI control gears is enabled. If the input is FALSE, the verification of the DALI control gears is disabled.
bStartVerification	BOOL	A positive edge at this input starts immediately the verification of the DALI line. This is independent of whether the verification has been enabled or disabled ( <i>bEnable</i> ).
tCycle	TIME	Cycle time in which the DALI control gears are verified. Set the cycle time such that as few DALI commands as possible are sent. If the time is set to 0 s, the verification of the DALI line is not executed cyclically.
nOptions	DWORD	Options for checking the DALI line (see table below). The individual constants must be linked with OR operators.

Options for verification

Constant	Description
Tc3_DALI.GVL.cCheckControlGearFailure	It is checked whether there are one or more defective DALI control gears on the DALI line.
Tc3_DALI.GVL.cCheckLampFailure	It is checked whether there are one or more DALI control gears with a defective lamp on the DALI line.
Tc3_DALI.GVL.cCheckUnaddressedControlGears	It is checked whether there are one or more new DALI control gears (without short address) on the DALI line.
Tc3_DALI.GVL.cOpticalFeedback	Prior to addressing, all control gears are set to <i>minLevel</i> . Newly addressed DALI control gears are assigned <i>maxLevel</i> brightness after assignment of the short address.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  aStates         : ARRAY [0..63] OF BYTE;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
aStates	BYTE	Array containing the read states of all DALI control gears (see table above).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

**4.1.1.1.3.3 FB\_DALI102SetSettings**



This function block writes the values stored in the structure [ST\\_DALIControlGearSettings](#) [▶ 852] to several variables (see table [below](#) [▶ 40]) of selected DALI control gears in a DALI line.

The DALI control gears that are to be written to can be specified by the input variable *nDevices*. Each bit of this variable corresponds to a short address.

Applying a positive edge to the *bStart* input starts the function block, and the *bBusy* output goes TRUE. A check is first made as to whether a DALI control gear is present at all. If this is the case, then the *bPresent* bit is set in the respective structure (see [ST\\_DALIControlGearSettings](#) [▶ 852]), after which the settings are written one by one to the control gear. If a device is detected as being missing, the writing is skipped and work continues with the next device. The structure index here reflects the address of the device. In other words, data for the device with short address 0 is located at *arrDALIDeviceSettings*[0], and so on through to the device with short address 63 having its data at *arrDALIDeviceSettings*[63]. If a device error occurs during writing, the corresponding bit in *nErrors* is set for the respective structure without the function block itself switching to error mode. The following table shows which bit is set in the *nErrors* variable when an error occurs during the reading of a variable from a control gear.



Bit	Error
0	An error occurred while attempting to seek the DALI control gear.
2	Error while writing the variable <code>powerOnLevel</code> [▶ 150].
3	Error while writing the variable <code>systemFailureLevel</code> [▶ 150].
4	Error while writing the variable <code>minLevel</code> [▶ 150].
5	Error while writing the variable <code>maxLevel</code> [▶ 150].
6	Error while writing the variable <code>fadeRate</code> [▶ 150].
7	Error while writing the variable <code>fadeTime</code> [▶ 151].
8	Error while writing the variables <code>extendedFadeTimeBase</code> [▶ 152] and <code>extendedFadeTimeMultiplier</code> [▶ 152].
10	Error while writing the variable <code>gearGroups</code> [▶ 154].
11	Error while writing the variable <code>scene0-scene15</code> [▶ 154].

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nDevices        : LWORD;
  nOptions        : DWORD;
  bCancel         : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nDevices	LWORD	Variable for selecting the DALI control gears that are to be initialized. Each bit of the variable represents a DALI control gear (0...63). All short addresses are written to with a setting of 16#FFFF_FFFF_FFFF_FFFF.
nOptions	DWORD	Options for writing the variables (see table below). The individual constants must be linked with OR operators.
bCancel	BOOL	A positive edge at this input will disable the function block and hence abort the initialization of the variable.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

Options for writing to the variables:

Constant	Description
Tc3_DALI.GVL.cOptionPowerOnLevel	Initializes the current <code>switch-on value</code> [▶ 150].
Tc3_DALI.GVL.cOptionSystemFailureLevel	Initializes the current <code>setting value</code> [▶ 150] in case of a DALI bus error.
Tc3_DALI.GVL.cOptionMinLevel	Initializes the <code>maximum</code> [▶ 150] output value.
Tc3_DALI.GVL.cOptionMaxLevel	Initializes the <code>minimum</code> [▶ 150] output value.
Tc3_DALI.GVL.cOptionFadeRate	Initializes the <code>FadeRate</code> [▶ 150].
Tc3_DALI.GVL.cOptionFadeTime	Initializes the <code>FadeTime</code> [▶ 151].
Tc3_DALI.GVL.cOptionExtendedFadeTime	Initializes the extended <code>FadeTimeBase</code> [▶ 152] and the <code>FadeTimeMultiplier</code> [▶ 152].
Tc3_DALI.GVL.cOptionGroups	Initializes the <code>group allocations</code> [▶ 154].
Tc3_DALI.GVL.cOptionSceneLevels	Initializes the <code>scene settings</code> [▶ 154].
Tc3_DALI.GVL.cOptionAll	Initializes all variables.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Inputs/outputs**

```
VAR_IN_OUT
  arrDALIDeviceSettings : ARRAY [0..63] of ST_DALIControlGearSettings;
END_VAR
```

Name	Type	Description
arrDALIDeviceSettings	ST_DALIControlGearSettings [▶ 852]	Array containing the read parameters of the individual DALI control gears.

**Outputs**

```
VAR_OUTPUT
  bError : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy : BOOL;
  nCurrentShortAddress : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nCurrentShortAddress	BYTE	Indicates which short address is currently being written to.

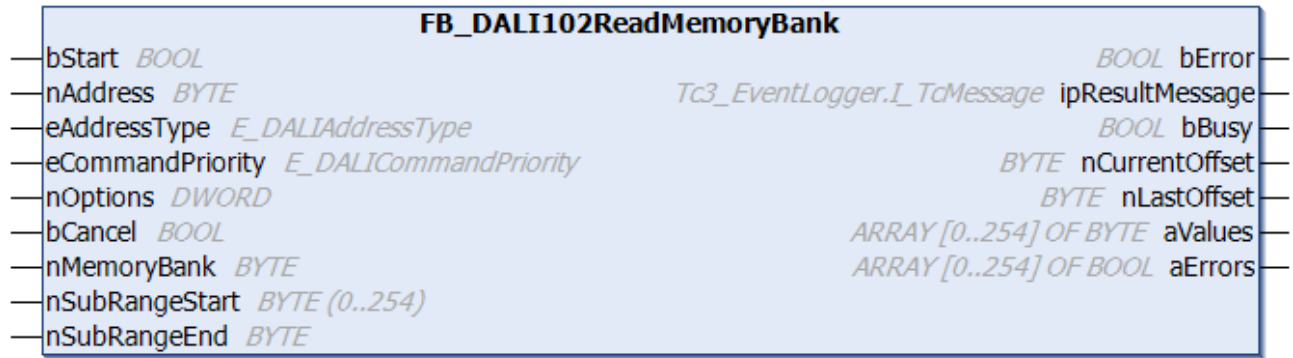
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

4.1.1.1.3.4 FB\_DALI102ReadMemoryBank



The specified memory bank of the DALI control gear is read (see [Memory banks](#) [▶ 16]). The memory bank is specified by the parameter *nMemoryBank*. The range to be read can be limited with the help of the parameters *nSubRangeStart* and *nSubRangeEnd*.

**i** The function block changes the DTR0 and the DTR1 of all connected DALI control gears.

**i** While the function block is active (*bBusy* = TRUE), all function blocks that also have write access to the DTRs of the connected DALI control gears on the same DALI line are blocked.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nOptions    : DWORD;
  bCancel     : BOOL;
  nMemoryBank : BYTE;
  nSubRangeStart : BYTE(0..254) := 2;
  nSubRangeEnd : BYTE := 255;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nOptions	DWORD	Reserved for future extensions.
bCancel	BOOL	A rising edge at this input disables the function block and aborts the reading of the variables.
nMemoryBank	BYTE	Specifies the memory bank for the read access.
nSubRangeStart	BYTE	Offset inside the memory bank from which reading is to take place.
nSubRangeEnd	BYTE	Offset inside the memory bank up to which reading is to take place.

nSubRangeEnd	Description
255	Offset 0 of the memory bank determines the offset up to which reading is to take place.
254	Offset 0 is not read at the beginning. Reading always takes place up to Offset 254.
<= 253	<i>nSubRangeEnd</i> may not be greater than the value in Offset 0. The partial areas of the memory bank are read with this setting.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nCurrentOffset  : BYTE;
  nLastOffset     : BYTE;
  aValues         : ARRAY [0..254] OF BYTE;
  aErrors         : ARRAY [0..254] OF BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nCurrentOffset	BYTE	The offset that is currently being read is output at this output.
nLastOffset	BYTE	Last valid offset of the memory bank that was read.
aValues	BYTE	Array containing the read parameters of the memory bank.
aErrors	BOOL	Array containing errors of the parameters of the memory bank that occurred during reading.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.1.2 Part 103 (control devices)

4.1.1.2.1 Addressing

4.1.1.2.1.1 FB\_DALI103Addressing



This function block addresses the DALI control devices at random. The user has no influence on which control device is assigned which short address. Short addresses are allocated in ascending order.

Applying a positive edge to the *bStart* input starts the function block, and the *bBusy* output goes TRUE. The terminal now addresses all DALI control devices independently. Once all DALI control devices have been addressed, the *bBusy* output goes back to FALSE. The output variable *nAddressedDevices* provides information on how many DALI control devices were assigned a short address. Depending on how many DALI control devices are connected, processing of this function block can take several minutes.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nStartWithShortAddress : BYTE := 0;
  nOptions        : DWORD := Tc3_DALI.GVL.cOpticalFeedback;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nStartWithShortAddress	BYTE	Short address assigned to the first DALI control device (0..63).
nOptions	DWORD	Options for addressing the DALI control devices (see table below). The individual constants must be linked with OR operators.

Constant	Description
Tc3_DALI.GVL.cCompleteNewInstallation	All DALI control devices are readdressed, even those that already have a short address.
Tc3_DALI.GVL.cOpticalFeedback	Newly addressed DALI control devices are allocated the DALI command IDENTIFY DEVICE once the short address has been assigned.

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
```

```
bBusy          : BOOL;
nAddressedDevices : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nAddressedDevices	BYTE	If addressing has been completed ( <i>bBusy</i> is FALSE), then the number of addressed DALI control devices is shown at this output.

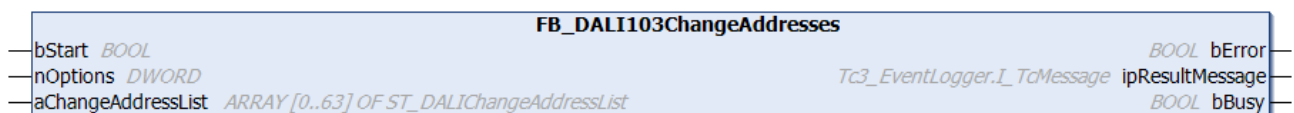
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.1.2.1.2 FB\_DALI103ChangeAddresses**



This function block can be used to change the short addresses of several DALI control devices.

A list of the DALI control devices for which the short address is to be changed is transferred in the array *aChangeAddressList* of type *ST\_DALIChangeAddressList* [▶ 852]. The list has 64 entries from 0 to 63. Each entry contains a variable *nOldAddress* and *nNewAddress* with which the address assignment is parameterized. The end of the list is programmed with a 255 entry at *nOldAddress*, so that the whole list does not necessarily have to be filled in. If this entry is missing, however, then all entries are accepted. When the function block is started (positive edge on *bStart*), the list end is first determined on the basis of the described entry and afterwards the valid list range is examined for the following false entries:

- Address entries > 63
- Double address entry on the source page *nOldAddress*
- Double address entry on the target page *nNewAddress* (leads to double assignment of an address and, hence, to errors)

The function block then uses the short addresses to determine the internal long addresses of the DALI devices and enters them in the *nRandomAddress* parameter of the list.

If an error occurs during queries or during reprogramming, this leads to an error entry for the respective device in the list element *nErrors* (see [ST\\_DALIChangeAddressList](#) [▶ 852]).

The individual bits in the list element *nErrors* have the following meaning:

Bit	Error
0	Error during reading of the long address
1	-
2	-
3	-
4	-
5	Error during programming of the new short address
6	<i>nOldAddress</i> and <i>nNewAddress</i> have the same values



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nOptions        : DWORD := 0;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nOptions	DWORD	Reserved for future extensions.

**Inputs/outputs**

```
VAR_IN_OUT
  aChangeAddressList : ARRAY [0..63] OF ST_DALIChangeAddressList;
END_VAR
```

Name	Type	Description
aChangeAddressList	<a href="#">ST_DALIChangeAddressList</a> [▶ 852]	List of short addresses to be changed.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

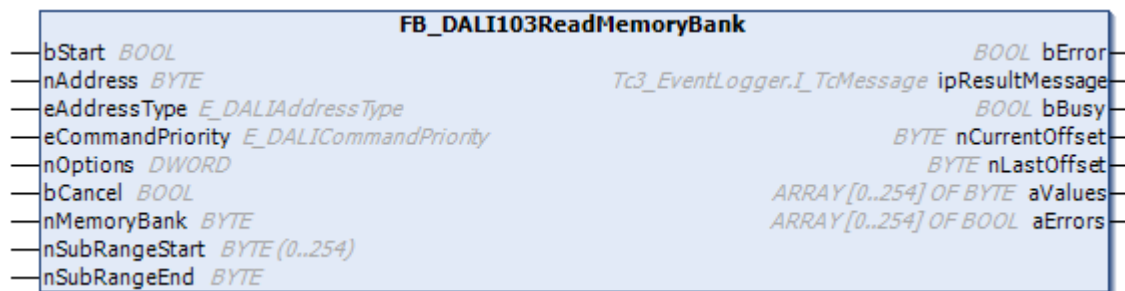
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.1.2.2 Settings**

**4.1.1.2.2.1 FB\_DALI103ReadMemoryBank**



The specified memory bank of the DALI control device is read (see [Memory banks](#) [▶ 16]). The memory bank is specified by the parameter *nMemoryBank*. The range to be read can be limited with the help of the parameters *nSubRangeStart* and *nSubRangeEnd*.



The function block changes the DTR0 and the DTR1 of all connected DALI control devices.



While the function block is active (*bBusy* = TRUE), all function blocks that also have write access to the DTRs of the connected DALI control devices on the same DALI line are blocked.



The function block cannot be used when using the KL6811.



 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nOptions        : DWORD;
  bCancel        : BOOL;
  nMemoryBank     : BYTE;
  nSubRangeStart  : BYTE(0..254) := 2;
  nSubRangeEnd    : BYTE := 255;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nOptions	DWORD	Reserved for future extensions.
bCancel	BOOL	A rising edge at this input disables the function block and aborts the reading of the variables.
nMemoryBank	BYTE	Specifies the memory bank for the read access.
nSubRangeStart	BYTE	Offset inside the memory bank from which reading is to take place.
nSubRangeEnd	BYTE	Offset inside the memory bank up to which reading is to take place.

nSubRangeEnd	Description
255	Offset 0 of the memory bank determines the offset up to which reading is to take place.
254	Offset 0 is not read at the beginning. Reading always takes place up to Offset 254.
<= 253	<i>nSubRangeEnd</i> may not be greater than the value in Offset 0. The partial areas of the memory bank are read with this setting.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nCurrentOffset  : BYTE;
  nLastOffset     : BYTE;
  aValues         : ARRAY [0..254] OF BYTE;
  aErrors         : ARRAY [0..254] OF BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nCurrentOffset	BYTE	The offset that is currently being read is output at this output.
nLastOffset	BYTE	Last valid offset of the memory bank that was read.
aValues	BYTE	Array containing the read parameters of the memory bank.
aErrors	BOOL	Array containing errors of the parameters of the memory bank that occurred during reading.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

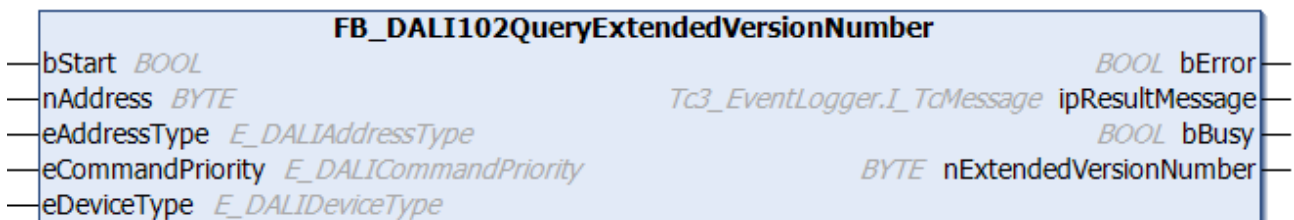
Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2 Commands**

**4.1.2.1 Part 102 (control gears)**

**4.1.2.1.1 Application extended commands**

**4.1.2.1.1.1 FB\_DALI102QueryExtendedVersionNumber**



The function block reads the version number for the implemented, application extended commands from the DALI control gear.

This command belongs to the application extended commands.



Since each DALI control gear can support several device types, the DALI command ENABLE DEVICE TYPE must be called before each application-related extension command. Application extended commands are defined in the IEC 62386-2xx standard.

The function block FB\_DALI102QueryExtendedVersionNumber independently executes the DALI command ENABLE DEVICE TYPE internally.

The input variable *eDeviceType* specifies for which device type the version number is to be read out

The output *nExtendedVersionNumber* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICCommandPriority := E_DALICCommandPriority.MiddleLow;
  eDeviceType     : E_DALIDeviceType;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICCommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eDeviceType	E_DALIDeviceType	Identification of the device type (see <a href="#">E_DALIDeviceType ▶ 857</a> ).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nExtendedVersionNumber : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nExtendedVersionNumber	BYTE	Version number of the selected device type within the DALI control gear.

 **Properties**

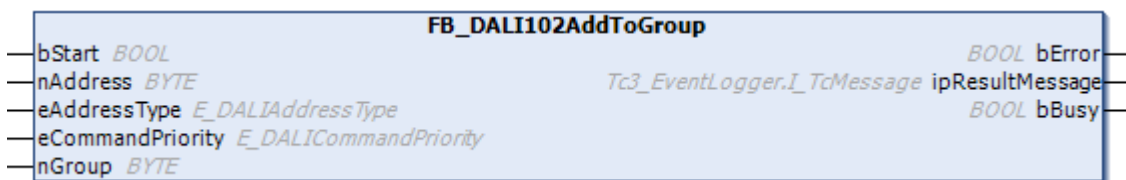
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.1.2 Configuration commands**

**4.1.2.1.2.1 FB\_DALI102AddToGroup**



The function block adds the DALI control gear to the *nGroup* group.

Internally, the group assignment is stored in the variable [gearGroups](#) [▶ 154].



The function block sends the DALI command ADD TO GROUP twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nGroup      : BYTE;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nGroup	BYTE	Group number (0...15).

### 🔌 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

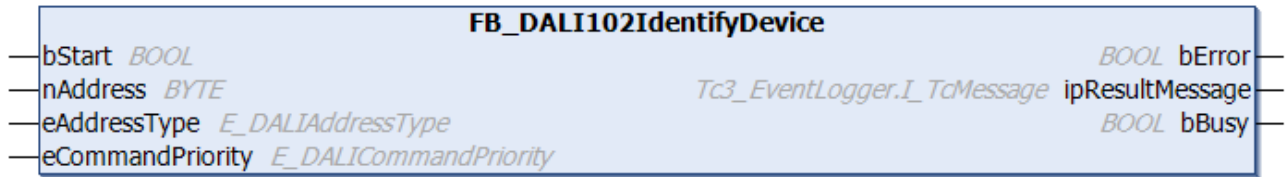
### 📄 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.1.2.2 FB\_DALI102IdentifyDevice



The function block starts the identification routine for the DALI control gear.

It takes approx. 10 seconds and ends automatically. The exact scope of the identification routine depends on the manufacturer of the DALI control gear.



The function block sends the DALI command IDENTIFY DEVICE twice, as required by the DALI standard for certain DALI commands.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

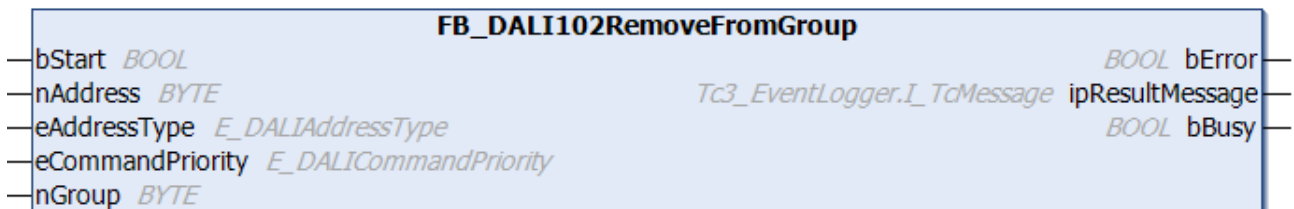
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.2.3 FB\_DALI102RemoveFromGroup**



The function block removes the DALI control gear from the group *nGroup*.

Internally, the group membership is stored in the variable [gearGroups \[▶ 154\]](#).

Each bit of *gearGroups* represents a group. Bit 0 corresponds to group 0 etc., up to bit 15 in group 15. If a bit is set, the DALI control gear belongs to the corresponding group.



The function block sends the DALI command REMOVE FROM GROUP twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nGroup     : BYTE;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nGroup	BYTE	Group number (0...15).

### Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

### Properties

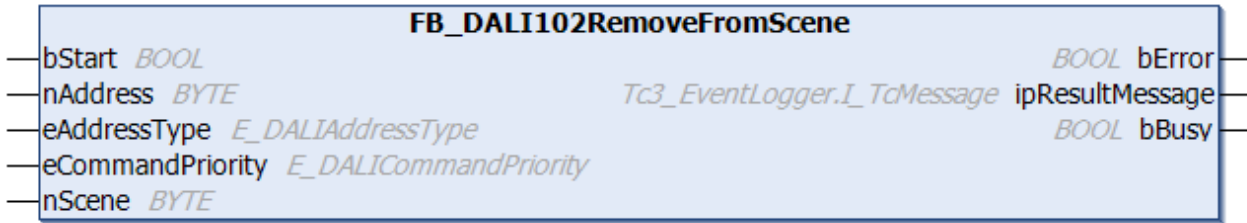
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0



4.1.2.1.2.4 FB\_DALI102RemoveFromScene



The function block removes the DALI control gear from the scene *nScene*.

Internally, the scenes are stored in the variables *scene0* to *scene15* (see [scene0...scene15](#) [▶ 154]).



The function block sends the DALI command REMOVE FROM SCENE twice, as required by the DALI standard for certain DALI commands.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nScene     : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nScene	BYTE	Scene number (0...15).

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.2.5 FB\_DALI102Reset**



The function block resets all variables of the DALI control gear to their default values.

Control gears can take up to 300 ms to execute the DALI command. Only then are further DALI commands executed by the respective DALI control gear.



The function block sends the DALI command RESET twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.1.2.6 FB\_DALI102ResetMemoryBank



The values of the selected memory bank of the DALI control gear are set to their respective default values by the function block (see [Memory banks](#) [▶ 16]).

During the reset, the DALI control gear does not react to other commands.

The function block also resets the values that are write-protected by the lock byte.

The DALI commands ENABLE WRITE MEMORY, WRITE MEMORY LOCATION – NO REPLY and RESET MEMORY BANK are called internally.



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.



The function block sends the DALI command RESET MEMORY BANK twice, as foreseen by the DALI standard for certain DALI commands.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMemoryBank : BYTE := 2;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
MemoryBank	BYTE	Specification of the memory bank. If a 0 is specified, all memory banks are reset (except for memory bank 0, as this is read-only). For all other values, only the respective memory bank is reset.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.1.2.7 FB\_DALI102SavePersistentVariables**



The function block immediately triggers the saving of variables in the persistent memory by the DALI command SAVE PRESISTENT VARIABLES.

DALI control gears do not save the variables directly in the persistent memory. Instead, the values are copied to the persistent memory in the background. Since this happens over a longer period of time (a few seconds), the values can be lost if the DALI bus is disconnected.

The SAVE PRESISTENT VARIABLES DALI command triggers immediate copying to the persistent memory. In most cases, no communication with the DALI device can take place during copying (up to 350 ms).



The function block sends the DALI command SAVE PRESISTENT VARIABLES twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.1.2.8 FB\_DALI102SetExtendedFadeTime



This function block writes the values *eExtendedFadeTimeBase* and *eExtendedFadeTimeMultiplier* to the variables [extendedFadeTimeBase \[▶ 152\]](#) and [extendedFadeTimeMultiplier \[▶ 152\]](#).

The Extended Fade Time is calculated from the product of the two variables *extendedFadeTimeBase* and *extendedFadeTimeMultiplier*.

The value can be set between 100 ms and 16 min. With a value of 0 s, the lighting output reaches the required value as quickly as possible.

The *extendedFadeTime* is used in the function blocks [FB\\_DALI102DirectArcPowerControl \[▶ 82\]](#), [FB\\_DALI102GoToScene \[▶ 87\]](#) and [FB\\_DALI102GoToLastActiveLevel \[▶ 85\]](#).

**i** The two variables *extendedFadeTimeBase* and *extendedFadeTimeMultiplier* are used for the calculation only if [fadeTime \[▶ 151\]](#) was set to *E\_DALIFadeTime.Disabled* and, if implemented, [fastFadeTime \[▶ 330\]](#) is equal to *E\_DALIFastFadeTime.Disabled*.

**i** The function block changes the DTR0 of all connected DALI control gears.

**i** The function block sends the DALI command SET EXTENDED FADE TIME twice, as required by the DALI standard for certain DALI commands.

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eExtendedFadeTimeBase : E_DALIExtendedFadeTimeBase := E_DALIExtendedFadeTimeBase.Base01;
  eExtendedFadeTimeMultiplier : E_DALIExtendedFadeTimeMultiplier := E_DALIExtendedFadeTimeMultiplier.NoFade;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eExtendedFadeTime Base	E_DALIExtendedFadeTime Base [▶ 859]	Basis for calculating the Extended FadeTime.
eExtendedFadeTime Multiplier	E_DALIExtendedFadeTime Multiplier [▶ 860]	Time multiplier for calculating Extended FadeTime.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

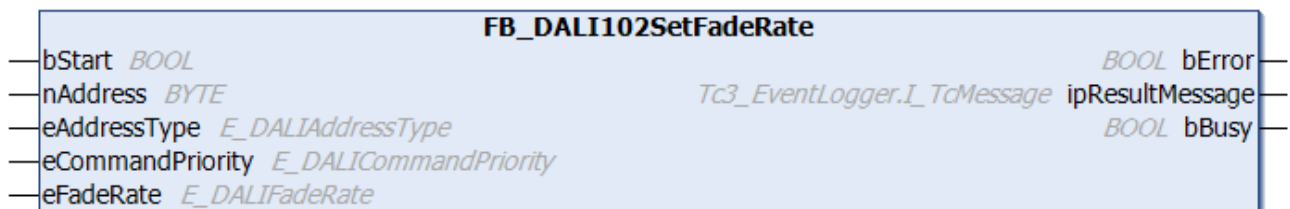
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

**4.1.2.1.2.9 FB\_DALI102SetFadeRate**



The function block sets the [fadeRate \[▶ 150\]](#) to the value *eFadeRate*.

The *fadeRate* determines the rate of change, in steps per second, of the output value.

The *fadeRate* is used in the function blocks [FB\\_DALI102Down \[▶ 84\]](#) and [FB\\_DALI102Up \[▶ 97\]](#).





The function block changes the DTR0 of all connected DALI control gears.



The function block sends the DALI command SET FADE RATE twice, as required by the DALI standard for certain DALI commands.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eFadeRate      : E_DALIFadeRate := E_DALIFadeRate.N045StepsPerSec;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eFadeRate	E_DALIFadeRate	Fade rate for the change of the output value (see E_DALIFadeRate [▶ 860]).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see error evaluation [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see runtime messages [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

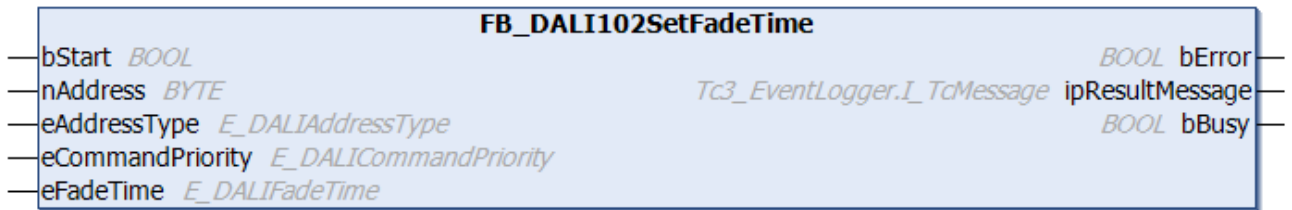
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.0	Tc3_DALI from v3.1.5.0

**4.1.2.1.2.10 FB\_DALI102SetFadeTime**



The function block sets the [fadeTime](#) [[▶ 151](#)] to the value *eFadeTime*.

The *fadeTime* defines the time the current output value takes to reach the required value.

The *fadeTime* is used in the function blocks [FB\\_DALI102DirectArcPowerControl](#) [[▶ 82](#)], [FB\\_DALI102GoToScene](#) [[▶ 87](#)] and [FB\\_DALI102GoToLastActiveLevel](#) [[▶ 85](#)].



The function block changes the DTR0 of all connected DALI control gears.



The function block sends the DALI command SET FADE TIME twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eFadeTime      : E_DALIFadeTime := E_DALIFadeTime.Disabled;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [ <a href="#">▶ 854</a> ]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [ <a href="#">▶ 856</a> ]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eFadeTime	E_DALIFadeTime	Fade time for the change of the output value (see <a href="#">E_DALIFadeTime [► 860]</a> ).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

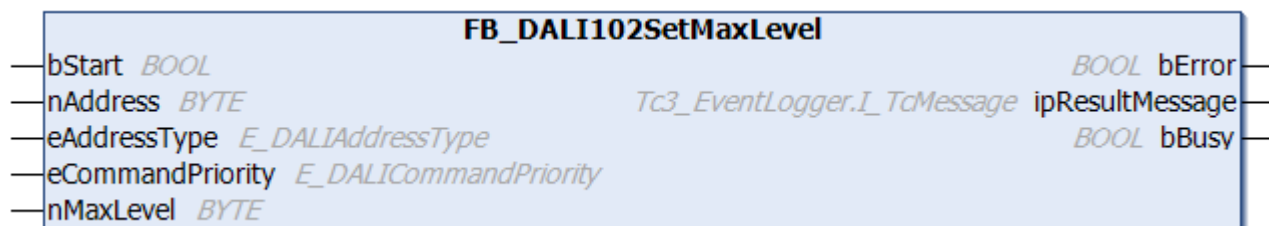
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.0	Tc3_DALI from v3.1.5.0

**4.1.2.1.2.11 FB\_DALI102SetMaxLevel**



The function block stores the value *nMaxLevel* in the variable [maxLevel \[► 150\]](#) of the DALI control gear.

If a value smaller than *minLevel* is specified, the value is set to *minLevel*.

If the value 255 (MASK) is specified, *maxLevel* is automatically set to 254.



The function block changes the DTR0 of all connected DALI control gears.



The function block sends the DALI command SET MAX LEVEL twice, as required by the DALI standard for certain DALI commands.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMaxLevel       : BYTE := 254;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMaxLevel	BYTE	Maximum allowed output value ( <i>minLevel</i> ...254).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.2.12 FB\_DALI102SetMinLevel



The function block stores the value *nMinLevel* in the variable `minLevel` [[► 150](#)] of the DALI control gear.

If a value greater than *maxLevel* or 255 (MASK) is specified, the value is set to *maxLevel*.

If a value smaller than *physicalMinimum* is specified, the value is set to *physicalMinimum*.



The function block changes the DTR0 of all connected DALI control gears.



The function block sends the DALI command SET MIN LEVEL twice, as required by the DALI standard for certain DALI commands.

 Inputs

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMinLevel       : BYTE := 128;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [ <a href="#">► 854</a> ]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [ <a href="#">► 856</a> ]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMinLevel	BYTE	Minimum allowed output value ( <i>physicalMinimum...maxLevel</i> ).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

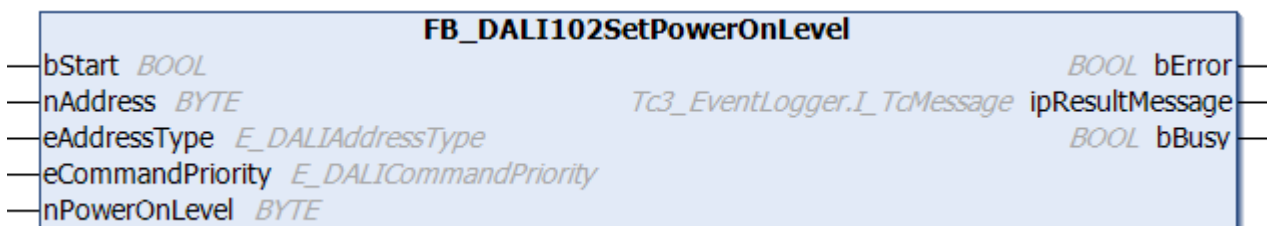
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.2.13 FB\_DALI102SetPowerOnLevel**



The function block stores the value *nPowerOnLevel* in the variable [powerOnLevel \[▶ 150\]](#) of the DALI control gear.



The function block changes the DTR0 of all connected DALI control gears.



The function block sends the DALI command SET POWER ON LEVEL twice, as required by the DALI standard for certain DALI commands.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nPowerOnLevel   : BYTE := 254;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nPowerOnLevel	BYTE	Switch-on value (0...255).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [►_892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.2.14 FB\_DALI102SetScene**



The function block stores the value *nSceneLevel* in the specified scene *nScene* of the DALI control gear.

If the value is greater than *maxLevel* and less than 255 (MASK), the value *maxLevel* is output when the scene is called.

If a value less than *minLevel* is specified, the value *minLevel* is output when the scene is called.

Internally, the scenes are stored in the variables *scene0* to *scene15* (see [scene0...scene15 \[►\\_154\]](#)).



The function block changes the DTR0 of all connected DALI control gears.



The function block sends the DALI command SET SCENE twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nSceneLevel : BYTE := 255;
  nScene      : BYTE := 0;
END_VAR
  
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nSceneLevel	BYTE	Output value of the desired scene (0...255).
nScene	BYTE	Scene whose output value is to be changed (0...15).

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

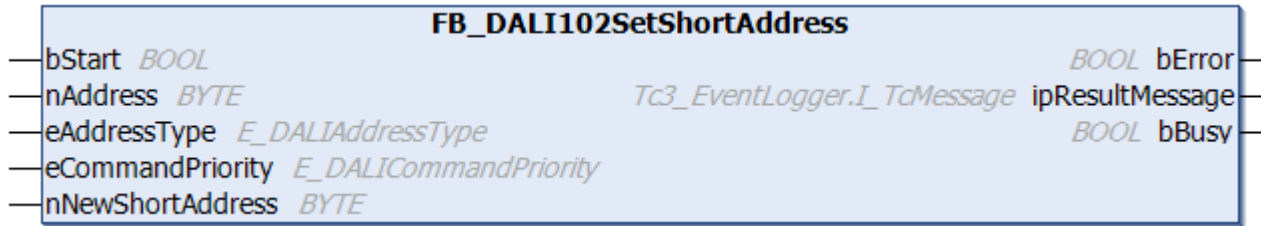
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.1.2.15 FB\_DALI102SetShortAddress



The function block stores the new short address *nNewShortAddress* in the shortAddress [▶ 153] variable of the DALI control gear.

The existing short address in the DALI control gear is deleted if a value of 255 is entered at the input.

To address a device without a known short address, the command must be sent as a broadcast (*eAddressType* = *E\_DALIAddressType.Broadcast*). This gives all DALI control gears connected to the DALI terminal the short address *nNewShortAddress*.

Alternatively, the function block can be called with Broadcast unaddressed (*eAddressType* = *E\_DALIAddressType.BroadcastUnaddr*). In this case, only the DALI devices that do not yet have a short address are assigned a short address.



The function block changes the DTR0 of all connected DALI control gears.



The function block sends the DALI command SET SHORT ADDRESS twice, as required by the DALI standard for certain DALI commands.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nNewShortAddress : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<u>E_DALIAddressType</u> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices ( <i>BroadcastUnaddr</i> ) has been selected.
eCommandPriority	<u>E_DALICommandPriority</u> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nNewShortAddress	BYTE	New short address (0...63, 255).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

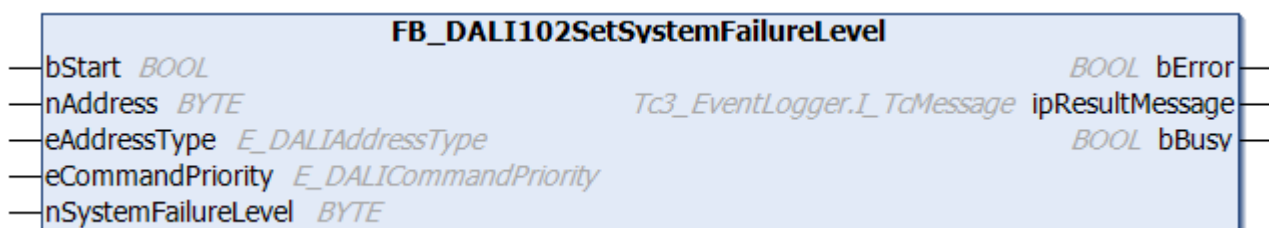
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.2.16 FB\_DALI102SetSystemFailureLevel**



The function block stores the value *nSystemFailureLevel* in the variable [systemFailureLevel \[▶ 150\]](#) of the DALI control gear.



The function block changes the DTR0 of all connected DALI control gears.



The function block sends the DALI command SET SYSTEM FAILURE LEVEL twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nSystemFailureLevel : BYTE := 254;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nSystemFailureLevel	BYTE	Output value in the case of a system failure (0...255).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.2.17 FB\_DALI102StoreActualLevelInDTR0



The function block writes the current output value of the DALI control gear to DTR0 [▶ 154].



The function block sends the DALI command STORE ACTUAL LEVEL IN DTR0 twice, as required by the DALI standard for certain DALI commands.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.3 Control commands**

**4.1.2.1.3.1 FB\_DALI102DirectArcPowerControl**



The function block switches the DALI control gear to the default output value *nArcPowerLevel*.

If *nArcPowerLevel* is outside the specified variable [maxLevel \[▶ 150\]](#) or [minLevel \[▶ 150\]](#) of the DALI control gear, it is set to the corresponding lowest or highest value. The DALI control gear is switched on by the command if it was previously off.

If the value *nArcPowerLevel* = 0, the DALI control gear is switched off and if *nArcPowerLevel* = 255 (MASK), the DALI control gear retains its current value.

The speed at which the specified value is to be reached is specified by the variable [fadeTime \[▶ 151\]](#).

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nArcPowerLevel  : BYTE;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nArcPowerLevel	BYTE	Output value for the DALI control gear (0...255).

### 🔌 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

### 📄 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.1.3.2 FB\_DALI102Down



The output value of the DALI control gear is reduced over the specified period of 200 ms.

The variable `fadeRate` [▶ 150] determines the number of steps by which the output value is changed within the 200 ms.

If the current output value is already at the `minLevel` [▶ 150] value, the output value remains unchanged.

This command does not switch off the DALI control gear.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

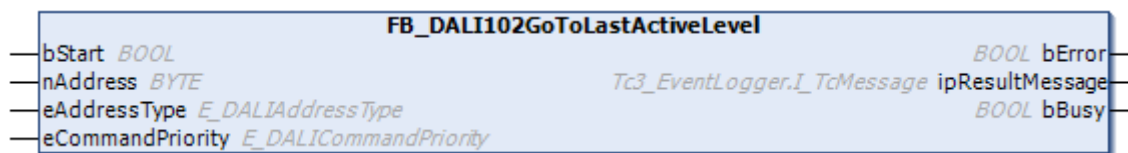
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.3.3 FB\_DALI102GoToLastActiveLevel**



If the output value of the DALI control gear is 0, it is set to the last active output value (*minLevel...maxLevel*) that was used before switching off. This function block is used to reset DALI control gears that have been switched off to their last output value (switched on).

If the output value of the DALI control gear is greater than 0, the output value remains unchanged.

Directly after applying the supply voltage to the DALI control gear, this function block sets the output value to *maxLevel*.

The variable [fadeTime \[▶ 151\]](#) defines the speed at which the last active output value is to be reached.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

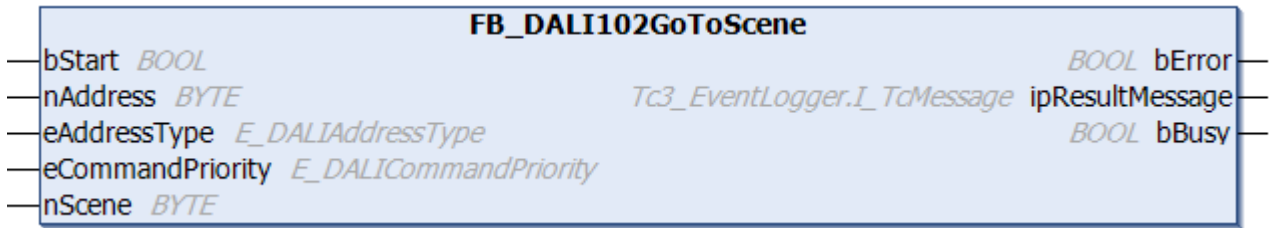
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

4.1.2.1.3.4 FB\_DALI102GoToScene



The output value of the DALI control gear is set to the value stored in *nScene*.

The variable `fadeTime` [▶ 151] determines the speed at which the specified value of the scene is to be reached.

If the value of the selected scene in the DALI control gear is 255, the output of the DALI control gear retains its current value.

If the DALI control gear is switched off and the value stored in *nScene* is greater than 0 and less than 255, the DALI control gear is switched on with this command.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nScene      : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nScene	BYTE	Scene to be called in the DALI control gear (0...15).

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.3.5 FB\_DALI102Off**



The DALI control gear is switched off immediately.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

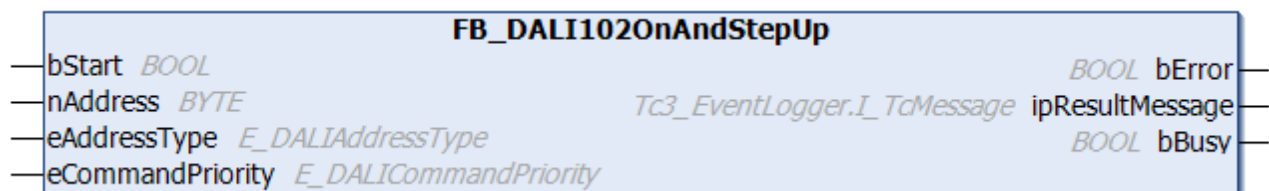
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.3.6 FB\_DALI102OnAndStepUp**



The output value of the DALI control gear is increased by one step.

If the DALI control gear is off, the command switches it on, and the output value is set to the stored value [minLevel](#) [▶ 150].

If the output value has already reached the [maxLevel](#) [▶ 150] value, it retains its current value.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.3.7 FB\_DALI102RecallMaxLevel



The output value of the DALI control gear is set to `maxLevel` [▶ 150].

If the DALI control gear is switched off, this command switches it on.

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

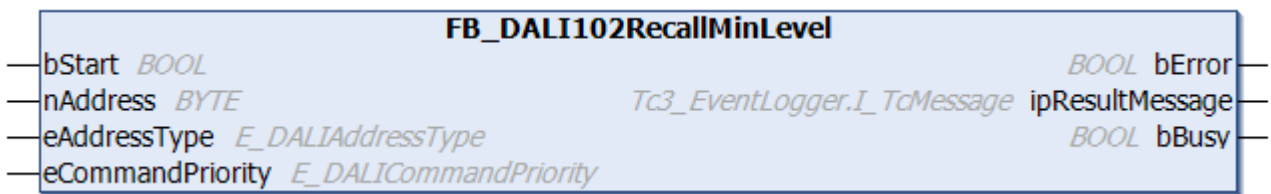
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.3.8 FB\_DALI102RecallMinLevel**



The output value of the DALI control gear is set to [minLevel \[► 150\]](#) value.

If the DALI control gear is switched off, this command switches it on.

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType [► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority [► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.3.9 FB\_DALI102StepDown**



The output value of the DALI control gear is reduced by one step.

If the output of the DALI control gear is already at the [minLevel \[▶ 150\]](#) value, the value is not reduced further.

The command does not switch off the DALI control gear.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.3.10 FB\_DALI102StepDownAndOff



The output value of the DALI control gear is reduced by one step.

If the output of the DALI control gear is already at the `minLevel` [[▶ 150](#)] value, it is switched off.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.3.11 FB\_DALI102StepUp**



The output value of the DALI control gear is increased by one step.

If the output value of the DALI control gear is already at the maxLevel [[► 150](#)] value, the value remains unchanged.

The command does not switch on the DALI control gear.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [ <a href="#">► 854</a> ]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [ <a href="#">► 856</a> ]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.3.12 FB\_DALI102Up**



The output value of the DALI control gear is increased over the specified period of 200 ms.

The variable [fadeRate \[▶ 150\]](#) determines the number of steps by which the output value is changed within the 200 ms.

If the current output value is already at the [maxLevel \[▶ 150\]](#) value, the output value remains unchanged.

This command does not switch on the DALI control gear.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

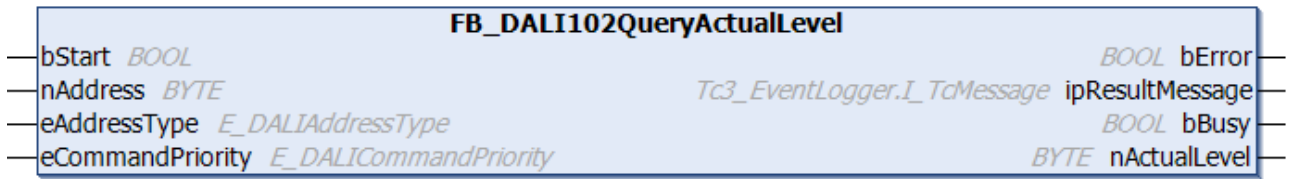
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.4 Query commands

4.1.2.1.4.1 FB\_DALI102QueryActualLevel



The function block reads the current output value of the DALI control gear (see [actualLevel \[▶ 150\]](#)).

The current output value is 0 or lies within the value range from [minLevel \[▶ 150\]](#) to [maxLevel \[▶ 150\]](#).

If the value 255 (MASK) is returned by the DALI control gear, *bError* is set to TRUE. In this case the DALI control gear is in the start-up phase or there is a technical defect.

The output *nActualValue* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType [▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority [▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nActualLevel : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nActualLevel	BYTE	Current output value (0, <i>minLevel</i> ... <i>maxLevel</i> ).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.4.2 FB\_DALI102QueryContentDTR0**



The function block reads the contents of [DTR0](#) [▶ 154] (Data Transfer Register 0) from the DALI control gear.

The output *nContentDTR0* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 🔌 Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nContentDTR0    : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nContentDTR0	BYTE	Content of the DTR0 (Data Transfer Register 0) (0...255).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.1.4.3 FB\_DALI102QueryContentDTR1



The function block reads the contents of [DTR1 \[► 154\]](#) (Data Transfer Register 1) from the DALI control gear.

The output *nContentDTR1* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType [► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority [► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nContentDTR1    : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nContentDTR1	BYTE	Content of the DTR1 (Data Transfer Register 1) (0...255).

 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.4.4 FB\_DALI102QueryContentDTR2



The function block reads the contents of [DTR2 \[► 154\]](#) (Data Transfer Register 2) from the DALI control gear.

The output *nContentDTR2* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nContentDTR2    : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nContentDTR2	BYTE	Content of the DTR2 (Data Transfer Register 1) (0...255).

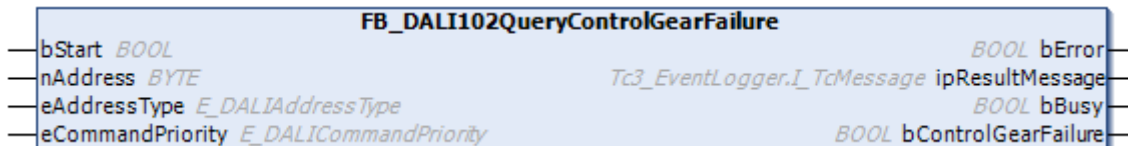
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.4.5 FB\_DALI102QueryControlGearFailure**



The function block checks whether there is a defect in one or more DALI control gears.

This function block can be used to determine whether there are defective DALI control gears on a DALI line. For this purpose, the function block is called with the parameter *eAddressType* = *E\_DALIAddressType.Broadcast*. If the outputs *bControlGearFailure* = FALSE and *bError* = FALSE, there is no defect in any DALI control gear. If the output *bControlGearFailure* = TRUE and *bError* = FALSE, a DALI control gear is defective. If multiple DALI control gears are defective, a 27 (multiple DALI control gears have responded) is returned to *ipResultMessage.nEventId*.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bBusy            : BOOL;
  bControlGearFailure : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
ControlGearFailure	BOOL	The output is set as soon as a defect has been detected in the queried DALI control gear.



**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

### 4.1.2.1.4.6 FB\_DALI102QueryControlGearPresent

FB_DALI102QueryControlGearPresent		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BOOL bControlGearPresent

The function block indicates whether the required DALI control gear can be reached via its short address.

The output *bControlGearPresent* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType ▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority ▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  bControlGearPresent : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see error evaluation ▶ 887]) that can be used to obtain detailed information about the processing of the function block (see runtime messages ▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bControlGearPresent	BOOL	The output is set as soon as the queried DALI control gear is available.

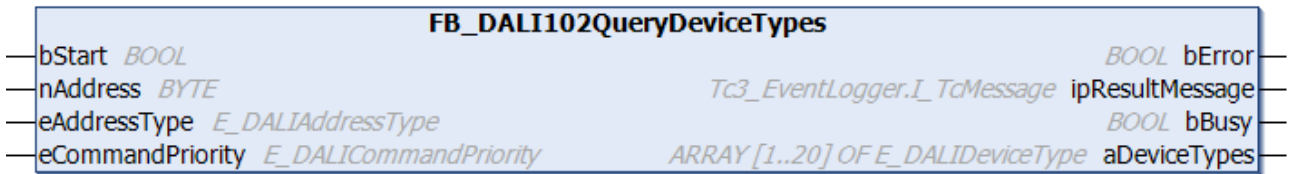
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.4.7 FB\_DALI102QueryDeviceTypes



The function block returns a list of all device types supported by the DALI control gear.

The DALI commands QUERY DEVICE TYPE and QUERY NEXT DEVICE TYPE are called internally.

As long as the function block is active (*bBusy* = TRUE), no further DALI commands may be sent. Otherwise, the list of supported device types may not be read completely.

The output *aDeviceTypes* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType [► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority [► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.



 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  aDeviceTypes   : ARRAY [1..20] OF E_DALIDeviceType;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
aDeviceTypes	E_DALIDeviceType	List of supported device types (see <a href="#">E_DALIDeviceType</a> [▶ 857]).

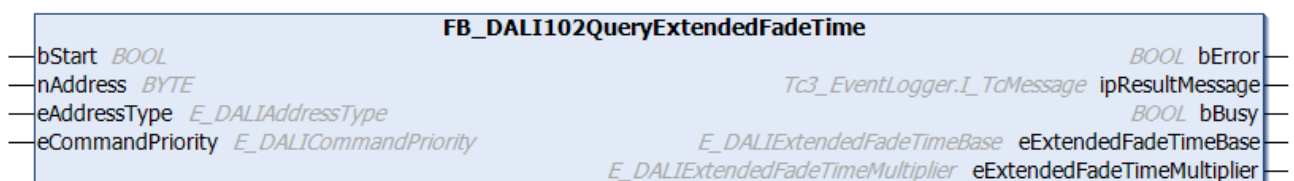
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.1.4.8 FB\_DALI102QueryExtendedFadeTime**



This function block reads the variables [extendedFadeTimeBase](#) [▶ 152] and [extendedFadeTimeMultiplier](#) [▶ 152] from the DALI control gear. The Extended Fade Time (100 ms to 16 min) can be calculated from the product of the two variables.

The *extendedFadeTime* is used in the function blocks [FB\\_DALI102DirectArcPowerControl](#) [▶ 82], [FB\\_DALI102GoToScene](#) [▶ 87] and [FB\\_DALI102GoToLastActiveLevel](#) [▶ 85].

**i** The two variables *extendedFadeTimeBase* and *extendedFadeTimeMultiplier* are used for the calculation only if [fadeTime](#) [▶ 151] was set to *E\_DALIFadeTime.Disabled* and, if implemented, [fastFadeTime](#) [▶ 330] is equal to *E\_DALIFastFadeTime.Disabled*.



The outputs *eExtendedFadeTimeBase* and *eExtendedFadeTimeMultiplier* only contain a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  eExtendedFadeTimeBase : E_DALIExtendedFadeTimeBase;
  eExtendedFadeTimeMultiplier : E_DALIExtendedFadeTimeMultiplier;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eExtendedFadeTimeBase	E_DALIExtendedFadeTimeBase [▶ 859]	Basis for calculating the Extended Fade Time
eExtendedFadeTimeMultiplier	E_DALIExtendedFadeTimeMultiplier [▶ 860]	Time multiplier for calculating the Extended Fade Time

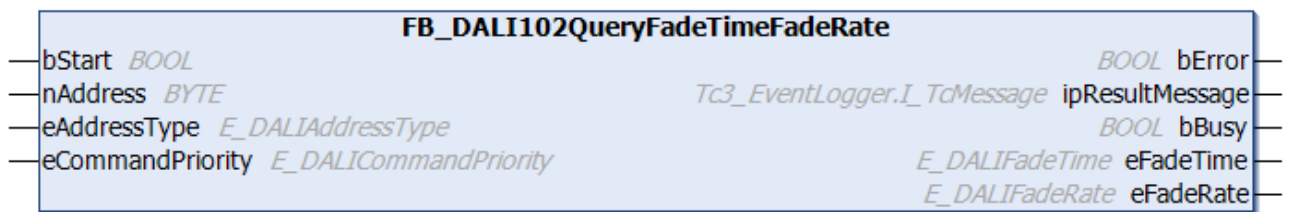
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

**4.1.2.1.4.9 FB\_DALI102QueryFadeTimeFadeRate**



The function block reads the variables `fadeTime` [▶ 151] and `fadeRate` [▶ 150] from the DALI control gear.

The `fadeTime` is used in the function blocks `FB_DALI102DirectArcPowerControl` [▶ 82], `FB_DALI102GoToScene` [▶ 87] and `FB_DALI102GoToLastActiveLevel` [▶ 85], while `fadeRate` is used with `FB_DALI102Up` [▶ 97] and `FB_DALI102Down` [▶ 84].

The outputs `eFadeTime` and `eFadeRate` only contain a valid value if the function block was executed without errors (`bError` = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  eFadeTime       : E_DALIFadeTime;
  eFadeRate       : E_DALIFadeRate;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eFadeTime	E_DALIFadeTime	Fade time for the change of the output value (see <a href="#">E_DALIFadeTime</a> [▶ 860])
eFadeRate	E_DALIFadeRate	Fade rate for the change of the output value (see <a href="#">E_DALIFadeRate</a> [▶ 860])

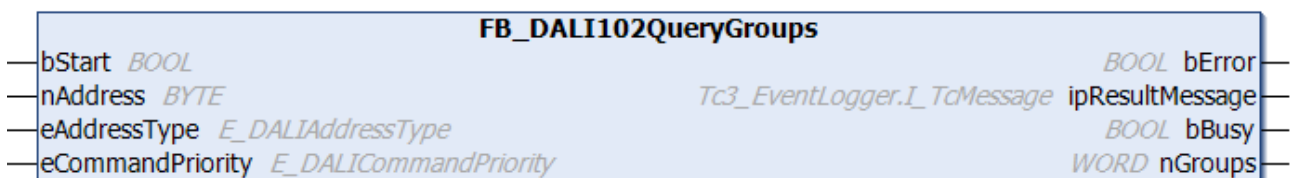
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.0	Tc3_DALI from v3.1.5.0

**4.1.2.1.4.10 FB\_DALI102QueryGroups**



The function block reads the variable [gearGroups](#) [▶ 154] from the DALI control gear.

Each bit represents one group. Bit 0 (group 0) to bit 15 (group 15). If the bit is set, the DALI control gear belongs to the corresponding group.

The DALI commands QUERY GROUPS 0-7 and QUERY GROUPS 8-15 are called internally.

The output *nGroups* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nGroups        : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nGroups	WORD	Group assignment

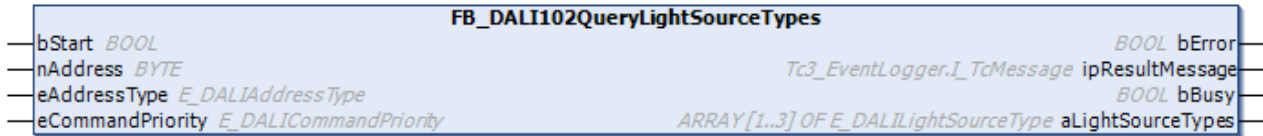
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.4.11 FB\_DALI102QueryLightSourceTypes



The function block returns a list of all light source types supported by the DALI control gear.

Internally the DALI commands QUERY LIGHT SOURCE TYPE, QUERY CONTENT DTR0, QUERY CONTENT DTR1 and QUERY CONTENT DTR2 are called.

The output *aLightSourceTypes* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

Inputs

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```

VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  aLightSourceTypes : ARRAY [1..3] OF E_DALILightSourceType;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
aLightSourceTypes	E_DALILightSourceType	List of supported light source types (see <a href="#">E_DALILightSourceType</a> [▶ 862]).

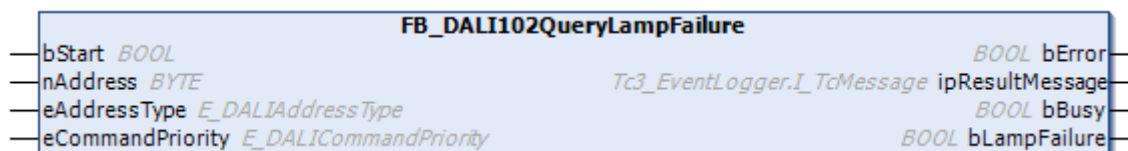
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

**4.1.2.1.4.12 FB\_DALI102QueryLampFailure**



The function block checks whether there is a defect in the lamp of a DALI control gear.

This function block can be used to determine whether there are DALI control gears with defective lamps on a DALI line. For this purpose, the function block is called with the parameter *eAddressType* = *E\_DALIAddressType.Broadcast*. If the outputs *bLampFailure* = FALSE and *bError* = FALSE, no defective lamp was detected on any DALI control gear. If the output *bLampFailure* = TRUE and *bError* = FALSE, the lamp is defective in a DALI control gear. If the lamps are defective at several DALI control gears, a 27 (several DALI control gears have responded) is returned to *ipResultMessage.nEventId*.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bLampFailure    : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bLampFailure	BOOL	The output is set as soon as a defective lamp has been detected at the queried DALI control gear.

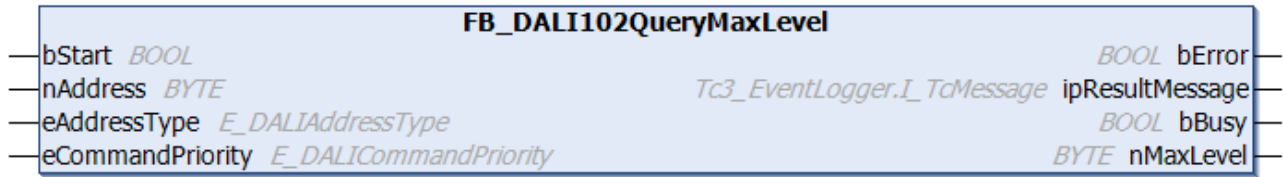
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

### 4.1.2.1.4.13 FB\_DALI102QueryMaxLevel



The function block reads the variable `maxLevel` [▶ 150] from the DALI control gear.

The output `nMaxLevel` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nMaxLevel   : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <code>ipResultMessage</code> . The output is set to FALSE again as soon as <code>bBusy</code> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see error evaluation [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see runtime messages [▶ 873]). The interface pointer is valid after <code>bBusy</code> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nMaxLevel	BYTE	Maximum allowed output value ( <code>minLevel...254</code> ).



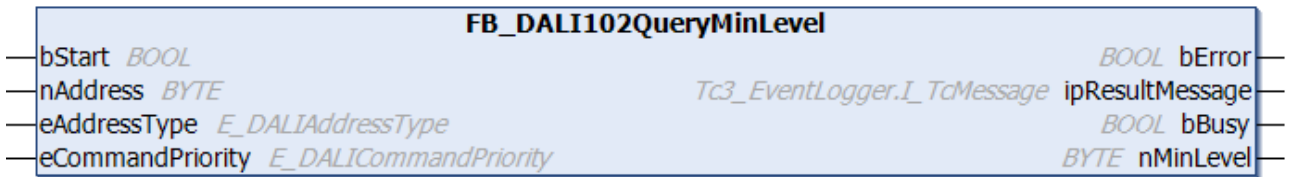
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.4.14 FB\_DALI102QueryMinLevel



The function block reads the variable `minLevel` [► 150] from the DALI control gear.

The output `nMinLevel` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nMinLevel   : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nMinLevel	BYTE	Minimum allowed output value ( <i>physicalMinimum... maxLevel</i> ).

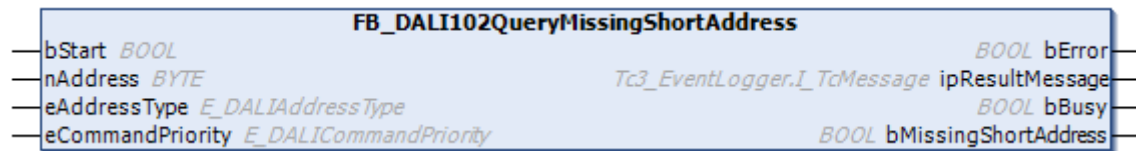
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.4.15 FB\_DALI102QueryMissingShortAddress**



The function block checks whether one or more DALI control gears do not have a short address.

This function block can be used to determine whether there are DALI control gears without short address on a DALI line (e.g. new DALI control gears). For this purpose, the function block is called with the parameter *eAddressType* = *E\_DALIAddressType.Broadcast*. If the outputs *bMissingShortAddress* = FALSE and *bError* = FALSE, all DALI control gears have a short address. If the output *bMissingShortAddress* = TRUE and *bError* = FALSE, there is one DALI control gear that has no short address. If multiple DALI control gears do not have a short address, a 27 (multiple DALI control gears have responded) is returned to *ipResultMessage.nEventId*.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bMissingShort Address : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bMissingShortAddress	BOOL	The output is set as soon as the queried DALI control gear does not have a short address.

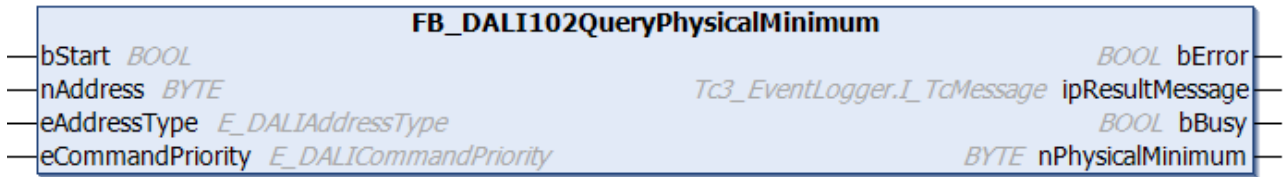
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

### 4.1.2.1.4.16 FB\_DALI102QueryPhysicalMinimum



The function block reads the variable `physicalMinimum` [► 154] from the DALI control gear.

The value is read-only and is fixed by the manufacturer.

The output `nPhysicalMinimum` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nPhysicalMinimum : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nPhysicalMinimum	BYTE	Physically smallest possible output value (1...254).

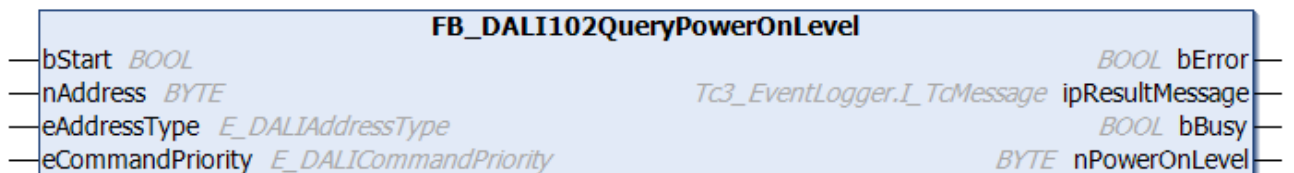
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.4.17 FB\_DALI102QueryPowerOnLevel**



The function block reads the variable [powerOnLevel](#) [▶ 150] from the DALI control gear.

The output *nPowerOnLevel* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bBusy            : BOOL;
  nPowerOnLevel    : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nPowerOnLevel	BYTE	Switch-on value (0...255).

### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.4.18 FB\_DALI102QueryRandomAddress



The function block reads the variable `randomAddress` [► 153] from the DALI control gear.

The DALI commands QUERY RANDOM ADDRESS (H), QUERY RANDOM ADDRESS (M) and QUERY RANDOM ADDRESS (L) are called internally.

The output `nRandomAddress` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nRandomAddress : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRandomAddress	UDINT	Random address (16#00_00_00...16#FF_FF_FE).

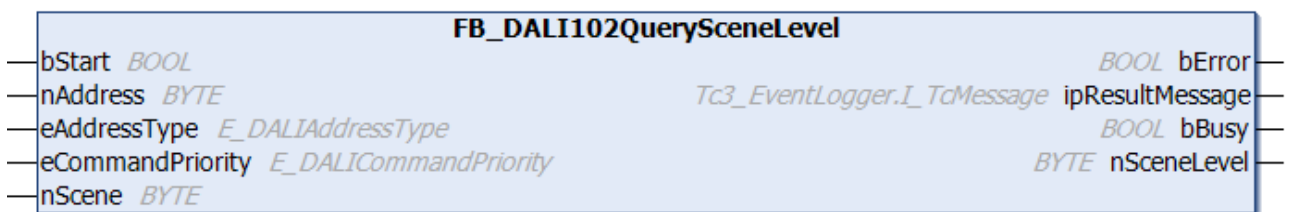
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.4.19 FB\_DALI102QuerySceneLevel**



The function block reads the output value from the DALI control gear for the specified scene *nScene*.

Internally, the scenes are stored in the variables *scene0* to *scene15* (see [scene0...scene15](#) [▶ 154]).

The output *nSceneLevel* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nScene     : BYTE := 0;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nScene	BYTE	Scene whose output value is to be read (0...15).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nSceneLevel    : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nSceneLevel	BYTE	Output value of the desired scene (0...255)

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.1.4.20 FB\_DALI102QueryStatus



The function block reads the status of the DALI control gear.

The status contains the eight most important status messages of a DALI control gear. The meaning of the bits is defined as follows:

Bit	Description
0	Fault of the DALI control gear (see <a href="#">controlGearFailure</a> [▶ 153])
1	Lamp failure (see <a href="#">lampFailure</a> [▶ 153])
2	Lamp power ON (see <a href="#">lampOn</a> [▶ 153])
3	Limit value error (see <a href="#">limitError</a> [▶ 153])
4	Dimming is active (see <a href="#">fadeRunning</a> [▶ 153])
5	Reset state (see <a href="#">resetState</a> [▶ 154])
6	Short address missing (variable <a href="#">shortAddress</a> [▶ 153] is 255 (MASK))
7	The power supply was enabled (see <a href="#">powerCycleSeen</a> [▶ 154])

The output `nStatus` only contains a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
<code>bStart</code>	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
<code>nAddress</code>	BYTE	Address of a DALI control gear or a DALI group.
<code>eAddressType</code>	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
<code>eCommandPriority</code>	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nStatus     : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nStatus	BYTE	Status information (see table above).

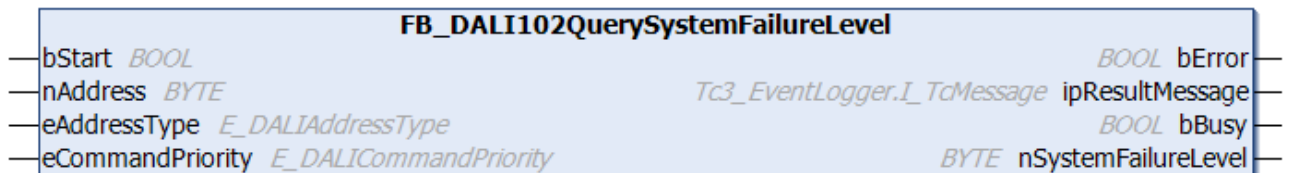
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.4.21 FB\_DALI102QuerySystemFailureLevel**



The function block reads the variable *systemFailureLevel* [▶ 150] from the DALI control gear.

The output *nSystemFailureLevel* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

## Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nSystemFailureLevel : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nSystemFailureLevel	BYTE	Output value in the case of a system failure (0...255).



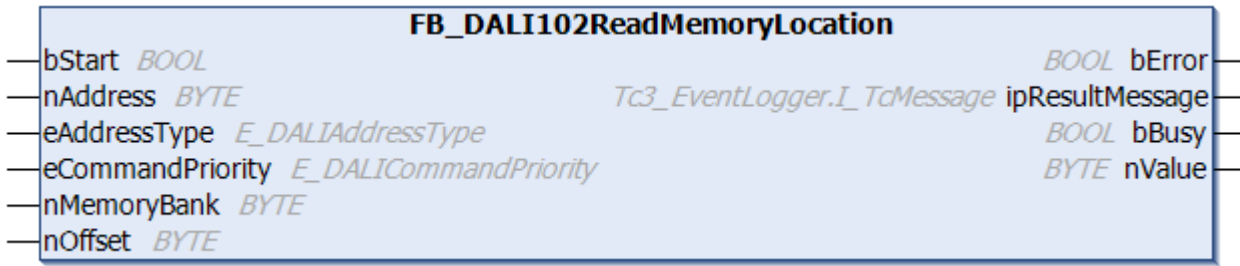
## Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

## Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.4.22 FB\_DALI102ReadMemoryLocation



One byte is read from the memory bank of the DALI control gear. The memory bank is specified by the parameter *nMemoryBank* and the address within the memory bank by the parameter *nOffset*.



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.

The output *nValue* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMemoryBank     : BYTE;
  nOffset         : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMemoryBank	BYTE	Specifies the memory bank for the read access (DTR1).
nOffset	BYTE	Address from which a value within the memory bank is to be read (DTR0).

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nValue          : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nValue	BYTE	Value read from the memory bank.

 **Properties**

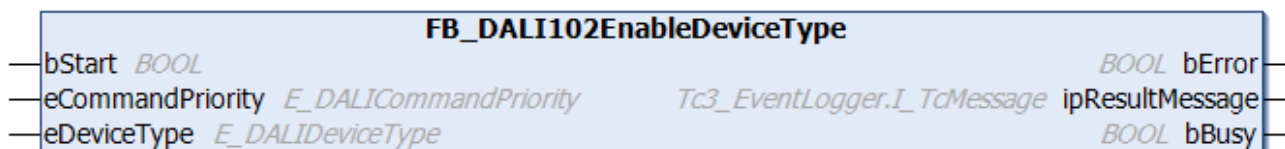
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

**4.1.2.1.5 Special commands**

**4.1.2.1.5.1 FB\_DALI102EnableDeviceType**



The function block must be called before an application extended command is called. This means that only the DALI control gears belonging to the corresponding device type respond.

It is not necessary to use the function block for device type 0. It is also not required if a function block is called that represents an application extended command in the Tc3\_DALI library. The DALI command ENABLE DEVICE TYPE is already executed internally in the corresponding function blocks. There is a corresponding note in the description of the respective function blocks.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eDeviceType     : E_DALIDeviceType;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority [▶ 856]</a> ).



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eDeviceType	E_DALIDeviceType	Selection of the desired device type (see <a href="#">E_DALIDeviceType [▶ 857]</a> ).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.1.5.2 FB\_DALI102Initialise**



The function block prepares the addressing of the DALI control gears.

This function block is required if addressing of the DALI control gears is to be implemented in the PLC. The function block [FB\\_DALI102Addressing \[▶ 21\]](#) can be used to assign the addresses from the PLC.

*nParameter* is used to define which DALI control gears are taken into account for addressing:

Value (binary)	Description
2#0000_0000	All DALI control gears respond.
2#0AAA_AAA1	DALI control gears with the address 2#00AA_AAAA respond.
2#1111_1111	DALI control gears without a short address respond.



The function block sends the DALI command INITIALIZE twice, as required by the DALI standard for certain DALI commands.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nParameter      : BYTE := 2#1111_1111;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority [▶ 856]</a> ).



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nParameter	BYTE	Specifies which control gears should react to this command (see table above)

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.5.3 FB\_DALI102ProgramShortAddress



The function block stores the value *nShortAddress* as a short address in the variable *shortAddress* [► 153] for all selected DALI control gears.


DALI control gears are selected if the *randomAddress* of the DALI control gear is the same as the *searchAddress* and the DALI command INITIALIZE (see [FB\\_DALI102Initialise \[► 131\]](#)) was executed previously.

This function block is required if addressing of the DALI control gears is to be implemented in the PLC. The function block [FB\\_DALI102Addressing \[► 21\]](#) can be used to assign the addresses from the PLC.

 Inputs

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nShortAddress   : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority [► 856]</a> ).

 *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nShortAddress	BYTE	Short address to be assigned (0...63, 255)

 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

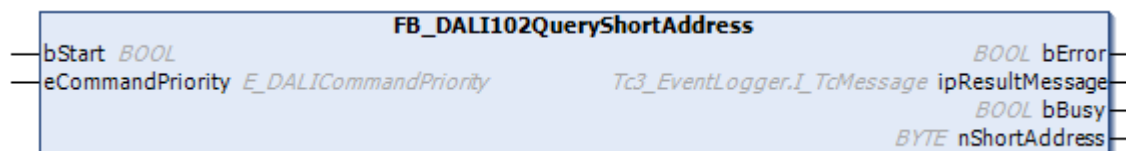
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.5.4 FB\_DALI102QueryShortAddress**



The function block reads the short address ([shortAddress](#) [▶ 153]) of the selected DALI control gear.

DALI control gears are selected if the [randomAddress](#) [▶ 153] of the DALI control gear is the same as the [searchAddress](#) [▶ 153] and the DALI command INITIALIZE (see [FB\\_DALI102Initialise](#) [▶ 131]) was executed previously.

The output *nShortAddress* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority</a> [▶ 856]).



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nShortAddress   : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nShortAddress	BYTE	The output returns the short address (0...63, 255) from the selected DALI control gear.



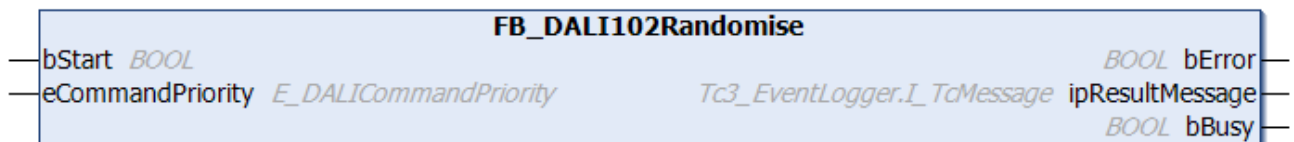
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

**4.1.2.1.5.5 FB\_DALI102Randomise**



The function block generates a new random address for all selected DALI control gears and writes it to the [randomAddress \[▶ 153\]](#) variable.

DALI control gears are selected if the DALI command INITIALIZE (see [FB\\_DALI102Initialise \[▶ 131\]](#)) was executed previously.

This function block is required if addressing of the DALI control gears is to be implemented in the PLC. The function block [FB\\_DALI102Addressing \[▶ 21\]](#) can be used to assign the addresses from the PLC.



The function block sends the DALI command RANDOMIZE twice, as required by the DALI standard for certain DALI commands.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see E_DALICommandPriority [▶ 856]).



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



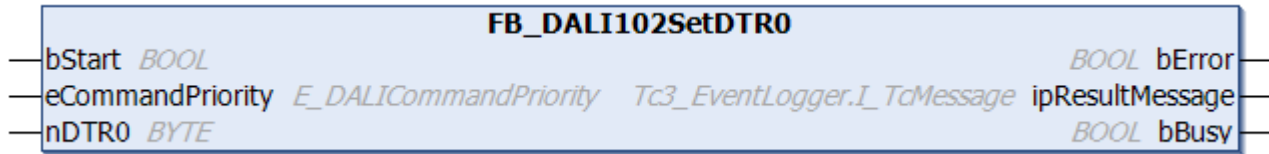
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.5.6 FB\_DALI102SetDTR0



The function block writes to the [DTR0 \[▶ 154\]](#) (Data Transfer Register 0) of all DALI control gears.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nDTR0      : BYTE;
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority [▶ 856]</a> ).



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDTR0	BYTE	Value to be written to the DTR0 (Data Transfer Register 0) (0...255).

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.5.7 FB\_DALI102SetDTR1**



The function block writes to the [DTR1 \[► 154\]](#) (Data Transfer Register 1) of all DALI control gears.

 **Inputs**

```

VAR_INPUT
    bStart          : BOOL;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
    nDTR1           : BYTE;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority [► 856]</a> ).

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDTR1	BYTE	Value to be written to the DTR1 (Data Transfer Register 1) (0...255).

 **Outputs**

```

VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.5.8 FB\_DALI102SetDTR2**



The function block writes to the [DTR2](#) [▶ 154] (Data Transfer Register 2) of all DALI control gears.

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nDTR2      : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority</a> [▶ 856]).



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDTR2	BYTE	Value to be written to the DTR2 (Data Transfer Register 2) (0...255).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

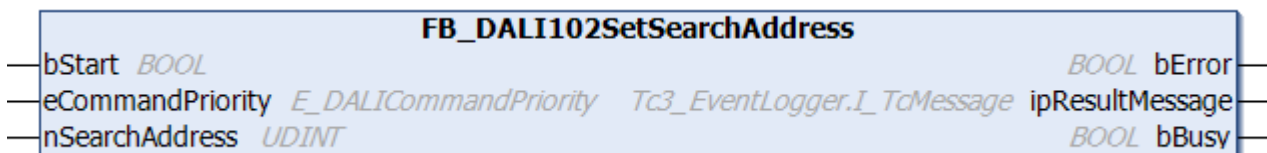
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.5.9 FB\_DALI102SetSearchAddress**



The function block stores the value *nSearchAddress* in the [searchAddress \[▶ 153\]](#) variable for all selected DALI control gears.

DALI control gears are selected if the DALI command INITIALIZE (see [FB\\_DALI102Initialise \[▶ 131\]](#)) was executed previously.

The DALI commands SEARCHADDRH, SEARCHADDRM and SEARCHADDRL are called internally.

This function block is required if addressing of the DALI control gears is to be implemented in the PLC. The function block [FB\\_DALI102Addressing \[▶ 21\]](#) can be used to assign the addresses from the PLC.



 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority := E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nSearchAddress  : UDINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority [▶ 856]</a> ).



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nSearchAddress	UDINT	The value to be written to the <i>searchAddress</i> variable (16#00_00_00...16#FF_FF_FF).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

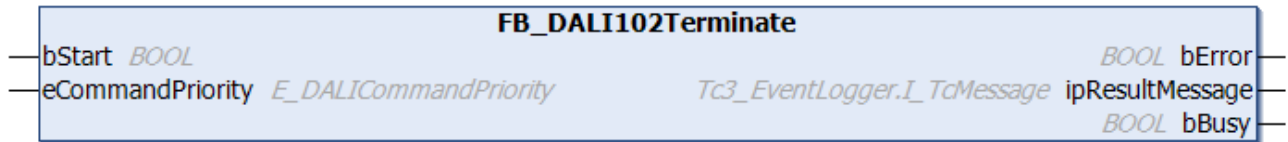
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.1.5.10 FB\_DALI102Terminate



The function block terminates addressing for all DALI control gears. The selection of the DALI control gears to be addressed by the [FB\\_DALI102Initialise](#) [▶ 131] function block is also canceled.

This function block is required if addressing of the DALI control gears is to be implemented in the PLC. The function block [FB\\_DALI102Addressing](#) [▶ 21] can be used to assign the addresses from the PLC.

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority</a> [▶ 856]).



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

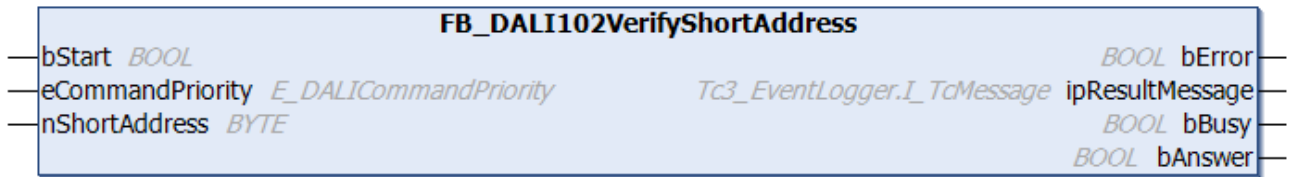
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.1.5.11 FB\_DALI102VerifyShortAddress



The function block checks whether the variable [shortAddress](#) [[▶ 153](#)] matches the value *nShortAddress* for all selected DALI control gears.

DALI control gears are selected if the [randomAddress](#) [[▶ 153](#)] of the DALI control gear is the same as the [searchAddress](#) [[▶ 153](#)] and the DALI command INITIALIZE (see [FB\\_DALI102Initialise](#) [[▶ 131](#)]) was executed previously.

This function block is required if addressing of the DALI control gears is to be implemented in the PLC. The function block [FB\\_DALI102Addressing](#) [[▶ 21](#)] can be used to assign the addresses from the PLC.

The output *bAnswer* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 Inputs

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nShortAddress   : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent (see <a href="#">E_DALICommandPriority</a> [ <a href="#">▶ 856</a> ]).

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nShortAddress	BYTE	Short address with which the own short address is compared

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  bAnswer        : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bAnswer	BOOL	Is TRUE if <i>nShortAddress</i> is equal to its own short address.

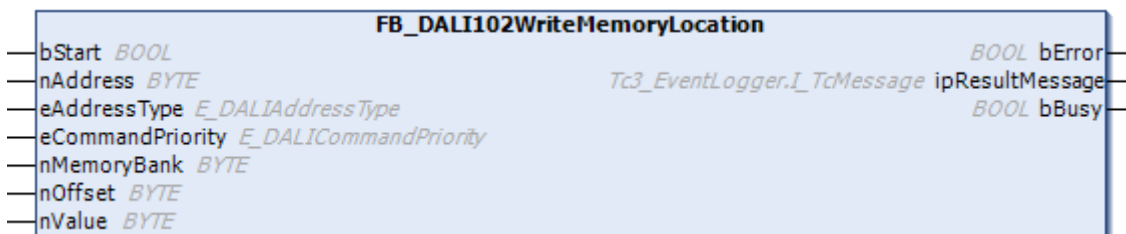
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.1.5.12 FB\_DALI102WriteMemoryLocation**



The value *nValue* is written in the memory bank of the DALI control gear. The corresponding memory bank is specified by the input variable *nMemoryBank*, the address within the memory bank by the input variable *nOffset*.

The DALI commands ENABLE WRITE MEMORY and WRITE MEMORY LOCATION are called internally.

In contrast to the function block [FB\\_DALI102WriteMemoryLocationNoReply \[▶ 146\]](#), the function block [FB\\_DALI102WriteMemoryLocation](#) checks that writing to the memory bank is correct.



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.



The function block sends the DALI command ENABLE WRITE MEMORY twice, as foreseen by the DALI standard for certain DALI commands.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMemoryBank : BYTE := 0;
  nOffset     : BYTE := 0;
  nValue      : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMemoryBank	BYTE	Specifies the memory bank for the write access.
nOffset	BYTE	Address within the memory bank selected with the variable <i>nMemoryBank</i> .
nValue	BYTE	Value to be written to the address within the memory bank.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

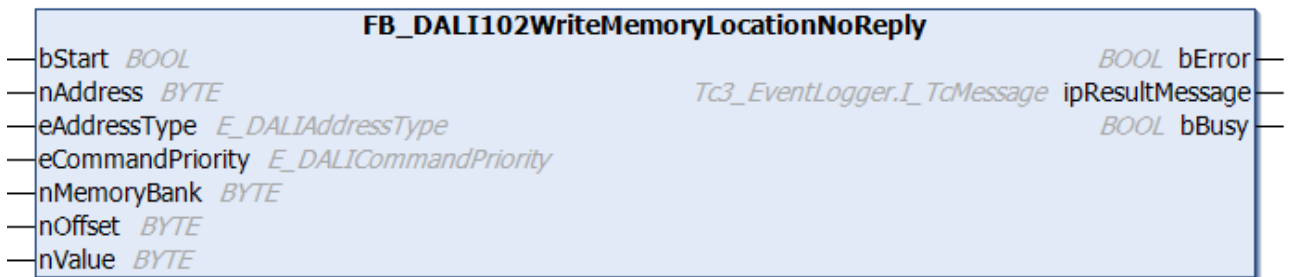
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICo mmunicati on	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.1.5.13 FB\_DALI102WriteMemoryLocationNoReply**



The value *nValue* is written in the memory bank of the DALI control gear. The corresponding memory bank is specified by the input variable *nMemoryBank*, the address within the memory bank by the input variable *nOffset*.

The DALI commands ENABLE WRITE MEMORY and WRITE MEMORY LOCATION – NO REPLY are called internally.

In contrast to the function block [FB\\_DALI102WriteMemoryLocation \[▶ 144\]](#), the function block [FB\\_DALI102WriteMemoryLocationNoReply](#) does not check that writing to the memory bank is correct.



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.




The function block sends the DALI command ENABLE WRITE MEMORY twice, as foreseen by the DALI standard for certain DALI commands.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMemoryBank     : BYTE := 0;
  nOffset         : BYTE := 0;
  nValue          : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

 *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMemoryBank	BYTE	Specifies the memory bank for the write access.
nOffset	BYTE	Address within the memory bank selected with the variable <i>nMemoryBank</i> .
nValue	BYTE	Value to be written to the address within the memory bank.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [►_892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.0	Tc3_DALI from v3.1.5.0

**4.1.2.1.6 Variables**

Each DALI control gear has a certain number of variables (parameters). These variables are used to configure the DALI control gear and thus influence its behavior. The values of the variables are stored in the respective DALI control gear.



Name	Reset value	Scope	Size	Comment
<a href="#">actualLevel [▶ 150]</a>	254	0, <i>minLevel...maxLevel</i>	1 byte	
<a href="#">powerOnLevel [▶ 150]</a>	254	0...255	1 byte	
<a href="#">systemFailureLevel [▶ 150]</a>	254	0...255	1 byte	
<a href="#">minLevel [▶ 150]</a>	PHM	PHM... <i>maxLevel</i>	1 byte	
<a href="#">maxLevel [▶ 150]</a>	254	<i>minLevel</i> ...254	1 byte	
<a href="#">fadeRate [▶ 150]</a>	<i>E_DALIFadeRate.N045StepsPerSec</i>	<i>E_DALIFadeRate.N003StepsPerSec</i> ... <i>E_DALIFadeRate.N358StepsPerSec</i>	1 byte	
<a href="#">fadeTime [▶ 151]</a>	<i>E_DALIFadeTime.Disabled</i>	<i>E_DALIFadeTime.Disabled</i> ... <i>E_DALIFadeTime.T90500ms</i>	1 byte	
<a href="#">extendedFadeTimeBase [▶ 152]</a>	<i>E_DALIExtendedFadeTimeBase.Base01</i>	<i>E_DALIExtendedFadeTimeBase.Base01</i> ... <i>E_DALIExtendedFadeTimeBase.Base15</i>	1 byte	
<a href="#">extendedFadeTimeMultiplier [▶ 152]</a>	<i>E_DALIExtendedFadeTimeMultiplier.Disabled</i>	<i>E_DALIExtendedFadeTimeMultiplier.Disabled</i> ... <i>E_DALIExtendedFadeTimeMultiplier.Multiplier1min</i>	1 byte	
<a href="#">shortAddress [▶ 153]</a>	No change	0...63, 255	1 byte	
<a href="#">searchAddress [▶ 153]</a>	16#FF_FF_FF	16#00_00_00...16#FF_FF_FF	3 bytes	
<a href="#">randomAddress [▶ 153]</a>	16#FF_FF_FF	16#00_00_00...16#FF_FF_FF	3 bytes	
<a href="#">controlGearFailure [▶ 153]</a>	The value can change after a RESET command.	TRUE, FALSE	1 byte	Read only
<a href="#">lampFailure [▶ 153]</a>	The value can change after a RESET command.	TRUE, FALSE	1 byte	Read only
<a href="#">lampOn [▶ 153]</a>	The value can change after a RESET command.	TRUE, FALSE	1 byte	Read only
<a href="#">limitError [▶ 153]</a>	FALSE	TRUE, FALSE	1 byte	Read only
<a href="#">fadeRunning [▶ 153]</a>	FALSE	TRUE, FALSE	1 byte	Read only
<a href="#">resetState [▶ 154]</a>	TRUE	TRUE, FALSE	1 byte	Read only
<a href="#">powerCycleSeen [▶ 154]</a>	FALSE	TRUE, FALSE	1 byte	Read only
<a href="#">gearGroups [▶ 154]</a>	16#0000	16#0000...16#FFFF	2 byte	
<a href="#">scene0...scene15 [▶ 154]</a>	255	0...255	1 byte	
<a href="#">DTR0 [▶ 154]</a>	No change	0...255	1 byte	
<a href="#">DTR1 [▶ 154]</a>	No change	0...255	1 byte	
<a href="#">DTR2 [▶ 154]</a>	No change	0...255	1 byte	
<a href="#">physicalMinLevel (PHM) [▶ 154]</a>	No change	1...255	1 byte	Read only

Name	Reset value	Scope	Size	Comment
<a href="#">deviceTypes</a> [ <a href="#">▶ 154</a> ]	No change	<i>E_DALIDeviceType.DT00FluorescentLamp</i> ... <i>E_DALIDeviceType.DT52DiagnosisMaintenance</i>	N bytes	Read only

### actualLevel

The variable contains the current output value of the DALI control gear.

The value of the variable can be read out with the function block [FB\\_DALI102QueryActualLevel](#) [[▶ 99](#)].

### powerOnLevel

When the supply voltage is applied to the DALI control gear, the lamp is controlled with the output value stored in the variable *powerOnLevel*. A prerequisite is that the DALI bus is supplied with power and that the quiescent level is maintained. If *powerOnLevel* is set to 255 (MASK), then after applying the supply voltage the output value is set to the last active output value (*minLevel*...*maxLevel*) before the supply voltage is switched off. If the output value is to be set to 0 after applying the supply voltage, *powerOnLevel* must also be 0. If *powerOnLevel* is greater than 0 and less than 255, the output value after applying the supply voltage is always limited by the variables *minLevel* and *maxLevel*.

The value of the variable can be queried with the function block [FB\\_DALI102QueryPowerOnLevel](#) [[▶ 121](#)] and changed with the function block [FB\\_DALI102SetPowerOnLevel](#) [[▶ 74](#)].

### systemFailureLevel

If an error occurs on the DALI bus (open-circuit voltage below the permissible level range for more than 500 ms), the DALI control gear is controlled with the output value from the *systemFailureLevel* variable. If the variable value is 255 (MASK), the output value does not change. The value range is limited by the variables *minLevel* and *maxLevel*.

The value of the variable can be queried with the function block [FB\\_DALI102QuerySystemFailureLevel](#) [[▶ 127](#)] and changed with the function block [FB\\_DALI102SetSystemFailureLevel](#) [[▶ 79](#)].

### minLevel / maxLevel

The output value is limited within the DALI control gear by the variables *minLevel* and *maxLevel*. Exceptions are the output values 0 (OFF) and 255 (MASK).

The value of the respective variable can be read out with the function blocks [FB\\_DALI102QueryMinLevel](#) [[▶ 117](#)] / [FB\\_DALI102QueryMaxLevel](#) [[▶ 116](#)] and changed with the function blocks [FB\\_DALI102SetMinLevel](#) [[▶ 73](#)] / [FB\\_DALI102SetMaxLevel](#) [[▶ 71](#)].

### fadeRate

The *fadeRate* determines the rate of change (in steps per second) of the output value. The variable has an effect on the DALI commands [FB\\_DALI102Up](#) [[▶ 97](#)] and [FB\\_DALI102Down](#) [[▶ 84](#)].

The value of the variable can be read with the function block [FB\\_DALI102QueryFadeTimeFadeRate](#) [[▶ 110](#)] and changed with the function block [FB\\_DALI102SetFadeRate](#) [[▶ 68](#)].

The following values can be specified via [E\\_DALIFadeRate](#) [[▶ 860](#)]:

E_DALIFadeRate	Value
N358StepsPerSec	358 steps/s
N253StepsPerSec	253 steps/s
N179StepsPerSec	179 steps/s
N127StepsPerSec	127 steps/s
N089StepsPerSec	89.4 steps/s
N063StepsPerSec	63.3 steps/s
N045StepsPerSec	44.7 steps/s
N032StepsPerSec	31.6 steps/s
N022StepsPerSec	22.4 steps/s
N016StepsPerSec	15.8 steps/s
N011StepsPerSec	11.2 steps/s
N008StepsPerSec	7.9 steps/s
N006StepsPerSec	5.6 steps/s
N004StepsPerSec	4.0 steps/s
N003StepsPerSec	2.8 steps/s

**fadeTime**

The *fadeTime* defines the time the current output value takes to reach the required value. If the lamp is switched on, the preheating and ignition times are not included in the fade time. The variable has an effect on the commands [FB\\_DALI102DirectArcPowerControl](#) [► 82], [FB\\_DALI102GoToScene](#) [► 87] and [FB\\_DALI102GoToLastActiveLevel](#) [► 85].

The value of the variable can be read with the function block [FB\\_DALI102QueryFadeTimeFadeRate](#) [► 110] and changed with the function block [FB\\_DALI102SetFadeTime](#) [► 70].

The following values can be specified via [E\\_DALIFadeTime](#) [► 860]:

E_DALIFadeTime	Value
Disabled	If <i>fadeTime</i> is set to <i>E_DALIFadeTime.Disabled</i> , the Extended Fade Time is used. This is calculated from the product of <i>extendedFadeTimeBase</i> and <i>extendedFadeTimeMultiplier</i> . If the Extended Fade Time is to be used, <i>fastFadeTime</i> must also be set to <i>E_DALIFastFadeTime.Disabled</i> if available.
T00707ms	0.7 s
T01000ms	1.0 s
T01400ms	1.4 s
T02000ms	2.0 s
T02800ms	2.8 s
T04000ms	4.0 s
T05700ms	5.7 s
T08000ms	8.0 s
T11300ms	11.3 s
T16000ms	16.0 s
T22600ms	22.6 s
T32000ms	32.0 s
T45300ms	45.3 s
T64000ms	64.0 s
T90500ms	90.5 s

**extendedFadeTimeBase**

This variable is the basic value for the calculation of the Extended Fade Time (0 ms to 16 min). The Extended Fade Time is used if `fadeTime` [► 151] was set to `E_DALIFadeTime.Disabled` and, if present, `fastFadeTime` [► 330] is equal to `E_DALIFastFadeTime.Disabled`.

The value of the variable can be read with the function block `FB_DALI102QueryExtendedFadeTime` [► 108] and changed with the function block `FB_DALI102SetExtendedFadeTime` [► 67].

The Extended Fade Time is calculated as follows:

Extended Fade Time = *extendedFadeTimeBase* x *extendedFadeTimeMultiplier*

The following basic values are available:

<b>E_DALIExtendedFadeTimeBase</b>	<b>Value</b>
Base01	1
Base02	2
Base03	3
Base04	4
Base05	5
Base06	6
Base07	7
Base08	8
Base09	9
Base10	10
Base11	11
Base12	12
Base13	13
Base14	14
Base15	15
Base16	16

**extendedFadeTimeMultiplier**

This variable is the multiplier for the calculation of the Extended Fade Time (0 ms to 16 min). The Extended Fade Time is used if `fadeTime` [► 151] was set to `E_DALIFadeTime.Disabled` and, if present, `fastFadeTime` [► 330] is equal to `E_DALIFastFadeTime.Disabled`.

If *extendedFadeTimeMultiplier* assumes the value `E_DALIExtendedFadeTimeMultiplier.Disabled`, the desired illuminance is reached as soon as possible.

The value of the variable can be read with the function block `FB_DALI102QueryExtendedFadeTime` [► 108] and changed with the function block `FB_DALI102SetExtendedFadeTime` [► 67].

The Extended Fade Time is calculated as follows:

Extended Fade Time = *extendedFadeTimeBase* x *extendedFadeTimeMultiplier*

The following multiplication values are available:

<b>E_DALIExtendedFadeTimeMultiplier</b>	<b>Value</b>
Disabled	Fade Time is not used
Multiplier100ms	x 100 ms
Multiplier1s	x 1 s
Multiplier10s	x 10 s
Multiplier1min	x 1 min

**shortAddress**

The short address is stored in this variable. A valid short address lies in the range between 0 and 63. With a value of 255 (MASK), the short address is considered deleted.

The short address is set with the function block [FB\\_DALI102SetShortAddress](#) [► 78].

**searchAddress**

The search address is only required for the assignment of short addresses.

The value of the variable can be changed with the function block [FB\\_DALI102SetSearchAddress](#) [► 140].

**randomAddress**

The random address, also known as the long address, is only required when short addresses are assigned.

With the help of the function block [FB\\_DALI102QueryRandomAddress](#) [► 123] the 3 bytes long random address can be read.

**controlGearFailure**

The variable indicates whether the DALI control gear operates as intended.

In the event of an error, the bit is set to TRUE after 30 s at the latest.

The bit is reset once the device works properly again.

The variable can be read out with the aid of the function block [FB\\_DALI102QueryStatus](#) [► 126].

**lampFailure**

The variable indicates whether the lamp is working correctly.

In the event of an error, the bit is set to TRUE after 30 s at the latest.

The fault may mean that the connection is faulty or that the lamp is defective.

The error state can only be detected when the lamp is switched on.

The variable can be read out with the aid of the function block [FB\\_DALI102QueryStatus](#) [► 126].

**lampOn**

The variable indicates whether the lamp is switched on.

This bit is set to FALSE when the lamp is off, during the startup phase and in the event of a total lamp failure.

The variable can be read out with the aid of the function block [FB\\_DALI102QueryStatus](#) [► 126].

**limitError**

The variable indicates whether the desired output value (*targetLevel*) lies within the limits of *minLevel* and *maxLevel*, or whether the *targetLevel* was changed by the values *minLevel* or *maxLevel*.

The bit is set to TRUE if one of the described cases occurs.

The variable can be read out with the aid of the function block [FB\\_DALI102QueryStatus](#) [► 126].

**fadeRunning**

This variable indicates whether dimming is active.

The bit is TRUE when dimming starts (after the start phase of the light) until the expiry of *fadeTime*.

The variable can be read out with the aid of the function block [FB\\_DALI102QueryStatus](#) [► 126].

### resetState

The variable indicates whether all device and instance variables have been set to their reset value (see table above).

In this case, the bit is TRUE.

Variables that show *no change* in the *Reset Value* column are not taken into account.

The variable can be read out with the aid of the function block [FB\\_DALI102QueryStatus \[▶ 126\]](#).

### powerCycleSeen

The variable indicates whether the DALI control gear has been supplied with power without a command having been executed to change the output value.

In this case, the bit is TRUE.

The variable can be read out with the aid of the function block [FB\\_DALI102QueryStatus \[▶ 126\]](#).

### gearGroups

The variable indicates whether a particular DALI control gear is assigned to groups 0 to 15. If the bit is set, the control gear belongs to the corresponding group.

The value of the variables can be read with the function block [FB\\_DALI102QueryGroups \[▶ 111\]](#) and changed with the function blocks [FB\\_DALI102AddToGroup \[▶ 56\]](#) and [FB\\_DALI102RemoveFromGroup \[▶ 59\]](#).

### scene0...scene15

Each DALI control gear can store output values for 16 different scenes. There is one output value per scene. If the command for calling a scene [FB\\_DALI102GoToScene \[▶ 87\]](#) is called for a device, a group or all devices (broadcast), the output value of the corresponding DALI control gear is set to the stored value. The output is limited by the values of *maxLevel*, *minLevel* and *physicalMinLevel*.

The value of the variable can be read from the function block [FB\\_DALI102QuerySceneLevel \[▶ 124\]](#) and changed with the function blocks [FB\\_DALI102SetScene \[▶ 76\]](#) and [FB\\_DALI102RemoveFromScene \[▶ 61\]](#).

### DTR0...DTR2

Data Transfer Registers (DTR) 0 to 2 serve as buffer for the execution of various DALI commands.

The DTRs are used by different function blocks. The description of the respective function block indicates which DTRs are used.

The values of the DTRs can be read out with the function blocks [FB\\_DALI102QueryContentDTRx \[▶ 100\]](#) and changed with the function blocks [FB\\_DALI102SetDTRx \[▶ 137\]](#).

### physicalMinLevel

The physically smallest possible output value is stored by the vendor in the variable *physicalMinLevel*.

The value can be read out with the function block [FB\\_DALI102QueryPhysicalMinimum \[▶ 120\]](#).

### deviceTypes

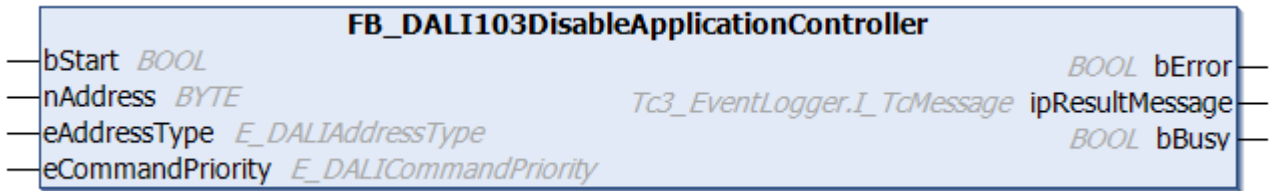
The variable contains the respective device types of a DALI control gear.

The device types (see [E\\_DALIDeviceType \[▶ 857\]](#)) of a DALI control gear can be read using the function block [FB\\_DALI102QueryDeviceTypes \[▶ 107\]](#).

4.1.2.2 Part 103 (control devices)

4.1.2.2.1 Configuration commands

4.1.2.2.1.1 FB\_DALI103DisableApplicationController



The function block deactivates the application controller (see [applicationActive](#) [▶ 230]).

The application controller can be activated with the function block [FB\\_DALI103EnableApplicationController](#) [▶ 159].

**i** The function block sends the DALI command DISABLE APPLICATION CONTROLLER twice, as required by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

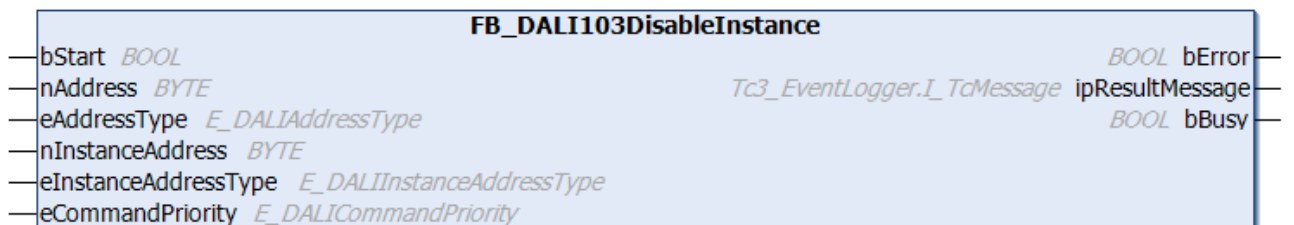
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.1.2 FB\_DALI103DisableInstance**



The function block disables the selected instance of the DALI control device (see [instanceActive](#) [▶ 232]).

The instance can be enabled with the function block [FB\\_DALI103EnableInstance](#) [▶ 161].

**i** The function block sends the DALI command DISABLE INSTANCE twice, as required by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
```



```
eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
eCommandPriority     : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	<a href="#">E_DALIInstanceAddressType</a> [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.2.1.3 FB\_DALI103DisablePowerCycleNotification



The function block locks the event *Power Cycle Notification*.

The event can be enabled with the function block [FB\\_DALI103EnablePowerCycleNotification](#) [▶ 162].



The function block sends the DALI command DISABLE POWER CYCLE NOTIFICATION twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

#### Inputs

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```

VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
  
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

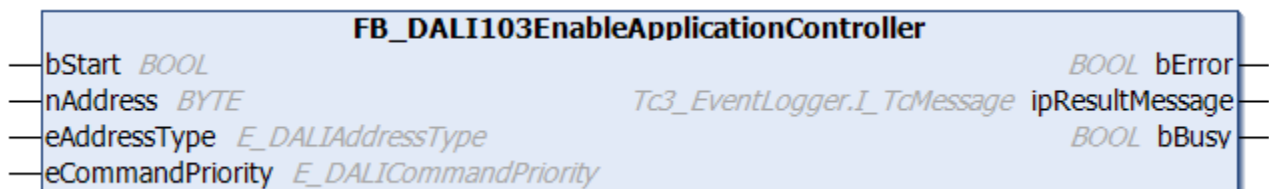
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.1.4 FB\_DALI103EnableApplicationController**



The function block enables the application controller (see [applicationActive \[▶ 230\]](#)).

The application controller can be disabled with the function block [FB\\_DALI103DisableApplicationController \[▶ 155\]](#).



The function block sends the DALI command ENABLE APPLICATION CONTROLLER twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

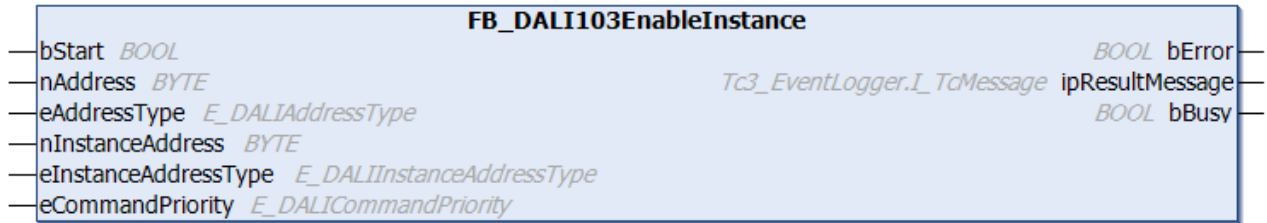
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.2.1.5 FB\_DALI103EnableInstance



The function block enables the selected instance of the DALI control device (see [instanceActive](#) [▶ 232]).

The instance can be disabled with the function block [FB\\_DALI103DisableInstance](#) [▶ 156].

**i** The function block sends the DALI command ENABLE INSTANCE twice, as required by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** The command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.1.6 FB\_DALI103EnablePowerCycleNotification**



The function block enables the event *Power Cycle Notification*.

The event can be disabled with the function block [FB\\_DALI103DisablePowerCycleNotification \[► 158\]](#).

**i** The function block sends the DALI command ENABLE POWER CYCLE NOTIFICATION twice, as required by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.2.1.7 FB\_DALI103IdentifyDevice



The function block starts the identification routine for a DALI control device.

The routine takes approx. 10 s and ends automatically. The exact scope of the identification routine depends on the manufacturer of the DALI control device.

**i** The function block sends the DALI command IDENTIFY DEVICE twice, as required by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

#### Inputs

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** The command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```

VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
  
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.1.8 FB\_DALI103Reset**



The function block resets all device and instance variables of the DALI control device to their default values (see [resetState \[▶ 231\]](#)).

DALI control devices can take up to 300 ms to execute the DALI command. Only then are further DALI commands executed by the respective control device.



The function block sends the DALI command RESET twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

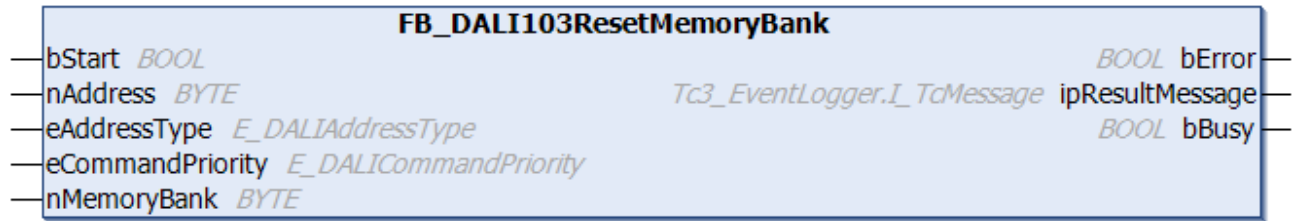
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.1.9 FB\_DALI103ResetMemoryBank



The values of the selected memory bank of the DALI control device are set to their respective default values by the function block (see [Memory banks](#) [▶ 16]).

During the reset, the DALI control device does not react to other commands.

The function block also resets the values that are write-protected by the lock byte.

The DALI commands ENABLE WRITE MEMORY, WRITE MEMORY LOCATION – NO REPLY and RESET MEMORY BANK are called internally.

**i** The function block changes the DTR0 and the DTR1 of all connected DALI control devices.

**i** The function block sends the DALI command RESET MEMORY BANK twice, as foreseen by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMemoryBank : BYTE := 2;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMemoryBank	BYTE	Specification of the memory bank. If a 0 is specified, all memory banks are reset (except for memory bank 0, as this is read-only). For all other values, only the respective memory bank is reset.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.2.1.10 FB\_DALI103SavePersistentVariables**



The function block immediately triggers the saving of variables in the persistent memory by the DALI command SAVE PRESISTENT VARIABLES.

DALI control devices do not save the variables directly in the persistent memory. Instead, the values are copied to the persistent memory in the background. Since this happens over a longer period of time (a few seconds), the values can be lost if the DALI bus is disconnected.

The SAVE PRESISTENT VARIABLES DALI command triggers immediate copying to the persistent memory. In most cases, no communication with the DALI device can take place during copying (up to 350 ms).



The function block sends the DALI command SAVE PRESISTENT VARIABLES twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

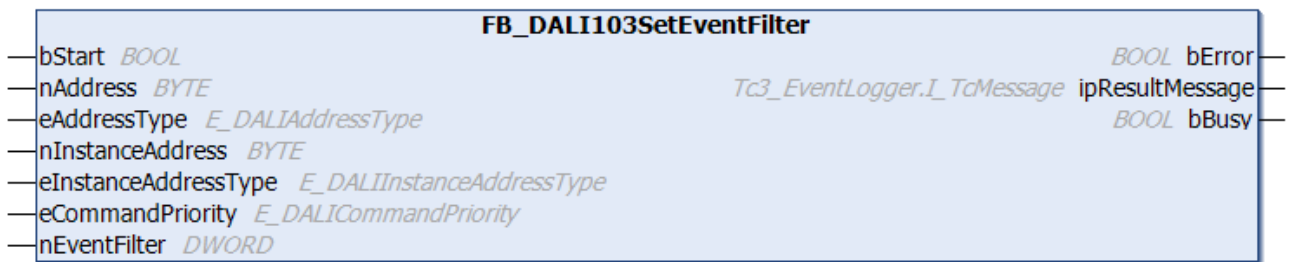
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.1.11 FB\_DALI103SetEventFilter**



The function block sets the event filter (see [eventFilter](#) [[▶ 232](#)]) for the corresponding instance of the DALI control device.

Each bit of the input variable *nEventFilter* represents an event. If the bit is set, the associated event is enabled.

The meanings of the individual bits for the respective instance types can be found here:

Part 301 (push buttons) – [Introduction](#) [[▶ 512](#)]

Part 302 (absolute input devices) - [Introduction](#) [[▶ 532](#)]

Part 303 (occupancy sensors) - [Introduction](#) [[▶ 543](#)]

Part 304 (light sensors) - [Introduction](#) [[▶ 561](#)]



The function block changes the DTR0, DTR1 and DTR2 of all connected DALI control devices.



The function block sends the DALI command SET EVENT FILTER twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
    
```

```
eCommandPriority      : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
nEventFilter          : DWORD := 0;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nEventFilter	DWORD	Each bit represents an event to be enabled or disabled.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

## Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

## 4.1.2.2.1.12 FB\_DALI103SetEventPriority



This function block sets the event priority (see [eventPriority](#) [► 233]) for the corresponding instance of the DALI control device.

With DALI control devices the event priority affects all *Input Notification Events* that the respective instance of the DALI control device sends. With DALI control devices, the event priority can assume values from *E\_DALIEventPriority.Low ... E\_DALIEventPriority.MiddleHigh*.

In most cases the default value should be used.



The function block changes the DTR0 of all connected DALI control devices.



The function block sends the DALI command SET EVENT PRIORITY twice, as foreseen by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

 Inputs

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eEventPriority   : E_DALIEventPriority := E_DALIEventPriority.Middle;
END_VAR

```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eEventPriority	E_DALIEventPriority	Priority (low, middle low, middle, middle high) with which the <i>Input Notification Events</i> are sent from the instance of the DALI control device (see E_DALIEventPriority [▶ 858]).

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

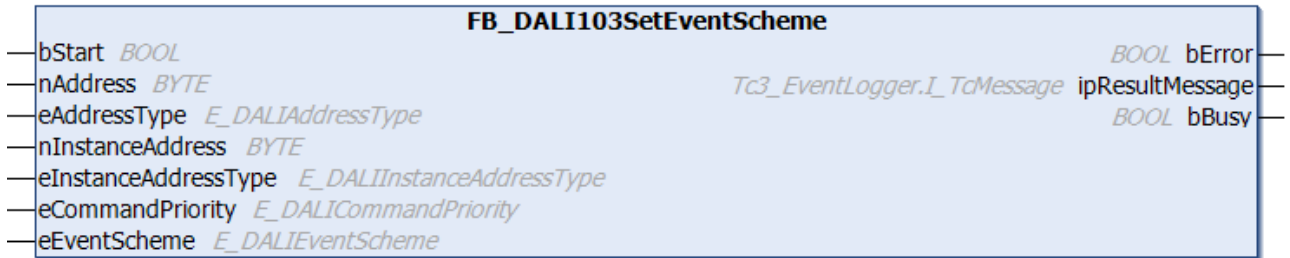
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.2.1.13 FB\_DALI103SetEventScheme**



The function block determines the event scheme (see [eventScheme](#) [▶ 233]) for the events of the respective instance of the DALI control device.



The function block changes the DTR0 of all connected DALI control devices.



The function block sends the DALI command SET EVENT SCHEME twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

**Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eEventScheme    : E_DALIEventScheme := E_DALIEventScheme.DeviceInstance;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eEventScheme	E_DALIEventScheme	Event scheme for the events (see <a href="#">E_DALIEventScheme</a> [ <a href="#">▶ 859</a> ])

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [ <a href="#">▶ 887</a> ]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [ <a href="#">▶ 873</a> ]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

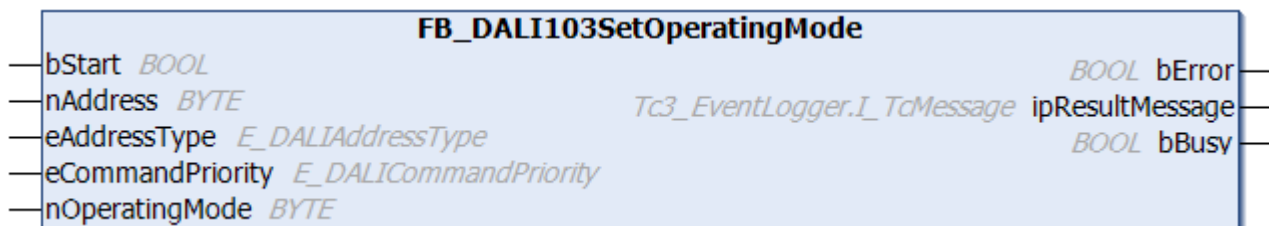
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.1.14 FB\_DALI103SetOperatingMode**



The function block sets the operating mode of the DALI control device (see [operatingMode](#) [[▶ 230](#)]).



The function block changes the DTR0 of all connected DALI control devices.



The function block sends the DALI command SET OPERATING MODE twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nOperatingMode  : BYTE := 0;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nOperatingMode	BYTE	Operating mode entry

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

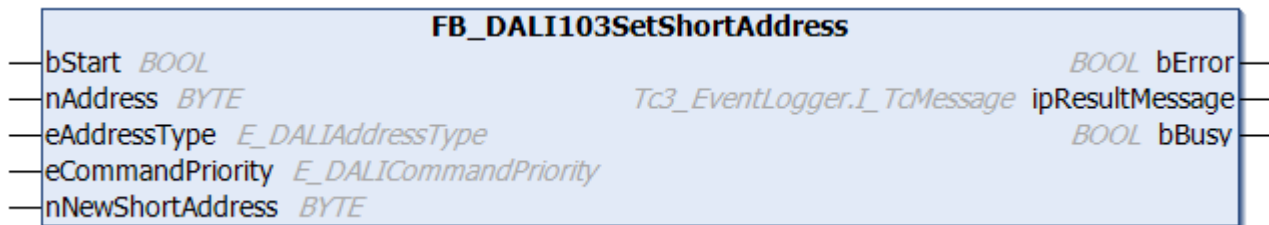
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.1.15 FB\_DALI103SetShortAddress**



The function block stores the new short address *nNewShortAddress* in the DALI control device (see [shortAddress \[▶ 229\]](#)).

The existing short address in the DALI control device is deleted if a value of 255 is specified at the input.

To address a device without a known short address, the command must be sent as a broadcast (*eAddressType* = *E\_DALIAddressType.Broadcast*). This gives all control devices connected to the DALI terminal the short address *nNewShortAddress*.

Alternatively, the function block can be called with Broadcast unaddressed (*eAddressType* = *E\_DALIAddressType.BroadcastUnaddr*). In this case, only the devices that do not yet have a short address are assigned a short address.



The function block changes the DTR0 of all connected DALI control devices.



The function block sends the DALI command SET SHORT ADDRESS twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nNewShortAddress : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nNewShortAddress	BYTE	New short address (0...63, 255)

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.1.16 FB\_DALI103StartQuiescentMode



The function block enables the [Quiescent Mode \[► 230\]](#) for approx. 15 minutes.

In Quiescent mode, no commands and events are sent from the DALI control device.

Quiescent mode can be terminated prematurely with the function block [FB\\_DALI103StopQuiescentMode \[► 180\]](#).

**i** The function block sends the DALI command START QUIESCENT MODE twice, as required by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.1.17 FB\_DALI103StopQuiescentMode**



The function block interrupts the [Quiescent mode](#) [▶ 230] prematurely.

Quiescent mode can be started with the function block [FB\\_DALI103StartQuiescentMode](#) [▶ 179].



The function block sends the DALI command STOP QUIESCENT MODE twice, as required by the DALI standard for certain DALI commands.





The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block ▶ 892</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2 Query commands**

**4.1.2.2.2.1 FB\_DALI103QueryContentDTR0**



The function block reads the contents of DTR0 [▶ 229] (Data Transfer Register 0) from the DALI control device.

The output *nContentDTR0* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nContentDTR0 : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nContentDTR0	BYTE	Content of the DTR0 (Data Transfer Register 0) (0...255)

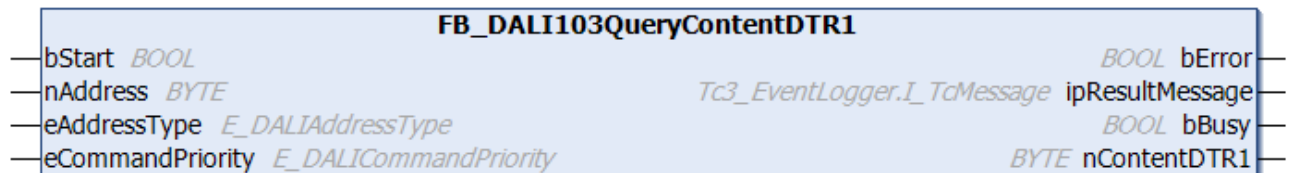
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2 FB\_DALI103QueryContentDTR1**



The function block reads the contents of [DTR1 \[▶ 229\]](#) (Data Transfer Register 1) from the DALI control device.

The output *nContentDTR1* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nContentDTR1    : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nContentDTR1	BYTE	Content of the DTR1 (Data Transfer Register 1) (0...255)

### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.2.3 FB\_DALI103QueryContentDTR2



The function block reads the contents of DTR2 [▶ 229] (Data Transfer Register 2) from the DALI control device.

The output *nContentDTR2* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nContentDTR2 : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nContentDTR2	BYTE	Content of the DTR2 (Data Transfer Register 2) (0...255)

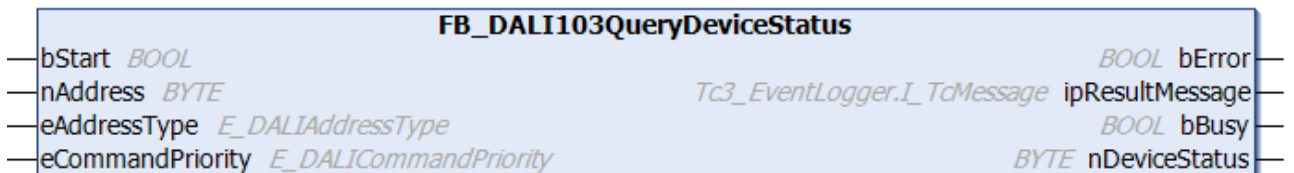
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2.4 FB\_DALI103QueryDeviceStatus**



The function block reads the status of the DALI control device.

The following status information is provided at output *nDeviceStatus*.

Bit	Description
0	Status of the DALI control device (see <a href="#">inputDeviceError [► 231]</a> )
1	Quiescent mode (see <a href="#">quiescentMode [► 230]</a> )
2	Short address missing (variable <a href="#">shortAddress [► 229]</a> is 255 (MASK))
3	Application controller active (see <a href="#">applicationActive [► 230]</a> )
4	Application controller error (see <a href="#">applicationControllerError [► 230]</a> )
5	The power supply was enabled (see <a href="#">powerCycleSeen [► 230]</a> )
6	Reset state (see <a href="#">resetState [► 231]</a> )
7	Unused (default value is 0)

The output *nDeviceStatus* only contains a valid value if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nDeviceStatus   : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nDeviceStatus	BYTE	Output value of the Device Status, see table above.

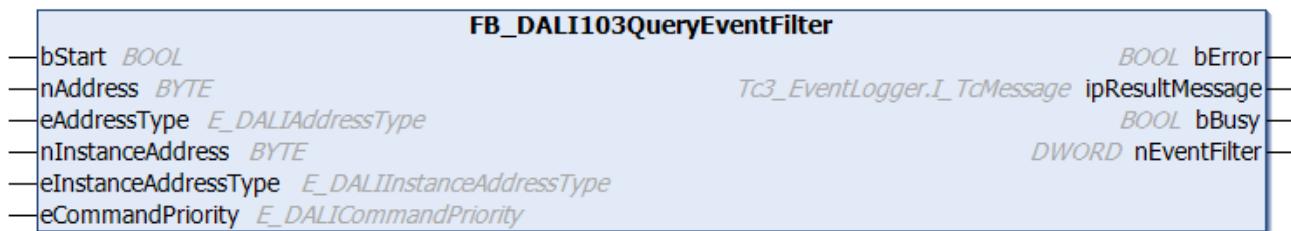
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2.5 FB\_DALI103QueryEventFilter**



The function block reads the event filter (see [eventFilter \[► 232\]](#)) for the selected instance of the DALI control device.

Each bit of the input variable *nEventFilter* represents an event. If the bit is set, the associated event is enabled.

The meanings of the individual bits for the respective instance types can be found here:

Part 301 (push buttons) – [Introduction \[► 512\]](#)

Part 302 (absolute input devices) - [Introduction \[► 532\]](#)

Part 303 (occupancy sensors) - [Introduction \[► 543\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The output *nEventFilter* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nEventFilter    : DWORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nEventFilter	DWORD	Each bit represents a manufacturer-specific event that has been enabled or disabled.

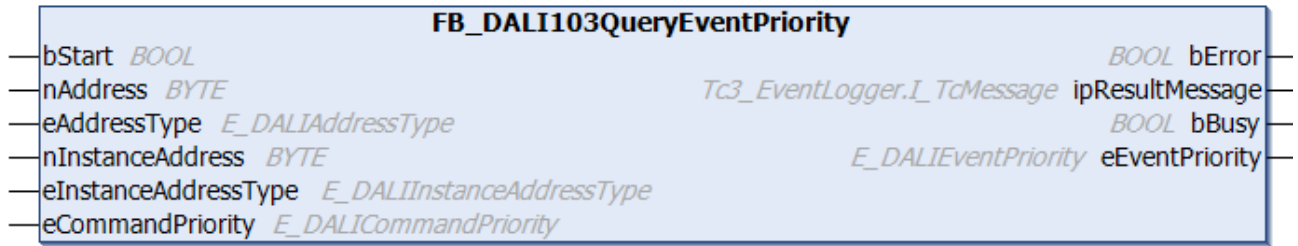
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.2.2.6 FB\_DALI103QueryEventPriority



This function block reads the event priority (see [eventPriority](#) [▶ 233]) for the selected instance of the DALI control device.

With DALI control devices the event priority affects all *Input Notification Events* that the respective instance of the DALI control device sends. With DALI control devices, the event priority can assume values from *E\_DALIEventPriority.Low ... E\_DALIEventPriority.MiddleHigh*.

The output *eEventPriority* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    nInstanceAddress : BYTE := 0;
    eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
```

```
bBusy          : BOOL;
eEventPriority : E_DALIEventPriority;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eEventPriority	E_DALIEventPriority	Returns the priority (low, middle low, middle, middle high) with which the <i>Input Notification Events</i> are sent from the instance of the DALI control device.

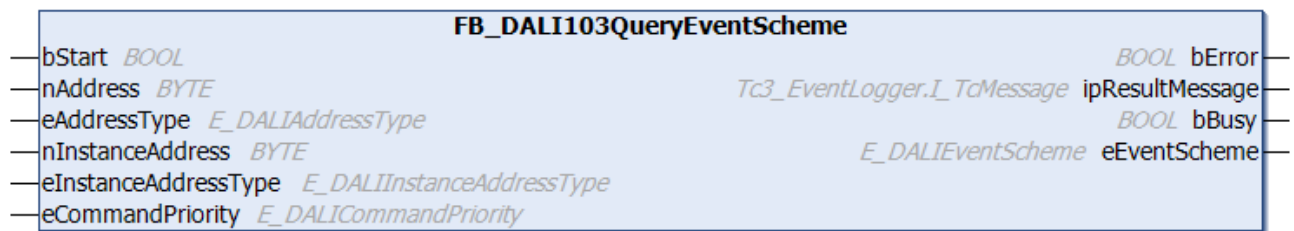
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.2.2.7 FB\_DALI103QueryEventScheme**



The function block reads the event scheme (see [eventScheme \[▶ 233\]](#)) for the events of the selected DALI control device instance.

The output *eEventScheme* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
```

```
eAddressType      : E_DALIAddressType := E_DALIAddressType.Short;
nInstanceAddress  : BYTE := 0;
eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
eCommandPriority  : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	<a href="#">E_DALIInstanceAddressType</a> [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

### 📡 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  eEventScheme    : E_DALIEventScheme;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eEventScheme	<a href="#">E_DALIEventScheme</a> [▶ 859]	Event scheme for the events



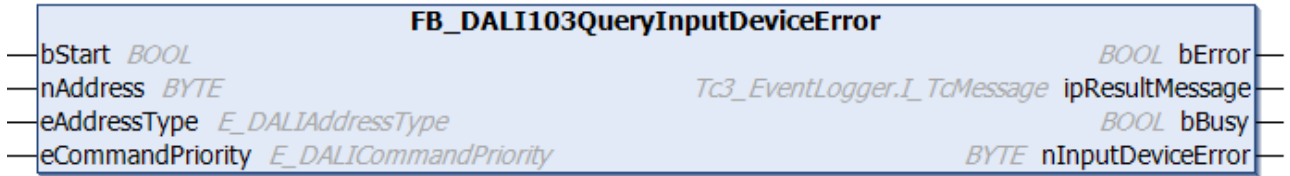
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	<a href="#">I_DALICommunication</a>	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.2.8 FB\_DALI103QueryInputDeviceError



The function block reads more detailed error information from the DALI control device (see [inputDeviceError](#) [▶ 231]).

The meaning depends on the manufacturer of the DALI control device.

The output *nInputDeviceError* only contains a valid value if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nInputDeviceError : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nInputDeviceError	BYTE	Output of the error information of the DALI control device (vendor-dependent).

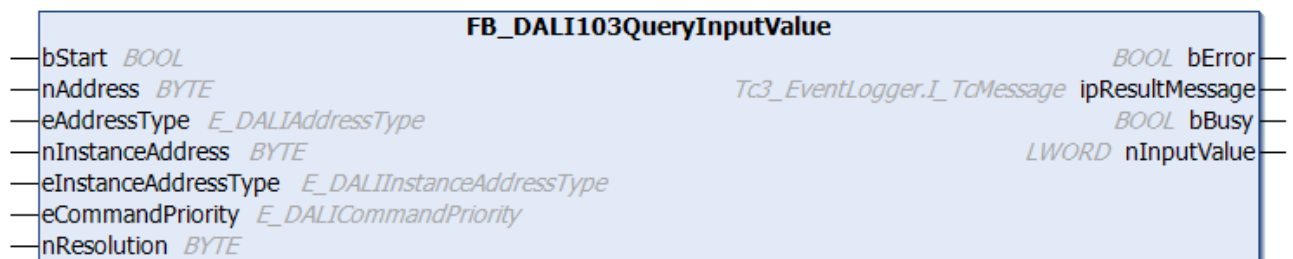
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2.9 FB\_DALI103QueryInputValue**



The function block queries the input value (see [inputValue](#) [▶ 232]) of the selected instance of the DALI control device.

The parameter *nResolution* specifies how high the resolution (in bits) of the input value is. The function block [FB\\_DALI103QueryResolution](#) [▶ 208] can be used to read the resolution.

All bytes of the input value are read and subsequently decoded.

Make sure that the parameter *nResolution* corresponds to the actual resolution of the device, otherwise errors may occur during decoding.

The possible input values for the respective instance types can be found here:

Part 301 (push buttons) – [Introduction](#) [▶ 511]

Part 302 (absolute input devices) - [Introduction](#) [▶ 532]

Part 303 (occupancy sensors) - [Introduction](#) [▶ 543]

Part 304 (light sensors) - [Introduction](#) [▶ 561]

The DALI commands QUERY INPUT VALUE and QUERY INPUT VALUE LATCH are called internally.

The output *nInputValue* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nResolution     : BYTE := 8;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nResolution	BYTE	Resolution of the input value in bits (1...64).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nInputValue     : LWORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nInputValue	LWORD	The input value of the selected instance of the DALI control device.

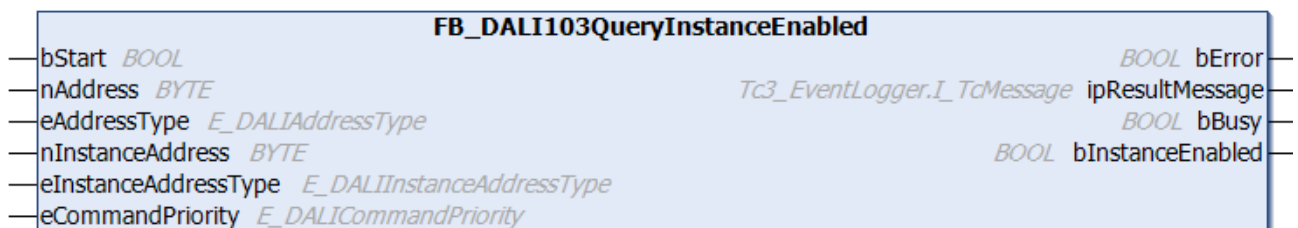
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICo mmunicati on	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2.10 FB\_DALI103QueryInstanceEnabled**



The function block queries whether the specified instance of the DALI control device is active (see [instanceActive](#) [▶ 232]).

The instance can be enabled with the function block [FB\\_DALI103EnableInstance](#) [▶ 161].

The instance can be disabled with the function block [FB\\_DALI103DisableInstance](#) [▶ 156].

The output *bInstanceEnabled* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
```



```
nInstanceAddress      : BYTE := 0;
eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
eCommandPriority     : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bInstanceEnabled : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bInstanceEnabled	BOOL	Is TRUE if the instance is active.

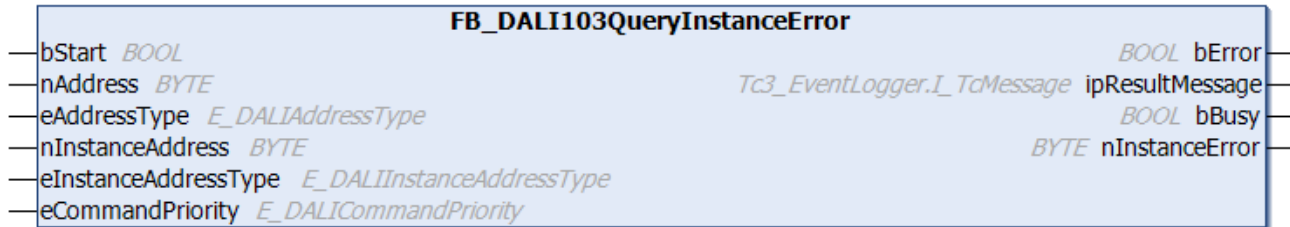
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2.11 FB\_DALI103QueryInstanceError**



The function block reads more detailed error information for the selected instance from the DALI control device (see [instanceError](#) [▶ 233]).

The meaning of the values depends on the selected instance type.

The output *nInstanceError* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

**Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nInstanceError  : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nInstanceError	BYTE	Error information for the selected instance.

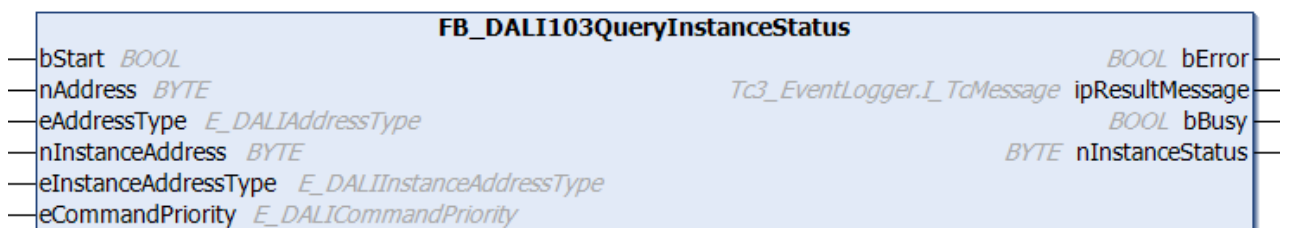
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.12 FB\_DALI103QueryInstanceStatus**



The function block reads the status of the selected instance from the DALI control device.

Bit	Description
0	Status of the instance (see <a href="#">instanceError</a> [▶ 233])
1	Events are enabled for the instance (see <a href="#">instanceActive</a> [▶ 232])
2...7	Unused (default value is 0)

The output *nInstanceStatus* only contains a valid value if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nInstanceStatus : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nInstanceStatus	BOOL	Status of the selected instance

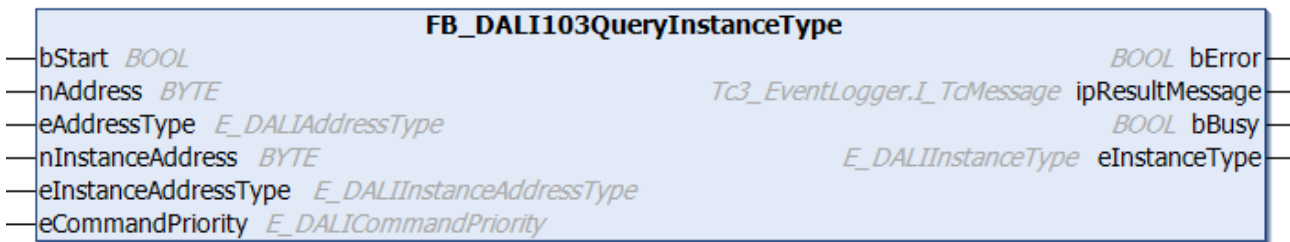
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.13 FB\_DALI103QueryInstanceType



The function block reads the instance type (see [instanceType](#) [[▶ 232](#)]) for the selected instance from the DALI control device.

The output *eInstanceType* only contains a valid value if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType <a href="#">▶ 861</a>	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  eInstanceType  : E_DALIInstanceType;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eInstanceType	E_DALIInstanceType	Output of the instance type (see <a href="#">instanceType [▶ 232]</a> ).



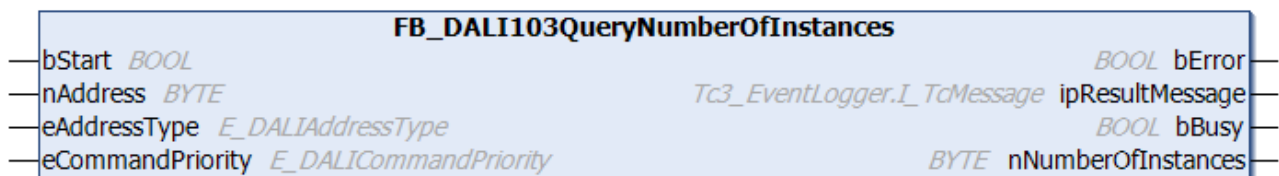
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.2.2.14 FB\_DALI103QueryNumberOfInstances**



The function block reads the number of existing instances from the DALI control device (see [numberOfInstances \[▶ 230\]](#)).

The output *nNumberOfInstances* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nNumberOfInstances : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nNumberOfInstances	DALI	Number of instances in the DALI control device.

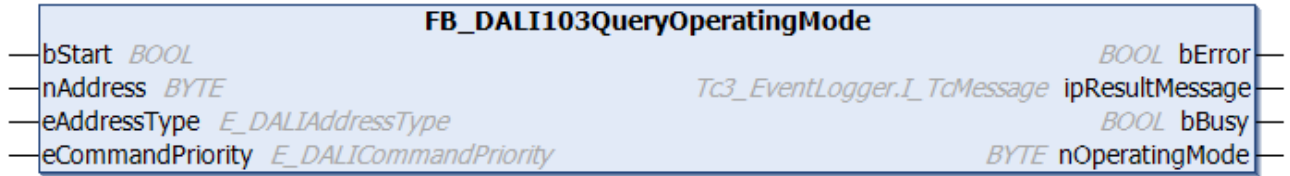
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2.15 FB\_DALI103QueryOperatigMode**



The function block reads the variable `operatingMode` [► 230] from the DALI control device.

The meaning of this variable is defined by the manufacturer of the DALI control device.

The output `nOperatingMode` only contains a valid value if the function block was executed without errors (`bError = FALSE`).



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...31). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nOperatingMode : BYTE;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nOperatingMode	BYTE	Output of the variable <a href="#">operatingMode</a> [▶ 230].

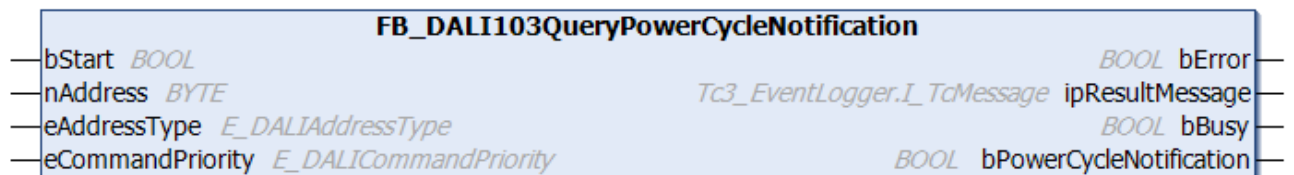
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2.16 FB\_DALI103QueryPowerCycleNotification**



The function block outputs whether the *Power Cycle Notification* event of a DALI control device is enabled.

The event can be enabled with the function block [FB\\_DALI103EnablePowerCycleNotification](#) [▶ 162].

The event can be disabled with the function block [FB\\_DALI103DisablePowerCycleNotification](#) [▶ 158].

The output *bPowerCycleNotification* only contains a valid value if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bBusy            : BOOL;
  bPowerCycleNotification : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bPowerCycleNotification	BOOL	Is TRUE if the <i>Power Cycle Notification</i> event is enabled.



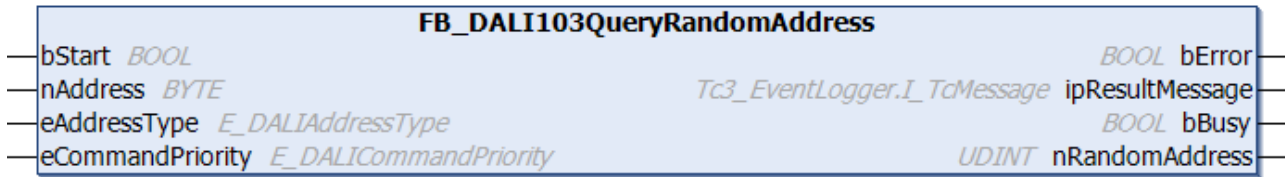
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.2.17 FB\_DALI103QueryRandomAddress



The function block reads the variable `randomAddress` [▶ 229] from the DALI control device.

The DALI commands QUERY RANDOM ADDRESS (H), QUERY RANDOM ADDRESS (M) and QUERY RANDOM ADDRESS (L) are called internally.

The output `nRandomAddress` contains a valid value only if the function block was executed without errors (`bError = FALSE`).



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...31). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nRandomAddress : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRandomAddress	UDINT	Random address (16#00_00_00...16#FF_FF_FE).

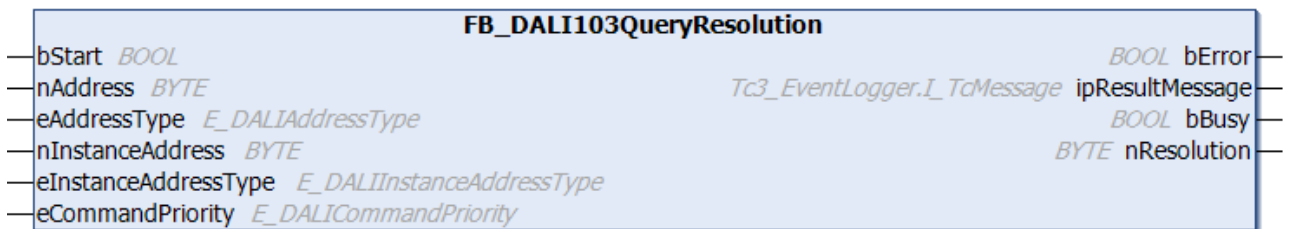
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.2.18 FB\_DALI103QueryResolution**



The function block reads the resolution of the input value for the selected instance from the DALI control device (see [resolution](#) [▶ 232]).

The output *nResolution* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nResolution     : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nResolution	BYTE	Number of bits with which the input value is resolved.

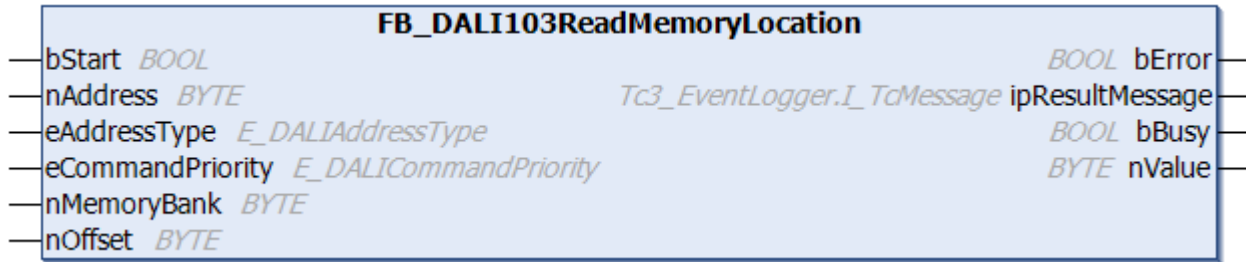
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.2.2.19 FB\_DALI103ReadMemoryLocation



The value *nValue* is read from the memory bank of the DALI control device. The corresponding memory bank is specified by the input variable *nMemoryBank*, the address within the memory bank by the input variable *nOffset*.

**i** The function block changes the DTR0 and the DTR1 of all connected DALI control devices.

The output *nValue* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

**i** The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMemoryBank : BYTE;
  nOffset     : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMemoryBank	BYTE	Specifies the memory bank for the read access (DTR1).
nOffset	BYTE	Address from which a value within the memory bank is to be read (DTR0).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nValue         : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nValue	BYTE	Value read from the memory bank.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

**4.1.2.2.3 Special commands**

**4.1.2.2.3.1 FB\_DALI103Initialise**



The function block prepares the addressing of the DALI control devices.

This function block is required if addressing of the DALI control devices is to be implemented in the PLC. The function block [FB\\_DALI103Addressing](#) [▶ 49] can be used to assign the addresses from the PLC.

*nParameter* is used to define which DALI control devices are taken into account for addressing:

Value (binary)	Description
2#0111_1111	DALI control devices without a short address respond.
2#00AA_AAAA	DALI control devices with the address 2#00AA_AAAA respond.
2#1111_1111	All DALI control devices respond.



The function block sends the DALI command INITIALIZE twice, as required by the DALI standard for certain DALI commands.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
  nParameter      : BYTE := 2#0111 1111;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nParameter	BYTE	Specifies which control devices are to respond to this command (see table above).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



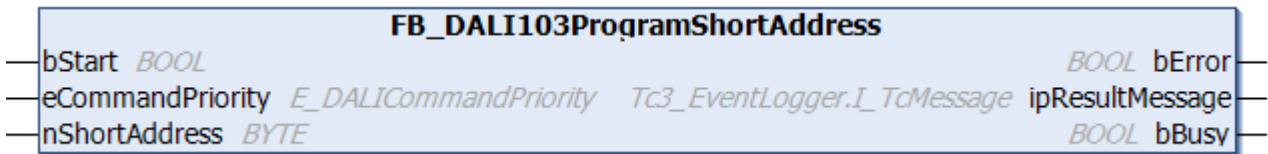
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.3.2 FB\_DALI103ProgramShortAddress



The function block stores the value *nShortAddress* as short address in the variable *shortAddress* [[▶ 229](#)] for all selected DALI control devices.

DALI control devices are selected if the *randomAddress* of the DALI control device is the same as the *searchAddress* and the DALI command INITIALISE (see [FB\\_DALI103Initialise](#) [[▶ 211](#)]) was executed previously.

This function block is required if addressing of the DALI control devices is to be implemented in the PLC. The function block [FB\\_DALI103Addressing](#) [[▶ 49](#)] can be used to assign the addresses from the PLC.



The function block cannot be used when using the KL6811.

 Inputs

```

VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
  nShortAddress   : BYTE;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [ <a href="#">▶ 856</a> ]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nShortAddress	BYTE	Short address to be assigned (0...63, 255).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.3.3 FB\_DALI103Randomise**



The function block generates a new random address for all selected DALI control devices and writes it to the *randomAddress* variable. DALI control devices are selected if the DALI command INITIALISE (see [FB\\_DALI103Initialise](#) [▶ 211]) was executed previously.

This function block is required if addressing of the DALI control devices is to be implemented in the PLC. The function block [FB\\_DALI103Addressing](#) [▶ 49] can be used to assign the addresses from the PLC.

**i** The function block sends the DALI command RANDOMIZE twice, as required by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.2.3.4 FB\_DALI103SetDTR0



The function block writes to the [DTR0 \[► 229\]](#) (Data Transfer Register 0) of all DALI control devices.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
  nDTR0           : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	<a href="#">E_DALICommandPriority [► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDTR0	BYTE	Value to be written to the DTR0 (Data Transfer Register 0) (0...255).

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.3.5 FB\_DALI103SetDTR1



The function block writes to [DTR1 \[► 229\]](#) (Data Transfer Register 1) of all DALI control devices.



The function block cannot be used when using the KL6811.

 Inputs

```

VAR_INPUT
    bStart          : BOOL;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
    nDTR1           : BYTE;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	<a href="#">E_DALICommandPriority [► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDTR1	BYTE	Value to be written to the DTR1 (Data Transfer Register 1) (0...255).

 Outputs

```

VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

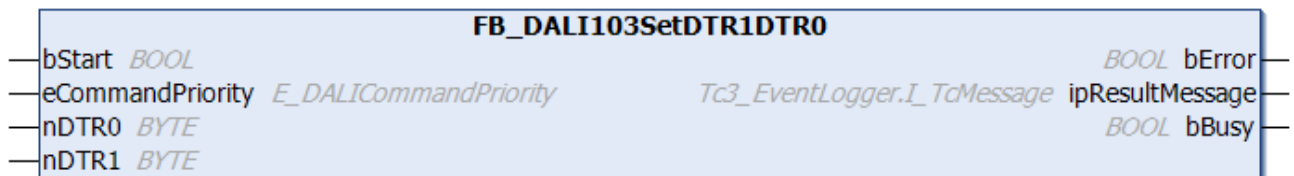
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.3.6 FB\_DALI103SetDTR1DTR0**



The function block writes the DTR0 (Data Transfer Register 0) and the DTR1 (Data Transfer Register 1) of all DALI control devices.



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
  nDTR0      : BYTE;
  nDTR1      : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	<a href="#">E_DALICommandPriority [▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDTR0	BYTE	Value to be written to the DTR0 (Data Transfer Register 0) (0...255).
nDTR1	BYTE	Value to be written to the DTR1 (Data Transfer Register 1) (0...255).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.0	Tc3_DALI from v3.1.5.0

**4.1.2.2.3.7 FB\_DALI103SetDTR2**



The function block writes to [DTR2 \[▶ 229\]](#) (Data Transfer Register 2) of all DALI control devices.



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
  nDTR2           : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDTR2	BYTE	Value to be written to the DTR2 (Data Transfer Register 2) (0...255).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

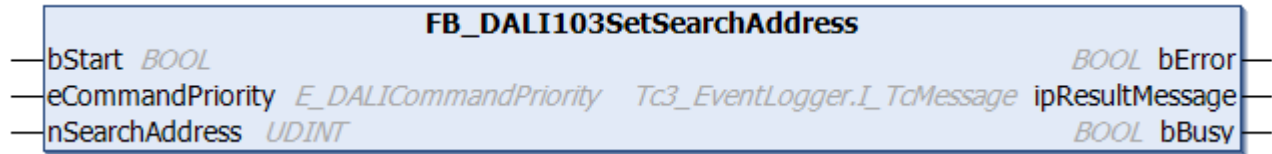
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block ▶ 892</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0



### 4.1.2.2.3.8 FB\_DALI103SetSearchAddress



The function block stores the value *nSearchAddress* in the *searchAddress* [▶ 229] variable for all selected DALI control devices.

DALI control devices are selected if the DALI command INITIALISE (see *FB\_DALI103Initialise* [▶ 211]) was executed previously.

The DALI commands SEARCHADDRH, SEARCHADDRM and SEARCHADDRL are called internally.

This function block is required if addressing of the DALI control devices is to be implemented in the PLC. The function block *FB\_DALI103Addressing* [▶ 49] can be used to assign the addresses from the PLC.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
  nSearchAddress : UDINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nSearchAddress	UDINT	The value to be written to the <i>searchAddress</i> variable (16#00_00_00...16#FF_FF_FF).

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

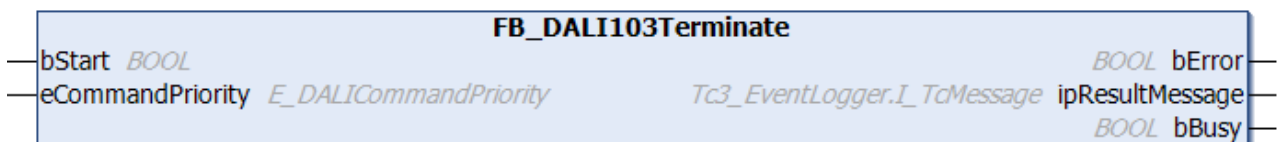
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.3.9 FB\_DALI103Terminate**



The function block terminates addressing in all DALI control devices. The selection of the devices to be addressed by the function block [FB\\_DALI103Initialise](#) [▶ 211] is also canceled.

This function block is required if addressing of the DALI control devices is to be implemented in the PLC. The function block [FB\\_DALI103Addressing](#) [▶ 49] can be used to assign the addresses from the PLC.



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

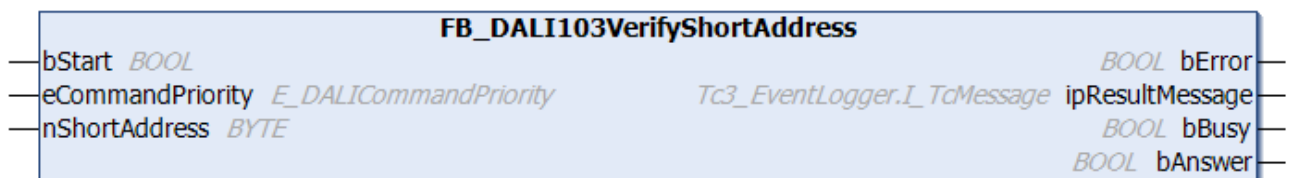
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.2.3.10 FB\_DALI103VerifyShortAddress**



The function block checks whether the variable [shortAddress](#) [▶ 229] matches the value *nShortAddress* for all selected DALI control devices.

DALI control devices are selected if the DALI command INITIALISE (see [FB\\_DALI103Initialise](#) [▶ 211]) was executed previously. The DALI commands SEARCHADDRH, SEARCHADDRM and SEARCHADDRL are called internally.

This function block is required if addressing of the DALI control devices is to be implemented in the PLC. The function block [FB\\_DALI103Addressing](#) [▶ 49] can be used to assign the addresses from the PLC.

The output *bAnswer* only contains a valid value if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.Middle;
  nShortAddress : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nShortAddress	BYTE	Short address with which the own short address is compared

### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  bAnswer    : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bAnswer	BOOL	Is TRUE if <i>nShortAddress</i> is equal to its own short address.

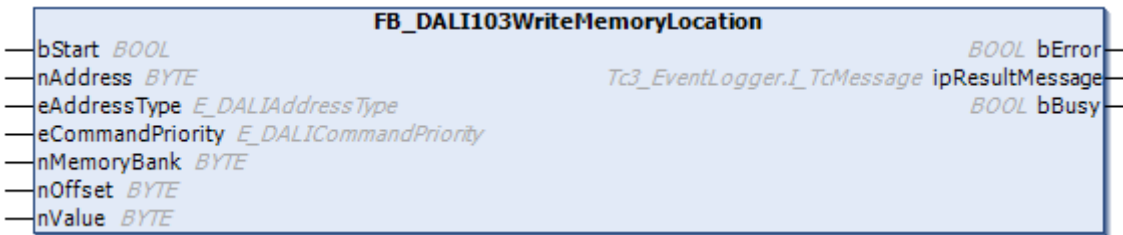
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.2.3.11 FB\_DALI103WriteMemoryLocation



The value *nValue* is written in the memory bank of the DALI control device. The corresponding memory bank is specified by the input variable *nMemoryBank*, the address within the memory bank by the input variable *nOffset*.

The DALI commands ENABLE WRITE MEMORY and WRITE MEMORY LOCATION are called internally.

In contrast to the function block [FB\\_DALI103WriteMemoryLocationNoReply](#) [[▶ 227](#)], the function block [FB\\_DALI103WriteMemoryLocation](#) checks that writing to the memory bank is correct.

**i** The function block changes the DTR0 and the DTR1 of all connected DALI control devices.

**i** The function block sends the DALI command ENABLE WRITE MEMORY twice, as foreseen by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

 Inputs

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMemoryBank     : BYTE := 0;
  nOffset         : BYTE := 0;
  nValue          : BYTE;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMemoryBank	BYTE	Specifies the memory bank for the write access.
nOffset	BYTE	Address within the memory bank selected with the variable <i>nMemoryBank</i> .
nValue	BYTE	Value to be written to the address within the memory bank.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

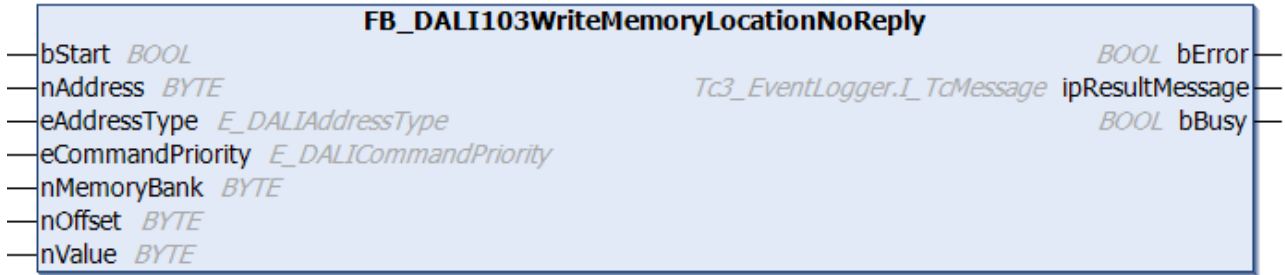
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

4.1.2.2.3.12 FB\_DALI103WriteMemoryLocationNoReply



The value *nValue* is written in the memory bank of the DALI control device. The corresponding memory bank is specified by the input variable *nMemoryBank*, the address within the memory bank by the input variable *nOffset*.

The DALI commands ENABLE WRITE MEMORY and WRITE MEMORY LOCATION – NO REPLY are called internally.

In contrast to the function block FB\_DALI103WriteMemoryLocation [▶ 225], the function block FB\_DALI103WriteMemoryLocationNoReply does not check that writing to the memory bank is correct.

**i** The function block changes the DTR0 and the DTR1 of all connected DALI control devices.

**i** The function block sends the DALI command ENABLE WRITE MEMORY twice, as foreseen by the DALI standard for certain DALI commands.

**i** The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nMemoryBank : BYTE := 0;
  nOffset     : BYTE := 0;
  nValue      : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nMemoryBank	BYTE	Specifies the memory bank for the write access.
nOffset	BYTE	Address within the memory bank selected with the variable <i>nMemoryBank</i> .
nValue	BYTE	Value to be written to the address within the memory bank.

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [ <a href="#">▶ 887</a> ]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [ <a href="#">▶ 873</a> ]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.0	Tc3_DALI from v3.1.5.0

#### 4.1.2.2.4 Device variables

Each DALI control device has a certain number of device variables (parameters). These device variables are used to configure the DALI control device and thus influence its behavior. The values of the device variables are stored in the respective DALI control device.



Name	Reset value	Scope	Size	Comment
<a href="#">shortAddress</a> [ <a href="#">▶ 229</a> ]	No change	0...63, 255	1 byte	255 (MASK): no address
<a href="#">searchAddress</a> [ <a href="#">▶ 229</a> ]	16#FF_FF_FF	16#00_00_00... 16#FF_FF_FF	3 bytes	
<a href="#">randomAddress</a> [ <a href="#">▶ 229</a> ]	16#FF_FF_FF	16#00_00_00... 16#FF_FF_FF	3 bytes	
<a href="#">DTR0</a> [ <a href="#">▶ 229</a> ]	No change	0...255	1 byte	
<a href="#">DTR1</a> [ <a href="#">▶ 229</a> ]	No change	0...255	1 byte	
<a href="#">DTR2</a> [ <a href="#">▶ 229</a> ]	No change	0...255	1 byte	
<a href="#">numberOfInstances</a> [ <a href="#">▶ 230</a> ]	No change	0...32	1 byte	Read only
<a href="#">operatingMode</a> [ <a href="#">▶ 230</a> ]	No change	0, 128...255	1 byte	
<a href="#">quiescentMode</a> [ <a href="#">▶ 230</a> ]	DISABLED	ENABLED, DISABLED	1 bit	
<a href="#">applicationActive</a> [ <a href="#">▶ 230</a> ]	No change	TRUE, FALSE	1 bit	
<a href="#">powerCycleSeen</a> [ <a href="#">▶ 230</a> ]	FALSE	TRUE, FALSE	1 bit	Read only
<a href="#">applicationControllerError</a> [ <a href="#">▶ 230</a> ]	FALSE	TRUE, FALSE	1 bit	Read only
<a href="#">inputDeviceError</a> [ <a href="#">▶ 231</a> ]	FALSE	TRUE, FALSE	1 bit	Read only
<a href="#">resetState</a> [ <a href="#">▶ 231</a> ]	TRUE	TRUE, FALSE	1 bit	Read only

**shortAddress**

The short address is stored in this variable. A valid short address lies in the range between 0 and 63. With a value of 255 (MASK), the short address is considered deleted.

The short address is set with the function block [FB\\_DALI103SetShortAddress](#) [[▶ 177](#)].

**searchAddress**

The search address is only required for the assignment of short addresses.

The value of the variable can be changed with the function block [FB\\_DALI103SetSearchAddress](#) [[▶ 140](#)].

**randomAddress**

The random address, also known as the long address, is only required when short addresses are assigned.

With the help of the function block [FB\\_DALI103QueryRandomAddress](#) [[▶ 207](#)] the 3 bytes long random address can be read.

**DTR0...DTR2**

Data Transfer Registers (DTR) 0 to 2 serve as buffer for the execution of various DALI commands.

The DTRs are used by different function blocks. The description of the respective function block indicates which DTRs are used.

The values of the DTRs can be read out with the function blocks [FB\\_DALI103QueryContentDTRx](#) [[▶ 182](#)] and changed with the function blocks [FB\\_DALI103SetDTRx](#) [[▶ 216](#)].

**numberOfInstances**

The variable indicates how many instances a DALI control device supports. A device has at least one and a maximum of 32 instances.

The number of instances is queried with the function block [FB\\_DALI103QueryNumberOfInstances](#) [► 202].

The type of instance is queried with the function block [FB\\_DALI103QueryInstanceType](#) [► 201].

**operatingMode**

The variable indicates the current operating mode of the DALI control device. The value 0 characterizes the standard operation mode, i.e. the device behaves according to its specification.

The operating modes 0 to 127 are reserved and must not be used.

The operating modes 128 to 255 are vendor-specific.

The operation mode can be read out from the DALI control device with the function block [FB\\_DALI103QueryOperatigMode](#) [► 204] and changed with the function block [FB\\_DALI103SetOperatingMode](#) [► 175].

**quiescentMode**

The variable indicates whether the DALI control device is in quiescent mode. If the variable is TRUE, no commands and events are passed on.

The mode is time-limited to 15 min +/- 1.5 min.

The mode can be started with the function block [FB\\_DALI103StartQuiescentMode](#) [► 179] and terminated prematurely with the function block [FB\\_DALI103StopQuiescentMode](#) [► 180].

The variable can be read out with the aid of the function block [FB\\_DALI103QueryDeviceStatus](#) [► 186].

**applicationActive**

The variable specifies whether the application controller (DALI controller) is active.

According to IEC 62386-103, DALI control devices can contain one or more DALI sensors (input devices) and optionally a DALI controller. If a DALI controller is contained within a DALI control device, it can independently send DALI commands to other control devices and/or DALI control gears (see [DALI](#) [► 9]).

If the application controller is disabled, it does not send any DALI frames other than the Power Cycle Notification.

The application controller can be enabled with the function block [FB\\_DALI103EnableApplicationController](#) [► 159] and disabled with the function block [FB\\_DALI103DisableApplicationController](#) [► 155].

**powerCycleSeen**

The variable indicates whether the DALI control device was supplied with power.

In this case, the bit is TRUE.

The variable can be read out with the aid of the function block [FB\\_DALI103QueryDeviceStatus](#) [► 186].

**applicationControllerError**

This variable specifies whether an application controller (DALI controller) of a DALI control device has detected an error.

In this case, the bit is TRUE.

If the DALI control device does not contain an application controller, this variable is always FALSE.

The variable can be read out with the aid of the function block [FB\\_DALI103QueryDeviceStatus](#) [► 186].

**inputDeviceError**

The variable indicates whether a DALI control device has detected an error on a DALI sensor (input device).

In this case, the bit is TRUE.

The variable can be read out with the aid of the function block [FB\\_DALI103QueryDeviceStatus \[▶ 186\]](#).

The function block [FB\\_DALI103QueryInputDeviceError \[▶ 193\]](#) provides more detailed information about the present error. The output depends on the manufacturer.

**resetState**

The variable indicates whether all device and instance variables have been set to their reset value (see table above).

In this case, the bit is TRUE.

The variable can be read out with the aid of the function block [FB\\_DALI103QueryDeviceStatus \[▶ 186\]](#).

**4.1.2.2.5 Instance variables**

Each DALI control device has at least one and a maximum of 32 instances (see variable *numberOfInstances*). Each instance has a certain number of instance variables (parameters), irrespective of the other instances. These instance variables are used to configure the respective instance in the DALI control device and thus influence its behavior. The values of the instance variables are stored in the respective DALI control device.

Name	Reset value	Scope	Size	Comment
<a href="#">instanceActive [▶ 232]</a>	No change	TRUE, FALSE	1 bit	
<a href="#">instanceType [▶ 232]</a>	No change	0...31	1 byte	Read only
<a href="#">resolution [▶ 232]</a>	No change	1...255	1 byte	Read only
<a href="#">inputValue [▶ 232]</a>	No change	0...(2 <sup>N</sup> -1)	N bytes (max. 32 bytes)	Read only
<a href="#">eventFilter [▶ 232]</a>	16#FF_FF_FF	16#00_00_00...16#FF_FF_FF	3 bytes	Other values can be defined for individual instance types. Details can be taken from the IEC 62386-3xx standard.
<a href="#">eventScheme [▶ 233]</a>	<i>E_DALIEventScheme.Instance</i>	<i>E_DALIEventScheme.Instance</i> ... <i>E_DALIEventScheme.InstanceGroup</i>	1 byte	
<a href="#">eventPriority [▶ 233]</a>	No change	<i>E_DALIEventPriority.Low</i> ... <i>E_DALIEventPriority.MiddleHigh</i>	1 byte	Other values can be defined for individual instance types. Details can be taken from the IEC 62386-3xx standard.
<a href="#">instanceError [▶ 233]</a>	FALSE	TRUE, FALSE	1 bit	Read only

N: resolution

### instanceActive

The variable indicates whether sending of events has been enabled for the respective instance.

If the variable is FALSE, the instance does not send any events. This variable has no effect on the DALI commands sent to the instance by other DALI controllers. These will be answered even if *instanceActive* is FALSE.

The function block [FB\\_DALI103EnableInstance \[► 161\]](#) is used to enable the sending of events and the function block [FB\\_DALI103DisableInstance \[► 156\]](#) is used to disable the sending of events.

The value of the variable can be read out with the function block [FB\\_DALI103QueryInstanceEnabled \[► 196\]](#).

### instanceType

The variable contains the respective instance type of an instance.

Instance type	IEC 62386-	Application
0	103	General application and control devices; not defined further
1-31	301-331	These IEC 62386-3xx parts describe instance types from 301 to 331

The instance type (see [E\\_DALIInstanceType \[► 862\]](#)) of an instance can be read using the function block [FB\\_DALI103QueryInstanceType \[► 201\]](#).

### resolution

The variable indicates the resolution of [inputValue \[► 232\]](#).

The resolution corresponds to the number of bits through which the input value is mapped.

The function block [FB\\_DALI103QueryResolution \[► 208\]](#) can be used to read the resolution.

### inputValue

The variable outputs the input value of an instance.

The number of bits in which the input value is mapped can be queried through the variable [resolution \[► 232\]](#).

The input value can be read out with the function block [FB\\_DALI103QueryInputValue \[► 194\]](#).

### eventFilter

The event filter enables or disables specific events for each instance. The meanings of the individual bits for the respective instance types can be found here:

Part 301 (push buttons) – [Introduction \[► 512\]](#)

Part 302 (absolute encoder) - [Introduction \[► 532\]](#)

Part 303 (occupancy sensors) - [Introduction \[► 543\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The function block [FB\\_DALI103QueryEventFilter \[► 188\]](#) can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter \[► 170\]](#) can be used to write to it.

**eventScheme**

The variable contains the event scheme for the respective instance of a DALI control device.

<i>E_DALIEventScheme</i>	Description
Instance	Instance addressing with instance type and number.
Device	Device addressing with short address and instance type.
DeviceInstance	Device/instance addressing with short address and instance number.
DeviceGroup	Device group addressing with device group and instance type.
InstanceGroup	Instance group addressing with instance group and type.

The event scheme can be read with the function block [FB\\_DALI103QueryEventScheme \[▶ 191\]](#) and written with the function block [FB\\_DALI103SetEventScheme \[▶ 174\]](#).

**eventPriority**

Different [Priorities \[▶ 11\]](#) for sending events can be configured for each instance. The device function blocks use the following default values for event priorities:

<i>E_DALIEventPriority</i>	Application
MiddleHigh	
Middle	Part 301 (push button), Part 302 (absolute encoder), Part 303 (occupancy sensor)
MiddleLow	Part 304 (light sensors)
Low	

The value of the event priority can be read with the function block [FB\\_DALI103QueryEventPriority \[▶ 190\]](#) and written with the function block [FB\\_DALI103SetEventPriority \[▶ 172\]](#).

**instanceError**

The variable indicates whether an instance of a control device has detected an error.

In this case, the bit is TRUE.

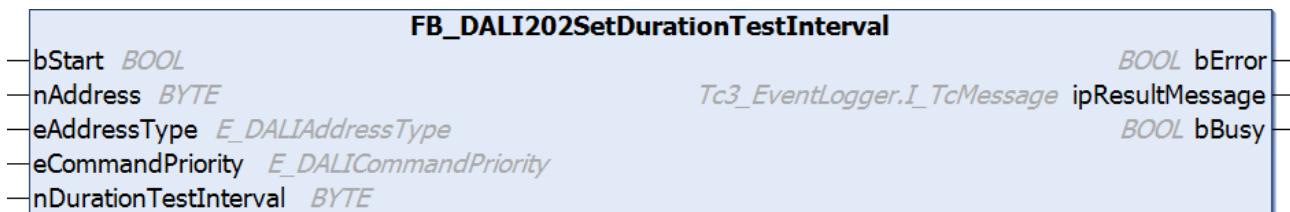
The variable can be read out with the aid of the function block [FB\\_DALI103QueryInstanceStatus \[▶ 199\]](#).

The function block [FB\\_DALI103QueryInstanceError \[▶ 198\]](#) provides more detailed information about the present error. The output depends on the manufacturer.

**4.1.2.3 Part 202 (emergency lighting)**

**4.1.2.3.1 Configuration commands**

**4.1.2.3.1.1 FB\_DALI202SetDurationTestInterval**



The function block stores the value *nDurationTestInterval* in the variable [durationTestInterval \[▶ 282\]](#) of the DALI control gear.

**i** Prior to the DALI command SET DURATION TEST INTERVAL, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [[▶ 130](#)]).

**i** The function block sends the DALI command SET DURATION TEST INTERVAL twice, as required by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eDimmingCurve   : E_DALIDimmingCurve := E_DALIDimmingCurve.Standard;
  nDurationTestInterval : BYTE := 52;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDurationTestInterval	BYTE	Interval time for the duration test (see <a href="#">durationTestInterval</a> [ <a href="#">▶ 282</a> ])

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

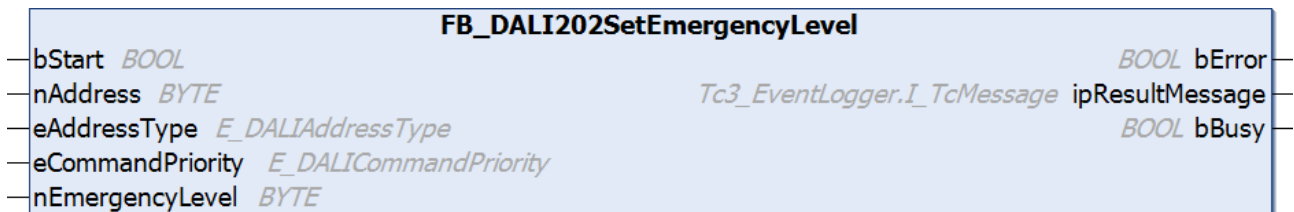
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.1.2 FB\_DALI202SetEmergencyLevel**



The function block stores the value *nEmergencyLevel* in the variable [emergencyLevel \[▶ 280\]](#) of the DALI control gear.

- i** Prior to the DALI command SET EMERGENCY LEVEL, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).
- i** The function block sends the DALI command SET EMERGENCY LEVEL twice, as required by the DALI standard for certain DALI commands.
- i** The function block changes the DTR0 of all connected DALI control gears.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  
```

```
eDimmingCurve      : E_DALIDimmingCurve := E_DALIDimmingCurve.Standard;
nEmergencyLevel    : BYTE := 254;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nEmergencyLevel	BYTE	Output value from DALI control gear in emergency mode (see <a href="#">emergencyLevel</a> [▶ 280]).

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0



4.1.2.3.1.3 FB\_DALI202SetFunctionTestInterval

FB_DALI202SetFunctionTestInterval		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	
nFunctionTestInterval	BYTE	

The function block stores the value *nFunctionTestInterval* in the variable functionTestInterval [► 281] of the DALI control gear.

**i** Prior to the DALI command SET FUNCTION TEST INTERVAL, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [► 130]).

**i** The function block sends the DALI command SET FUNCTION TEST INTERVAL twice, as required by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

 Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eDimmingCurve   : E_DALIDimmingCurve := E_DALIDimmingCurve.Standard;
  nfunctionTestInterval : BYTE := 7;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<u>E_DALIAddressType</u> [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<u>E_DALICommandPriority</u> [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nFunctionTestInterval	BYTE	Interval time for the function test (see <u>functionTestInterval</u> [► 281]).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

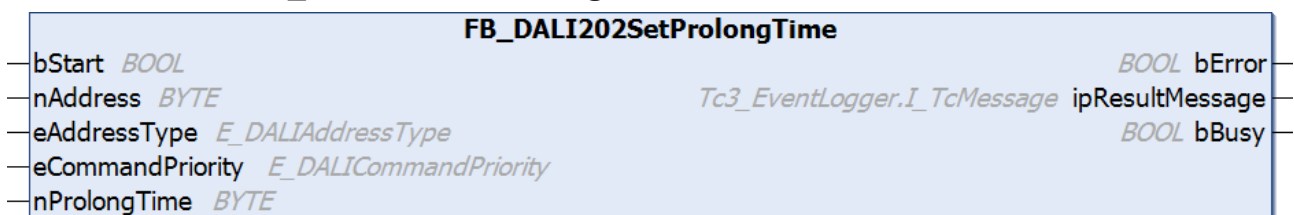
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.1.4 FB\_DALI202SetProlongTime**



The function block stores the value *nProlongTime* in the variable *prolongTime* [▶ 281] of the DALI control gear.

**i** Prior to the DALI command SET PROLONG TIME, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

**i** The function block sends the DALI command SET PROLONG TIME twice, as required by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eDimmingCurve   : E_DALIDimmingCurve := E_DALIDimmingCurve.Standard;
  nProlongTime    : BYTE := 0;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nProlongTime	BYTE	Time period for prolongation of emergency mode after return of mains voltage (see <a href="#">prolongTime</a> [▶ 281]).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### 4.1.2.3.1.5 FB\_DALI202SetTestDelayTime

FB_DALI202SetTestDelayTime		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	
nTestDelayTime	UINT	

The function block stores the value *nDelayTestTime* in the variable `testDelayTime` [► 281] of the DALI control gear.

**i** Prior to the DALI commands SET TEST DELAY TIME LSB and SET TEST DELAY TIME MSB, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [► 130]).

**i** The function block sends the DALI commands SET TEST DELAY TIME LSB and SET TEST DELAY TIME MSB twice, as required by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

#### Inputs

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eDimmingCurve   : E_DALIDimmingCurve := E_DALIDimmingCurve.Standard;
  nTestDelayTime  : UINT := 0;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nTestDelayTime	UINT	Value written to variable <code>testDelayTime</code> [► 281].

#### Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

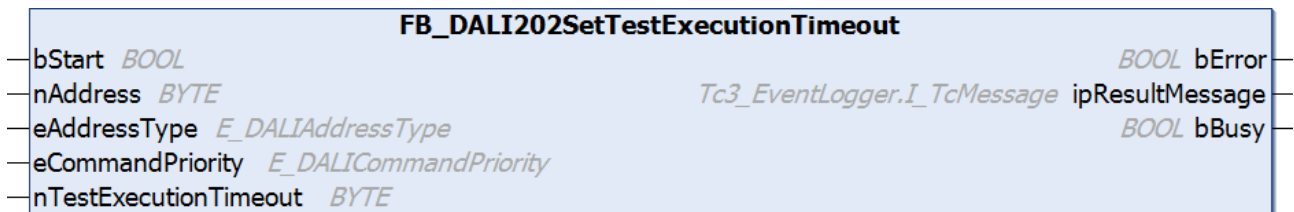
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.1.6 FB\_DALI202SetTestExecutionTimeout**



The function block stores the value *nTestExecutionTimeout* in the variable [testExecutionTimeout](#) [▶ 282] of the DALI control gear.

- i** Prior to the DALI command SET TEST EXECUTION TIMEOUT, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

---

- i** The function block sends the DALI command SET TEST EXECUTION TIMEOUT twice, as required by the DALI standard for certain DALI commands.

---

- i** The function block changes the DTR0 of all connected DALI control gears.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  
```

```
eCommandPriority      : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
eDimmingCurve        : E_DALIDimmingCurve := E_DALIDimmingCurve.Standard;
nTestExecutionTimeout : BYTE := 7;
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nTestExecutionTime out	BYTE	Maximum execution time for the tests (see <a href="#">testExecutionTimeout</a> [▶ 282]).

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

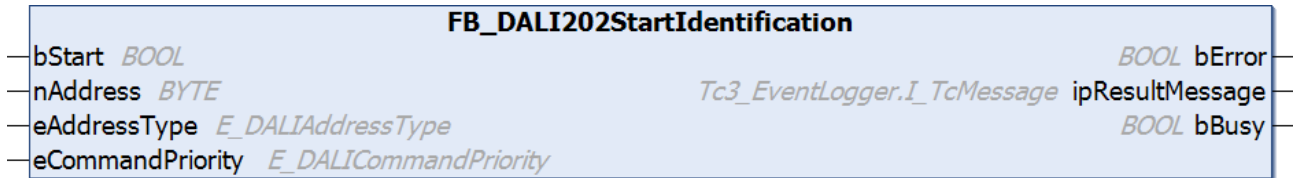
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

4.1.2.3.1.7 FB\_DALI202StartIdentification



The function block starts or extends the identification routine of the DALI control gear.

The routine takes approx. 10 s and ends automatically. The exact scope of the identification routine depends on the manufacturer of the DALI control gear.

**i** Prior to the DALI command START IDENTIFICATION, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

**i** The function block sends the DALI command START IDENTIFICATION twice, as required by the DALI standard for certain DALI commands.

 Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

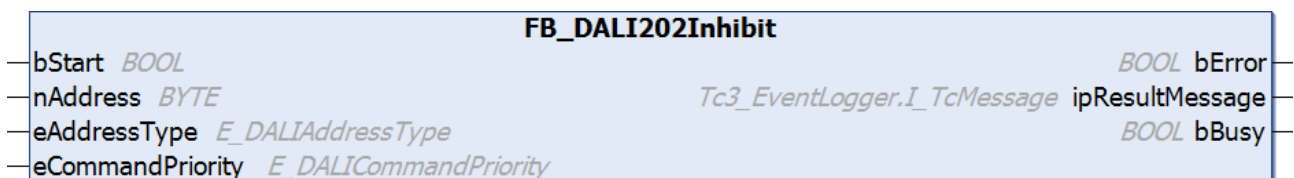
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.2 Control commands**

**4.1.2.3.2.1 FB\_DALI202Inhibit**



This function block prevents the DALI control gear from switching to emergency mode for 15 minutes.

The function block [FB\\_DALI202ReLightResetInhibit](#) [▶ 247] can be used to disable inhibition of emergency mode.

**i** Prior to the DALI command INHIBIT, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

**i** The function block sends the DALI command INHIBIT twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

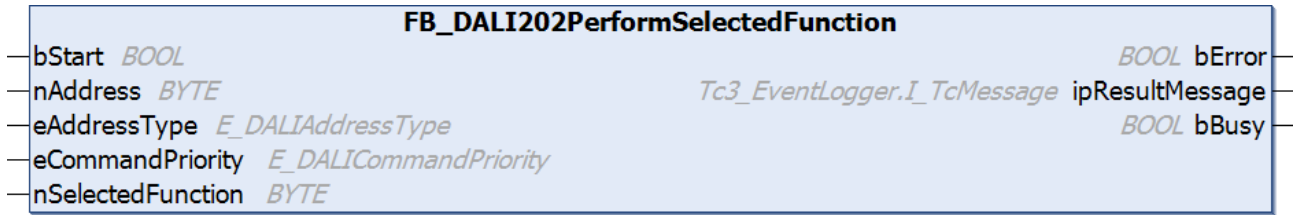
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

### 4.1.2.3.2.2 FB\_DALI202PerformSelectedFunction



This function block executes certain actions in the DALI control gear depending on the input *nSelectedFunction*.

The input *nSelectedFunction* currently defines the following actions:

nSelectedFunction	Description
0	All variables are reset to their respective reset values (see <a href="#">variables [► 280]</a> ).

**i** Prior to the DALI command PERFORM DTR SELECTED FUNCTION, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[► 130\]](#)).

**i** The function block sends the DALI command PERFORM DTR SELECTED FUNCTION twice, as required by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nSelectedFunction : BYTE;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType [► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority [► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nSelectedFunction	BYTE	Defines the action that the function block should perform (see table above).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

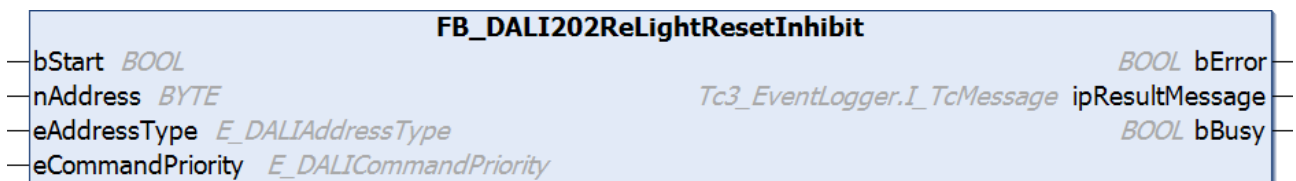
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.2.3 FB\_DALI202ReLightResetInhibit**



This function block is used to switch the DALI control gear back to emergency mode in the absence of mains voltage.

This deactivates the function of the function block [FB\\_DALI202Inhibit \[▶ 244\]](#).

**i** Prior to the DALI command RE-LIGHT/RESET INHIBIT, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

**i** The function block sends the DALI command RE-LIGHT/RESET INHIBIT twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

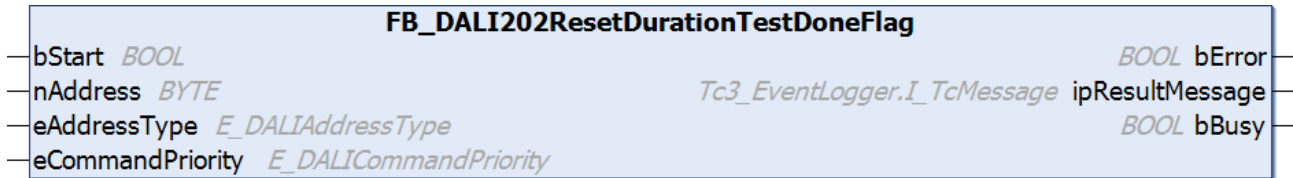
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

4.1.2.3.2.4 FB\_DALI202ResetDurationTestDoneFlag



This function block resets bit 2 of the variable `emergencyStatus` [▶ 284].

**i** Prior to the DALI command RESET DURATION TEST DONE FLAG, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

**i** The function block sends the DALI command RESET DURATION TEST DONE FLAG twice, as required by the DALI standard for certain DALI commands.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

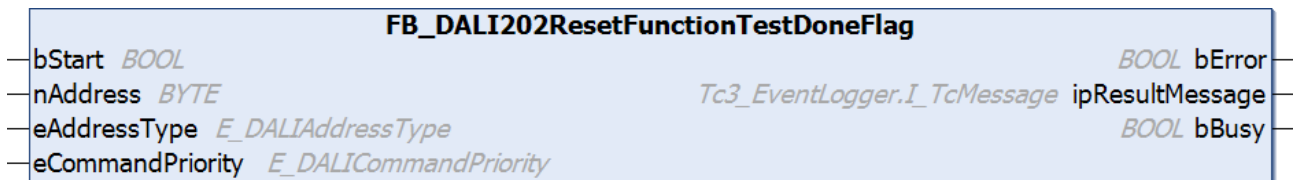
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.2.5 FB\_DALI202ResetFunctionTestDoneFlag**



This function block resets bit 1 of the variable [emergencyStatus \[▶ 284\]](#).

**i** Prior to the DALI command RESET FUNCTION TEST DONE FLAG, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

**i** The function block sends the DALI command RESET FUNCTION TEST DONE FLAG twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

### 4.1.2.3.2.6 FB\_DALI202ResetLampTime

FB_DALI202ResetLampTime		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	

The function block sets the variables `lampEmergencyTime` [▶ 282] and `lampTotalOperationTime` [▶ 283] to 0.

**i** Prior to the DALI command RESET LAMP TIME, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [▶ 130]).

**i** The function block sends the DALI command RESET LAMP TIME twice, as required by the DALI standard for certain DALI commands.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices ( <code>BroadcastUnaddr</code> ) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

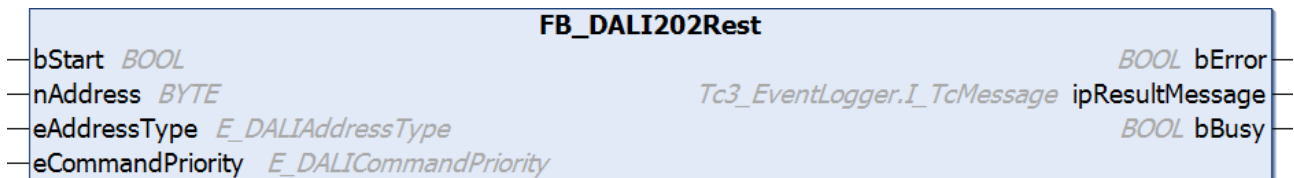
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.2.7 FB\_DALI202Rest**



This function block is used to switch off the lamp when the DALI control gear is in emergency mode.

Otherwise the system switches to normal operation if mains voltage is available again, or if the function block [FB\\_DALI202ReLightResetInhibit \[▶ 247\]](#) was called.

**i** Prior to the DALI command REST, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

**i** The function block sends the DALI command REST twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

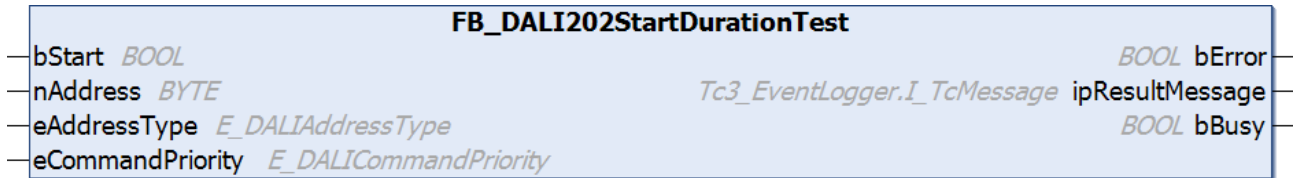
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

4.1.2.3.2.8 FB\_DALI202StartDurationTest



The function block starts a duration test.

If the duration test is started with a delay, this is indicated by bit 5 of the variable `emergencyStatus` [► 284].

**i** Prior to the DALI command START DURATION TEST, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [► 130]).

**i** The function block sends the DALI command START DURATION TEST twice, as required by the DALI standard for certain DALI commands.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

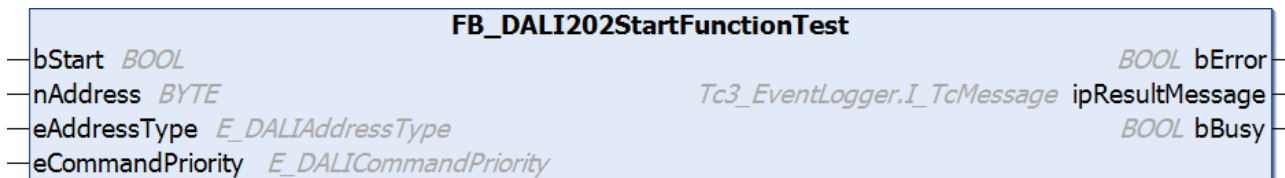
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.2.9 FB\_DALI202StartFunctionTest**



The function block starts a function test.

If the function test is started with a delay, this is indicated by bit 4 of the variable [emergencyStatus \[▶ 284\]](#).

**i** Prior to the DALI command START FUNCTION TEST, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

**i** The function block sends the DALI command START FUNCTION TEST twice, as required by the DALI standard for certain DALI commands.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

### 4.1.2.3.2.10 FB\_DALI202StopTest



The function block stops a function or duration test.

Since all tests are stopped by the function block, bit 4 and bit 5 are also reset in the variable `emergencyStatus` [▶ 284].



Prior to the DALI command STOP TEST, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).



The function block sends the DALI command STOP TEST twice, as required by the DALI standard for certain DALI commands.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices ( <code>BroadcastUnaddr</code> ) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

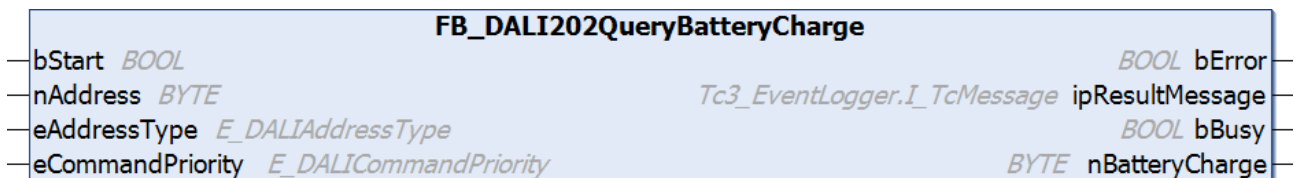
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.3 Query commands**

**4.1.2.3.3.1 FB\_DALI202QueryBatteryCharge**



The function block reads the variable [batteryCharge \[► 282\]](#) from the DALI control gear.

bError is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY BATTERY CHARGE, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[► 130\]](#)).

The output *nBatteryCharge* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nBatteryCharge  : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nBatteryCharge	BYTE	Battery charge status 0: empty / 254: full (see <a href="#">batteryCharge [▶ 282]</a> ).

### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0



### 4.1.2.3.3.2 FB\_DALI202QueryDurationTestResult

FB_DALI202QueryDurationTestResult		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BYTE nDurationTestResult

The function block reads the variable `durationTestResult` [▶ 282] from the DALI control gear.

**i** Prior to the DALI command QUERY DURATION TEST RESULT, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

The output `nDurationTestResult` only contains a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nDurationTestResult : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nDurationTestResult	BYTE	Result of the duration test in steps of 2 minutes (see <a href="#">durationTestResult [▶ 282]</a> ).

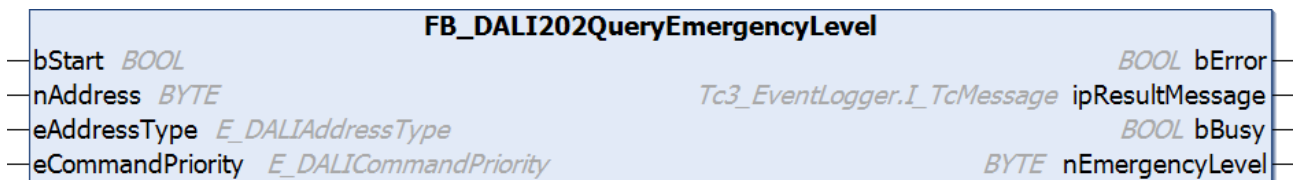
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.3.3 FB\_DALI202QueryEmergencyLevel**



The function block reads the variable [emergencyLevel \[▶ 280\]](#) from the DALI control gear.

*bError* is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY EMERGENCY LEVEL, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nEmergencyLevel* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nEmergencyLevel : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nEmergencyLevel	BYTE	Output value from DALI control gear in emergency mode (see <u>emergencyLevel</u> [▶ 280]).

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

### 4.1.2.3.3.4 FB\_DALI202QueryEmergencyMaxLevel

FB_DALI202QueryEmergencyMaxLevel		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BYTE nEmergencyMaxLevel

The function block reads the variable `emergencyMaxLevel` [▶ 281] from the DALI control gear.

`bError` is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY EMERGENCY MAX LEVEL, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

The output `nEmergencyMaxLevel` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nEmergencyMaxLevel : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nEmergencyMaxLevel	BYTE	Maximum output value from DALI control gear in emergency mode (see <a href="#">emergencyMaxLevel</a> [▶ 281]).

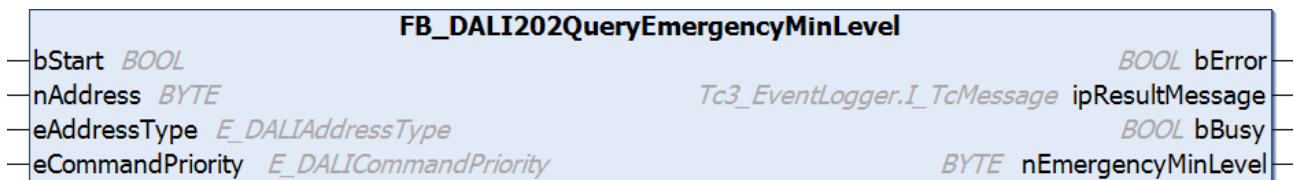
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

4.1.2.3.3.5 FB\_DALI202QueryEmergencyMinLevel



The function block reads the variable [emergencyMinLevel](#) [▶ 281] from the DALI control gear.

bError is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY EMERGENCY MIN LEVEL, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nEmergencyMinLevel* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 Inputs

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nEmergencyMinLevel : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nEmergencyMinLevel	BYTE	Minimum output value of the DALI control gear in emergency mode (see <a href="#">emergencyMinLevel ▶ 281</a> )



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block ▶ 892</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

4.1.2.3.3.6 FB\_DALI202QueryEmergencyMode

FB_DALI202QueryEmergencyMode		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BYTE nEmergencyMode

The function block reads the variable `emergencyMode` [▶ 283] from the DALI control gear.

**i** Prior to the DALI command QUERY EMERGENCY MODE, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

The output `nEmergencyMode` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nEmergencyMode : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nEmergencyMode	BYTE	Information about the current operating mode of the DALI control gear (see <a href="#">emergencyMode</a> [▶ 283]).

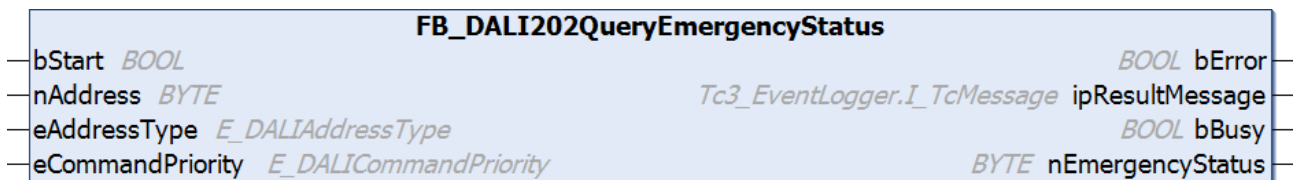
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.3.7 FB\_DALI202QueryEmergencyStatus**



The function block reads the variable [emergencyStatus](#) [▶ 284] from the DALI control gear.

**i** Prior to the DALI command QUERY EMERGENCY STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nEmergencyStatus* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bBusy            : BOOL;
  nEmergencyStatus : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nEmergencyStatus	BYTE	Current information on emergency mode (see <u>emergencyStatus</u> [▶ 284]).

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

### 4.1.2.3.3.8 FB\_DALI202QueryFailureStatus

FB_DALI202QueryFailureStatus		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BYTE nFailureStatus

The function block reads the variable `failureStatus` [▶ 284] from the DALI control gear.

**i** Prior to the DALI command QUERY FAILURE STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

The output `nFailureStatus` only contains a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nFailureStatus : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nFailureStatus	BYTE	Returns the failure status (see <a href="#">failureStatus</a> [▶ 284]).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.3.9 FB\_DALI202QueryFeatures**

FB_DALI202QueryFeatures	
bStart <i>BOOL</i>	<i>BOOL</i> bError
nAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
eAddressType <i>E_DALIAddressType</i>	<i>BOOL</i> bBusy
eCommandPriority <i>E_DALICommandPriority</i>	<i>BYTE</i> nFeatures

The function block reads the implemented properties from the DALI control gear (see [features](#) [▶ 283]).



Prior to the DALI command QUERY FEATURES, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nFeatures* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nFeatures       : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nFeatures	BYTE	Returns information about the implemented features (see <a href="#">features [▶ 283]</a> ).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

### 4.1.2.3.3.10 FB\_DALI202QueryLampEmergencyTime

FB_DALI202QueryLampEmergencyTime		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BYTE nLampEmergencyTime

The function block reads the variable `lampEmergencyTime` [▶ 282] from the DALI control gear.

The variable can be reset again using the function block `FB_DALI202ResetLampTime` [▶ 252].

**i** Prior to the DALI command QUERY LAMP EMERGENCY TIME, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [▶ 130]).

The output `nLampEmergencyTime` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nLampEmergencyTime : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLampEmergencyTime	BYTE	Emergency operation time of the lamp from the DALI control gear (see <a href="#">lampEmergencyTime [▶ 282]</a> ).

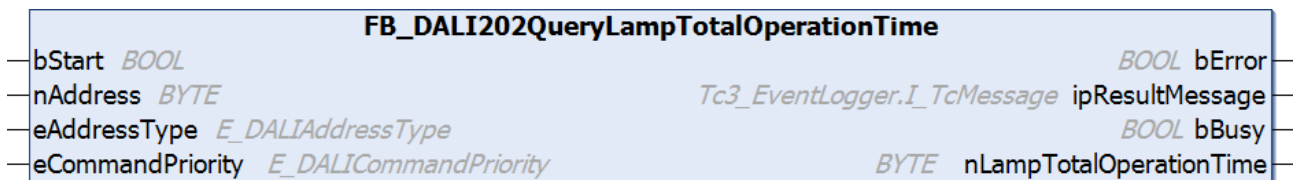
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.3.11 FB\_DALI202QueryLampTotalOperationTime**



The function block reads the variable [lampTotalOperationTime \[▶ 283\]](#) from the DALI control gear.

The variable can be reset again using the function block [FB\\_DALI202ResetLampTime \[▶ 252\]](#).

**i** Prior to the DALI command QUERY LAMP TOTAL OPERATING TIME, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nLampTotalOperationTime* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bBusy            : BOOL;
  nLampTotalOperationTime : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLampTotalOperationTime	BYTE	Total operation time of the lamp from the DALI control gear (see <u>lampTotalOperationTime</u> [▶ 283]).

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

### 4.1.2.3.3.12 FB\_DALI202QueryRatedDuration

FB_DALI202QueryRatedDuration		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BYTE nRatedDuration

The function block reads the variable `ratedDuration` [► 283] from the DALI control gear.

**i** Prior to the DALI command QUERY RATED DURATION, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [► 130]).

The output `nRatedDuration` only contains a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nRatedDuration : BYTE;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRatedDuration	BYTE	Nominal operating time from DALI control gear (see <a href="#">ratedDuration [▶ 283]</a> ).

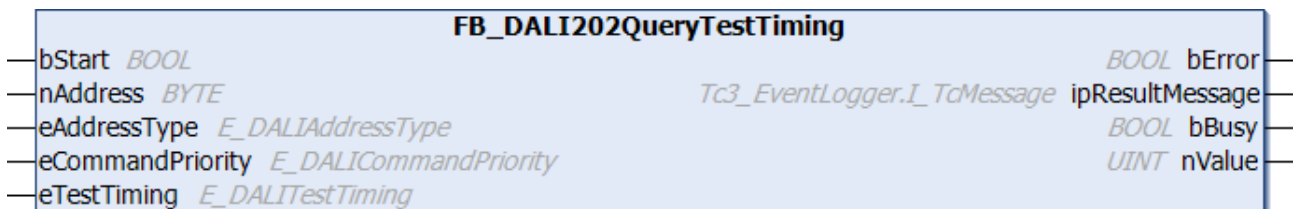
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.3.3.13 FB\_DALI202QueryTestTiming**



Depending on the input *eTestTiming*, this function block reads out the settings for the function and duration test from the DALI control gear.

The *eTestTiming* input currently defines the following actions:

eTestTiming	Meaning
FunctionTestDelayTime	The value of the variable <a href="#">functionTestDelayTime [▶ 281]</a> is read.
DurationTestDelayTime	The value of the variable <a href="#">durationTestDelayTime [▶ 281]</a> is read.
FunctionTestInterval	The value of the variable <a href="#">functionTestInterval [▶ 281]</a> is read.
DurationTestInterval	The value of the variable <a href="#">durationTestInterval [▶ 282]</a> is read out.
TestExecutionTimeout	The value of the variable <a href="#">testExecutionTimeout [▶ 282]</a> is read.
ProlongTime	The value of the variable <a href="#">prolongTime [▶ 281]</a> is read.

**i** Prior to the DALI command QUERY TEST TIMING, the function block sends the DALI command ENABLE DEVICE TYPE 1, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).



The function block changes the DTR0 of all connected DALI control gears.

The output *nValue* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eTestTiming     : E_DALITestTiming := E_DALITestTiming.TimeUntilNextFunctionTestMSB;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eTestTiming	E_DALITestTiming [▶ 863]	Defines the variable that the function block should read from the DALI control gear (see table above).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nValue          : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nValue	UINT	The value that was read out (see table above).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [►_892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

#### 4.1.2.3.4 Variables

Name	Reset value	Scope	Size	Comment
<a href="#">emergencyLevel</a> [▶ 280]	No change	emergencyMinLevel ... emergencyMaxLevel or 255 (MASK)	1 byte	
<a href="#">emergencyMinLevel</a> [▶ 281]	No change	1... emergencyMaxLevel or 255 (MASK)	1 byte	read only
<a href="#">emergencyMaxLevel</a> [▶ 281]	No change	emergencyMinLevel ... 254 or 255 (MASK)MASK	1 byte	read only
<a href="#">prolongTime</a> [▶ 281]	No change	0...255	1 byte	
<a href="#">testDelayTime</a> [▶ 281]	No change	0...65535	2 byte	write only
<a href="#">functionTestDelayTime</a> [▶ 281]	No change	0...65535	2 byte	
<a href="#">durationTestDelayTime</a> [▶ 281]	No change	0...65535	2 byte	
<a href="#">functionTestInterval</a> [▶ 281]	No change	0...255	1 byte	
<a href="#">durationTestInterval</a> [▶ 282]	No change	0...97	1 byte	
<a href="#">testExecutionTimeout</a> [▶ 282]	No change	0...255	1 byte	
<a href="#">batteryCharge</a> [▶ 282]	No change	0...255	1 byte	read only
<a href="#">durationTestResult</a> [▶ 282]	No change	0...255	1 byte	read only
<a href="#">lampEmergencyTime</a> [▶ 282]	No change	0...255	1 byte	
<a href="#">lampTotalOperationTime</a> [▶ 283]	No change	0...255	1 byte	
<a href="#">ratedDuration</a> [▶ 283]	No change	0...255	1 byte	read only
<a href="#">emergencyMode</a> [▶ 283]	No change	0...255	1 byte	read only
<a href="#">features</a> [▶ 283]	No change	0...255	1 byte	read only
<a href="#">failureStatus</a> [▶ 284]	No change	0...255	1 byte	read only
<a href="#">emergencyStatus</a> [▶ 284]	No change	0...255	1 byte	read only

#### emergencyLevel

This variable contains the output value when the DALI control gear is in emergency mode.

This value is limited by the variables [emergencyMinLevel](#) [▶ 281] and [emergencyMaxLevel](#) [▶ 281].

If the DALI control gear cannot determine the value, this variable contains the value 255 (MASK).

The value can be read out with the function block [FB\\_DALI202QueryEmergencyLevel](#) [▶ 262] and specified with [FB\\_DALI202SetEmergencyLevel](#) [▶ 235].

### **emergencyMinLevel / emergencyMaxLevel**

The output value in emergency mode is limited within the DALI control gear by the variables *emergencyMinLevel* and *emergencyMaxLevel*.

The values for *emergencyMinLevel* and *emergencyMaxLevel* are fixed by the manufacturer of the DALI control gear and cannot be changed.

If the DALI control gear cannot determine the value, the corresponding variable contains the value 255 (MASK).

The value of the respective variable can be queried with the function blocks [FB\\_DALI102QueryEmergencyMinLevel \[▶ 265\]](#) or [FB\\_DALI102QueryEmergencyMaxLevel \[▶ 264\]](#).

### **prolongTime**

The variable *prolongTime* defines how long emergency mode is extended after mains voltage is available again.

The value is given in steps of 30 s. The permissible value range is 0 (0 s) to 255 (127.5 min).

The value can be read out with the function block [FB\\_DALI202QueryTestTiming \[▶ 277\]](#) and specified with [FB\\_DALI202SetProlongTime \[▶ 238\]](#).

### **testDelayTime**

The variable *testDelayTime* is used to set the variables *functionTestDelayTime* and *durationTestDelayTime*.

This specifies the time behavior for the function test and the duration test.

The value can be specified with the function block [FB\\_DALI202SetTestDelayTime \[▶ 240\]](#).

Bit 3 of [features \[▶ 283\]](#) can be used to query whether this variable is supported by the DALI control gear.

### **functionTestDelayTime**

This variable is used to specify the delay time for the function test. Once this time has elapsed, the function test is executed for the first time.

The value is given in steps of 15 min. The permissible value range is 0 (0 min) to 65534 (16383.5 h).

If automatic testing is not supported, the variable contains the value 65535 (MASK).

This variable can be queried via the function block [FB\\_DALI202QueryTestTiming \[▶ 277\]](#). This variable is described via [testDelayTime \[▶ 281\]](#) and [functionTestInterval \[▶ 281\]](#).

Bit 3 of [features \[▶ 283\]](#) can be used to query whether this variable is supported by the DALI control gear.

### **durationTestDelayTime**

This variable is used to specify the delay time for the duration test. Once this time has elapsed, the duration test is executed for the first time.

The value is given in steps of 15 min. The permissible value range is 0 (0 min) to 65534 (16383.5 h).

If automatic testing is not supported, the variable contains the value 65535 (MASK).

This variable can be queried via the function block [FB\\_DALI202QueryTestTiming \[▶ 277\]](#). This variable is described via [testDelayTime \[▶ 281\]](#) and [durationTestInterval \[▶ 282\]](#).

Bit 3 of [features \[▶ 283\]](#) can be used to query whether this variable is supported by the DALI control gear.

### **functionTestInterval**

This variable is used to specify the interval time for the function test. The function test is executed periodically at these intervals.

The value is given in steps of 1 day. The permissible value range is 1 (1 day) to 255 (255 days). The value 0 deactivates the automatic function test.

If this variable is written with a value greater than 0, the value from *testDelayTime* is transferred to the variable *functionTestDelayTime* and thus the delay time for the next function test is defined.

The value can be read out with the function block [FB\\_DALI202QueryTestTiming \[▶ 277\]](#) and specified with [FB\\_DALI202SetFunctionTestInterval \[▶ 237\]](#).

Bit 3 of [features \[▶ 283\]](#) can be used to query whether this variable is supported by the DALI control gear.

### **durationTestInterval**

This variable is used to specify the interval time for the duration test. The duration test is executed periodically at these intervals.

The value is given in steps of 1 week. The permissible value range is 1 (1 week) to 255 (255 weeks). The value 0 disables the automatic duration test.

If this variable is written with a value greater than 0, the value from *testDelayTime* is transferred to the variable *durationTestDelayTime* and thus the delay time for the next duration test is defined.

The value can be read out with the function block [FB\\_DALI202QueryTestTiming \[▶ 277\]](#) and specified with [FB\\_DALI202SetDurationTestInterval \[▶ 233\]](#).

Bit 3 of [features \[▶ 283\]](#) can be used to query whether this variable is supported by the DALI control gear.

### **testExecutionTimeout**

This variable defines the maximum execution time in which the respective test must be completed.

If the maximum execution time is exceeded during a test, this is indicated in bit 4 or bit 5 of the variable [failureStatus \[▶ 284\]](#).

The value is given in steps of 1 day. The permissible value range is 1 (1 day) to 255 (255 days). The value 0 sets the maximum execution time to 15 min.

The value can be read out with the function block [FB\\_DALI202QueryTestTiming \[▶ 277\]](#) and specified with [FB\\_DALI202SetTestExecutionTimeout \[▶ 241\]](#).

### **batteryCharge**

This variable contains the current state of charge of the battery.

The permissible value range is from 0 (minimum charge) to 254 (maximum charge). If the DALI control gear cannot determine the state of charge, this variable contains the value 255 (MASK).

The state of charge can only be queried after a duration test has been successfully completed.

This variable can be read out with the function block [FB\\_DALI202QueryBatteryCharge \[▶ 259\]](#).

### **durationTestResult**

This variable contains the result of a duration test.

The value is given in steps of 2 min. The permissible value range is 0 (0 min) to 255 (510 min or longer).

The value is only valid if bit 2 is set in the variable *emergencyStatus*.

This variable can be read out with the function block [FB\\_DALI202QueryDurationTestResult \[▶ 261\]](#).

### **lampEmergencyTime**

This variable contains the operation time of the lamp in emergency mode (supply by means of the battery).

The value is given in steps of 1 hour. The permissible value range is 0 (0 h) to 255 (255 h or longer). The variable is always incremented at the start of the 1-hour interval.

This variable can be read out with the function block [FB\\_DALI202QueryLampEmergencyTime \[▶ 273\]](#) and reset with [FB\\_DALI202ResetLampTime \[▶ 252\]](#).

**lampTotalOperationTime**

This variable contains the total operation time of the lamp.

The value is given in steps of 4 hours. The permissible value range is 0 (0 h) to 255 (1020 h or longer). The variable is always incremented at the start of the 4-hour interval.

This variable can be read out with the function block [FB\\_DALI202QueryLampTotalOperationTime \[▶ 274\]](#) and reset with [FB\\_DALI202ResetLampTime \[▶ 252\]](#).

**ratedDuration**

This variable contains the nominal operation time of the battery.

The value is given in steps of 2 min. The permissible value range is 0 (0 min) to 255 (510 min or longer).

The value is specified by the vendor of the DALI control gear and can only be read.

This variable can be read out with the function block [FB\\_DALI202QueryRatedDuration \[▶ 276\]](#).

**emergencyMode**

This variable contains information about the current operation mode of the DALI control gear.

The variable can be read out with the function block [FB\\_DALI202QueryEmergencyMode \[▶ 267\]](#).

Bit	Description
0	Reset mode
1	Emergency mode readiness (normal operation)
2	Emergency mode
3	Extended emergency mode once mains voltage is available again
4	Function test active
5	Duration test active
6	Connected suppress push button is active
7	Connected switch active

**features**

This variable contains information about the implemented optional properties.

The variable can be read out with the function block [FB\\_DALI202QueryFeatures \[▶ 271\]](#).

Bit	Description
0	Integrated emergency lighting supply unit
1	Emergency lighting supply unit in continuous mode
2	Switchable emergency lighting supply unit
3	Automatic testing is supported
4	Intensity of the emergency lighting is adjustable
5	Connected suppress push button is supported
6	Addressing by physical selection is supported
7	Switching back to emergency mode is supported

**failureStatus**

This variable contains possible error states from the function test or the duration test.

The variable can be read out with the function block `FB_DALI202QueryFailureStatus` [► 270].

Bit	Description
0	Error in the DALI control gear circuit
1	Battery operation time fault
2	Battery fault
3	Emergency lamp fault
4	Timeout during function test
5	Timeout during duration test
6	Function test failed
7	Duration test failed

**emergencyStatus**

This variable contains current information about emergency operation from the DALI control gear.

The variable can be read out with the function block `FB_DALI202QueryEmergencyStatus` [► 268].

Bit	Description
0	Inhibit mode
1	Function test is completed and result is valid
2	Duration test is completed and result is valid
3	Battery charger ready for operation
4	Start of function test delayed
5	Start of duration test delayed
6	Identification active
7	Selected during addressing by physical selection

**4.1.2.4 Part 205 (incandescent lamps)**

**4.1.2.4.1 Configuration commands**

**4.1.2.4.1.1 FB\_DALI205SelectDimmingCurve**

**FB\_DALI205SelectDimmingCurve**

<code>bStart</code>	<i>BOOL</i>		<i>BOOL</i> bError
<code>nAddress</code>	<i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i>	ipResultMessage
<code>eAddressType</code>	<i>E_DALIAddressType</i>		<i>BOOL</i> bBusy
<code>eCommandPriority</code>	<i>E_DALICommandPriority</i>		
<code>eDimmingCurve</code>	<i>E_DALIDimmingCurve</i>		

The function block stores the value `eDimmingCurve` in the variable `dimmingCurve` [► 303] of the DALI control gear. This also changes the value of the Delay Time.

**i** Prior to the DALI command SELECT DIMMING CURVE, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [► 130]).

**i** The function block sends the DALI command SELECT DIMMING CURVE twice, as required by the DALI standard for certain DALI commands.





The function block changes the DTR0 of all connected DALI control gears.

**Inputs**

```

Var_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eDimmingCurve   : E_DALIDimmingCurve := E_DALIDimmingCurve.Standard;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eDimmingCurve	E_DALIDimmingCurve	Linear or standard dimming curve (see <a href="#">E_DALIDimmingCurve ▶ 858</a> )

**Outputs**

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

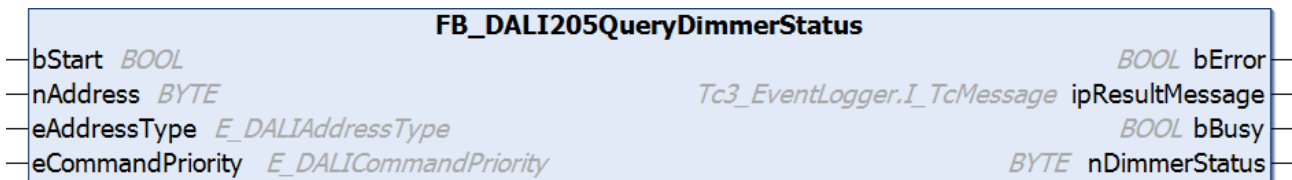
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.4.2 Query commands**

**4.1.2.4.2.1 FB\_DALI205QueryDimmerStatus**



The function block reads the current operation mode from the DALI control gear (see [dimmerStatus](#) [[▶ 303](#)]).

**i** Prior to the DALI command QUERY DIMMER STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [[▶ 130](#)]).

The output *nDimmerStatus* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nDimmerStatus   : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nDimmerStatus	BYTE	Returns the current operation mode (see <a href="#">dimmerStatus [▶ 303]</a> ).

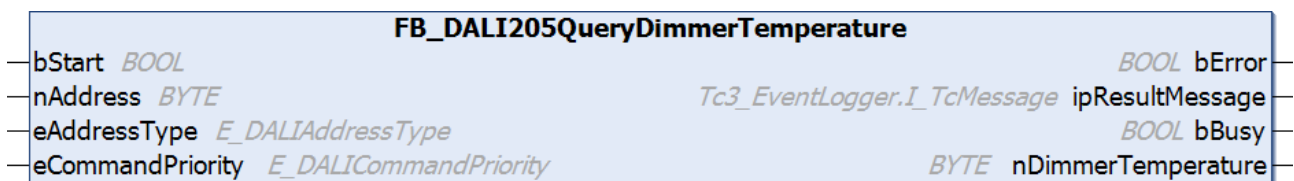
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.4.2.2 FB\_DALI205QueryDimmerTemperature**



The function block reads the current operating temperature from the DALI control gear (see [dimmerTemperature \[▶ 305\]](#)).

*bError* is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY DIMMER TEMPERATURE, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nDimmerTemperature* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nDimmerTemperature : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nDimmerTemperature	BYTE	Returns the current operating temperature (see <a href="#">dimmerTemperature</a> [▶ 305]).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

4.1.2.4.2.3 FB\_DALI205QueryDimmingCurve



The function block reads the set dimming curve from the DALI control gear (see [dimmingCurve](#) [▶ 303]).

**i** Prior to the DALI command QUERY DIMMING CURVE, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *eDimmingCurve* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  eDimmingCurve : E_DALI_DimmingCurve;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eDimmingCurve	BYTE	Returns the currently set dimming curve (see <a href="#">E_DALIDimmingCurve [▶ 858]</a> ).

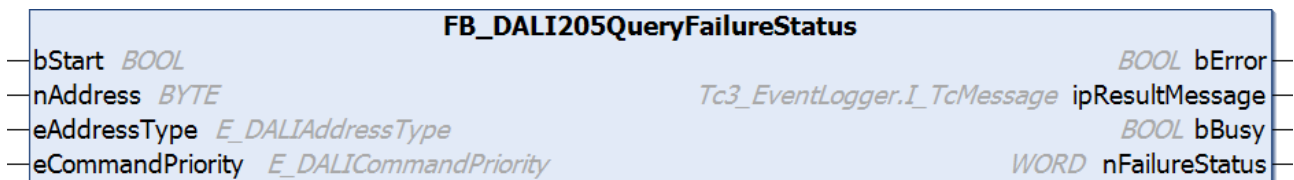
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.4.2.4 FB\_DALI205QueryFailureStatus**



The function block reads the failure status from the DALI control gear (see [failureStatus \[▶ 304\]](#)).

**i** Prior to the DALI command QUERY FAILURE STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nFailureStatus* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nFailureStatus  : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nFailureStatus	WORD	Returns the failure status (see <u>failureStatus</u> [▶ 304]).

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

### 4.1.2.4.2.5 FB\_DALI205QueryFeatures

FB_DALI205QueryFeatures		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	DWORD nFeatures

The function block reads the implemented properties from the DALI control gear (see [dimmerFeatures](#) [▶ 304]).



Prior to the DALI command QUERY FEATURES, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nFeatures* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nFeatures   : DWORD;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nFeatures	DWORD	Returns information about the implemented features (see <a href="#">dimmerFeatures [▶ 304]</a> ).

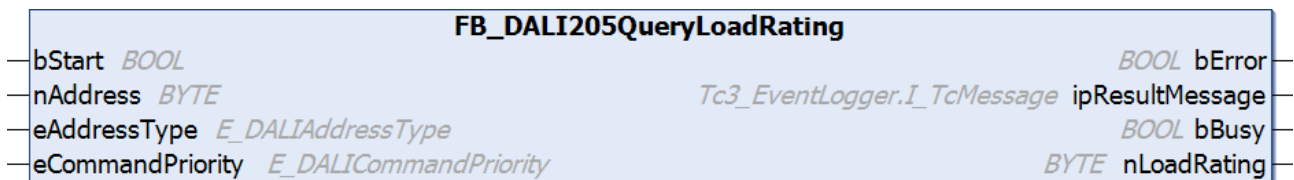
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.4.2.6 FB\_DALI205QueryLoadRating**



The function block reads the maximum permissible root mean square of the load current from the DALI control gear (see [loadRating \[▶ 306\]](#)).

*bError* is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.



Prior to the DALI command QUERY LOAD RATING, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nLoadRating* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nLoadRating     : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLoadRating	BYTE	Returns the maximum permissible root mean square of the load current (see <a href="#">loadRating [▶ 306]</a> ).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

4.1.2.4.2.7 FB\_DALI205QueryRealLoadPower

FB_DALI205QueryRealLoadPower		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	UINT nRealLoadPower

The function block reads the current active power of the load from the DALI control gear (see [realLoadPower](#) [▶ 306]).

*bError* is set to TRUE if the DALI control gear returns the value 65535 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY REAL LOAD POWER, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nRealLoadPower* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nRealLoadPower : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRealLoadPower	UINT	Returns the current active power of the load (see <a href="#">realLoadPower</a> [▶ 306]).

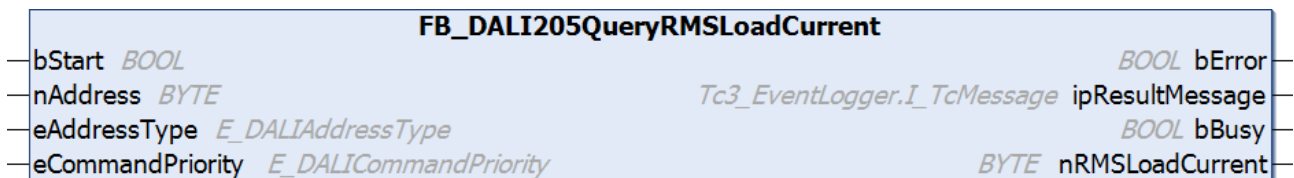
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.4.2.8 FB\_DALI205QueryRMSLoadCurrent**



The function block reads the root mean square of the load current from the DALI control gear ([rmsLoadCurrent](#) [▶ 306]).

*bError* is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY RMS LOAD CURRENT, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nRMSLoadCurrent* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 📡 Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nRMSLoadCurrent : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRMSLoadCurrent	BYTE	Returns the root mean square of the load current (see <a href="#">rmsLoadCurrent [▶ 306]</a> ).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

#### 4.1.2.4.2.9 FB\_DALI205QueryRMSLoadVoltage

FB_DALI205QueryRMSLoadVoltage		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BYTE nRMSLoadVoltage

The function block reads the root mean square of the load voltage from the DALI control gear (see [rmsLoadVoltage](#) [▶ 305]).

*bError* is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY RMS LOAD VOLTAGE, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nRMSLoadVoltage* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nRMSLoadVoltage : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRMSLoadVoltage	BYTE	Returns the root mean square of the load voltage (see <a href="#">rmsLoadVoltage [▶ 305]</a> ).

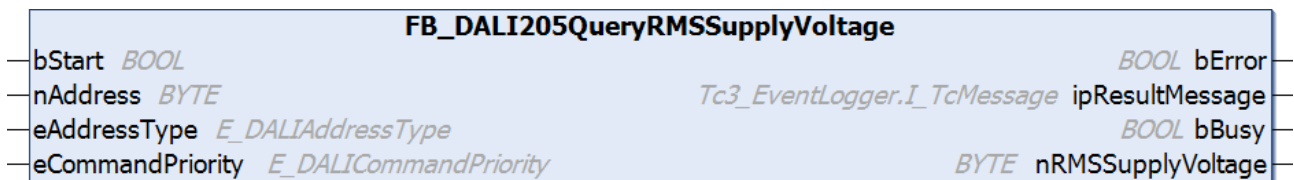
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.1.2.4.2.10 FB\_DALI205QueryRMSSupplyVoltage**



The function block reads out the root mean square of the supply voltage from the DALI control gear (see [rmsSupplyVoltage \[▶ 305\]](#)).

*bError* is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY RMS SUPPLY VOLTAGE, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nRMSSupplyVoltage* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bBusy            : BOOL;
  nRMSSupplyVoltage : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRMSSupplyVoltage	BYTE	Returns the root mean square of the supply voltage (see <a href="#">rmsSupplyVoltage [▶ 305]</a> ).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0



4.1.2.4.2.11 FB\_DALI205QuerySupplyFrequency

FB_DALI205QuerySupplyFrequency		
bStart	BOOL	BOOL bError
nAddress	BYTE	Tc3_EventLogger.I_TcMessage ipResultMessage
eAddressType	E_DALIAddressType	BOOL bBusy
eCommandPriority	E_DALICommandPriority	BYTE nSupplyFrequency

The function block reads the current frequency of the supply voltage from the DALI control gear (see [supplyFrequency](#) [▶ 305]).

*bError* is set to TRUE if the DALI control gear returns the value 255 (MASK). In this case, the value cannot be determined by the DALI control gear.

**i** Prior to the DALI command QUERY SUPPLY FREQUENCY, the function block sends the DALI command ENABLE DEVICE TYPE 4, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nSupplyFrequency* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nSupplyFrequency : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nSupplyFrequency	BYTE	Returns the current frequency of the supply voltage (see <a href="#">supplyFrequency [► 305]</a> ).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

4.1.2.4.3 Variables

Name	Reset value	Scope	Size	Comments
<a href="#">dimmingCurve</a> [▶ 303]	<i>E_DALIDimmingCurve.Standard</i>	<i>E_DALIDimmingCurve.Standard</i> , <i>E_DALIDimmingCurve.Linear</i>	1 byte	
<a href="#">dimmerStatus</a> [▶ 303]	no change, except bit 4 is set to FALSE	16#00...16#FF	1 byte	read only
<a href="#">dimmerFeatures</a> [▶ 304]	No change	16#00_00_00... 16#FF_FF_FF	3 bytes	read only
<a href="#">failureStatus</a> [▶ 304]	No change	16#00_00... 16#FF_FF	2 byte	read only
<a href="#">dimmerTemperature</a> [▶ 305]	No change	0...255	1 byte	read only
<a href="#">rmsSupplyVoltage</a> [▶ 305]	No change	0...255	1 byte	read only
<a href="#">supplyFrequency</a> [▶ 305]	No change	0...255	1 byte	read only
<a href="#">rmsLoadVoltage</a> [▶ 305]	No change	0...255	1 byte	read only
<a href="#">rmsLoadCurrent</a> [▶ 306]	No change	0...255	1 byte	read only
<a href="#">realLoadPower</a> [▶ 306]	No change	0...6535	2 byte	read only
<a href="#">loadRating</a> [▶ 306]	No change	0...255	1 byte	read only

**dimmingCurve**

The variable specifies the type of dimming curve.

The variable can be written with the function block [FB\\_DALI205SelectDimmingCurve](#) [▶ 284] and read with the function block [FB\\_DALI205QueryDimmingCurve](#) [▶ 289].

E_DALIDimmingCurve	Description
Standard	Standard dimming curve
Linear	Linear dimming curve

**dimmerStatus**

The variable contains information about the current operation mode of the DALI control gear.

The variable can be read with the function block [FB\\_DALI205QueryDimmerStatus](#) [▶ 286].

Bit	Description
0	Leading edge phase control active
1	Trailing edge phase control active
2	Reserve
3	Reserve
4	Linear dimming curve active
5	Reserve
6	Reserve
7	Reserve

**dimmerFeatures**

The variable contains information about the implemented properties of the DALI control gear.

The variable can be read with the function block `FB_DALI205QueryFeatures` [► 292].

Bit	Description
0	Load overcurrent shutdown can be queried
1	Idle detection (no load) can be queried
2	Reserve
3	Reserve
4	Reserve
5	Thermal shutdown can be queried
6	Reduction of the output level due to thermal overload can be queried
7	Reserve
8	Operating temperature can be queried
9	Supply voltage can be queried
10	Frequency of the supply voltage can be queried
11	Load voltage can be queried
12	Load current can be queried
13	Active power of the load can be queried
14	Maximum permissible load current can be queried
15	Reduction of the output level due to load overcurrent can be queried
16	Supported dimming methods (see table below)
17	Supported dimming methods (see table below)
18	Reserve
19	Linear dimming curve can be selected
20	Reserve
21	Reserve
22	Reserve
23	A load that is not compatible for the DALI control gear can be queried

Bit 16 and bit 17 define the supported dimming methods:

Bit 17	Bit 16	Meaning
0	0	Leading edge phase control and trailing edge phase control
0	1	Leading edge phase control
1	0	Trailing edge phase control
1	1	Sine curve

**failureStatus**

The variable contains information about the failure status.

The variable can be read with the function block `FB_DALI205QueryFailureStatus` [► 290].

Bit	Description
0	Shutdown due to load overcurrent
1	Idle (no load) detected
2	Reserve
3	Reserve
4	Reserve
5	Thermal shutdown
6	Thermal overload with output level reduction
7	Reserve
8	Shutdown because the load is not suitable for the selected dimming method.
9	Limit violation supply voltage
10	Limit violation frequency of the supply voltage
11	Limit violation load voltage
12	Reduction of the output level due to load overcurrent
13	Reserve
14	Reserve
15	Reserve

### dimmerTemperature

The variable contains the operating temperature.

The value is given in steps of 1 °C. The permissible value range is 0 (-40 °C) to 254 (214 °C). If the value 255 (MASK) is returned, the operating temperature is unknown.

The variable can be read with the function block [FB\\_DALI205QueryDimmerTemperature](#) [► 287].

Bit 8 of [dimmerFeatures](#) [► 304] can be used to query whether reading of the operating temperature is supported.

### rmsSupplyVoltage

The variable contains the root mean square of the supply voltage.

The value is given in steps of 2 V. The permissible value range is 0 (0 V) to 254 (508 V). If the value 255 (MASK) is returned, the supply voltage is unknown.

The variable can be read with the function block [FB\\_DALI205QueryRMSSupplyVoltage](#) [► 299].

Bit 9 of [dimmerFeatures](#) [► 304] can be used to query whether reading of the supply voltage is supported.

### supplyFrequency

The variable contains the current frequency of the supply voltage.

The value is given in steps of 0.5 Hz. The permissible value range is 0 (0 Hz) to 254 (127 Hz). If the value 255 (MASK) is returned, the frequency is unknown.

The variable can be read with the function block [FB\\_DALI205QuerySupplyFrequency](#) [► 301].

Bit 10 of [dimmerFeatures](#) [► 304] can be used to query whether reading of the frequency is supported.

### rmsLoadVoltage

The variable contains the root mean square of the load voltage.

The value is given in steps of 2 V. The permissible value range is 0 (0 V) to 254 (508 V). If the value 255 (MASK) is returned, the load voltage is unknown.

The variable can be read with the function block [FB\\_DALI205QueryRMSLoadVoltage](#) [► 298].

Bit 11 of [dimmerFeatures](#) [► 304] can be used to query whether reading of the load voltage is supported.

### **rmsLoadCurrent**

The variable contains the root mean square of the load current in relation to the maximum load current (see [loadRating](#) [► 306]).

The value is given in steps of 0.5 %. The permissible value range is 0 (0 %) to 254 (127 %). If the value 255 (MASK) is returned, the load current is unknown.

The variable can be read with the function block [FB\\_DALI205QueryRMSLoadCurrent](#) [► 296].

Bit 12 of [dimmerFeatures](#) [► 304] can be used to query whether reading of the load current is supported.

### **realLoadPower**

The variable contains the active power of the load.

The value is given in steps of 0.25 W. The permissible value range is 0 (0 W) to 65534 (16383.5 W). If the value 65535 (MASK) is returned, the active power is unknown.

The variable can be read with the function block [FB\\_DALI205QueryRealLoadPower](#) [► 295].

Bit 13 of [dimmerFeatures](#) [► 304] can be used to query whether reading of the active power is supported.

### **loadRating**

The variable contains the maximum permissible root mean square of the load current.

The value is given in steps of 150 mA. The permissible value range is 0 (0 A) to 254 (38.1 A). If the value 255 (MASK) is returned, the maximum permissible load current is unknown.

The variable can be read with the function block [FB\\_DALI205QueryRealLoadRating](#) [► 293].

Bit 14 of [dimmerFeatures](#) [► 304] can be used to query whether reading of the maximum permissible load current is supported.

## **4.1.2.5 Part 207 (LED modules)**

### **4.1.2.5.1 Introduction**

IEC 62386-207 describes DALI control gears for controlling LED modules.

This part includes functions that go beyond IEC 62386-102 for controlling LED modules.

#### **Functionality**

The Tc3\_DALI library offers function blocks for setting various parameters:

- Referencing system performance to detect a significant increase or drop.
- Dimming curve (standard or linear)
- Fast Fade Time (see [fastFadeTime](#) [► 330])

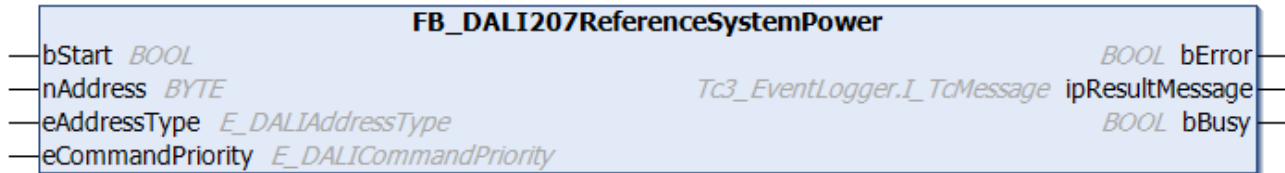
Furthermore, the Tc3\_DALI library contains function blocks for querying various parameters:

- Device type (see [ControlGearType](#) [► 330])
- Dimming curve (see [dimmingCurve](#) [► 331])
- Querying the failure status (see [failureStatus](#) [► 330])
- Fast Fade Time (see [fastFadeTime](#) [► 330] or [minFastFadeTime](#) [► 329])

- Device-specific features (see [Features \[▶ 330\]](#))
- Checking the load deviation
- Querying the referencing

### 4.1.2.5.2 Configuration commands

#### 4.1.2.5.2.1 FB\_DALI207ReferenceSystemPower



The DALI control gear measures and stores the performance level of the system, in order to detect load increase and decrease.

**i** Prior to the DALI command REFERENCE SYSTEM POWER, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

**i** The function block sends the DALI command REFERENCE SYSTEM POWER twice, as foreseen by the DALI standard for certain DALI commands.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType [▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority [▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

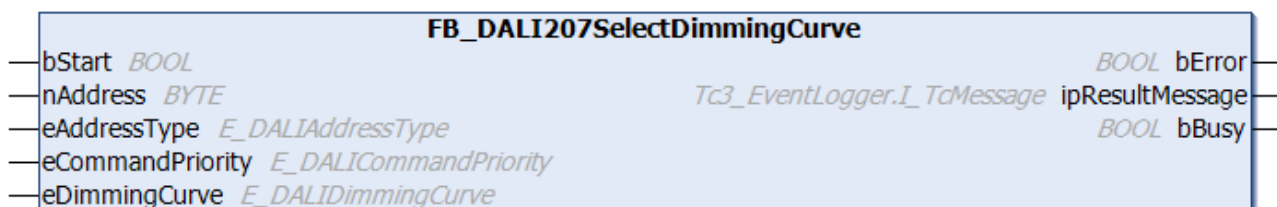
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.5.2.2 FB\_DALI207SelectDimmingCurve**



The function block saves the value *eDimmingCurve* in the variable [dimmingCurve](#) [▶ 331] of the DALI control gear.

**i** Prior to the DALI command SELECT DIMMING CURVE, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

**i** The function block sends the DALI command SELECT DIMMING CURVE twice, as required by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
```



```
eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
eDimmingCurve   : E_DALIDimmingCurve := E_DALIDimmingCurve.Standard;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eDimmingCurve	<a href="#">E_DALIDimmingCurve</a> [▶ 858]	Standard or linear dimming curve

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

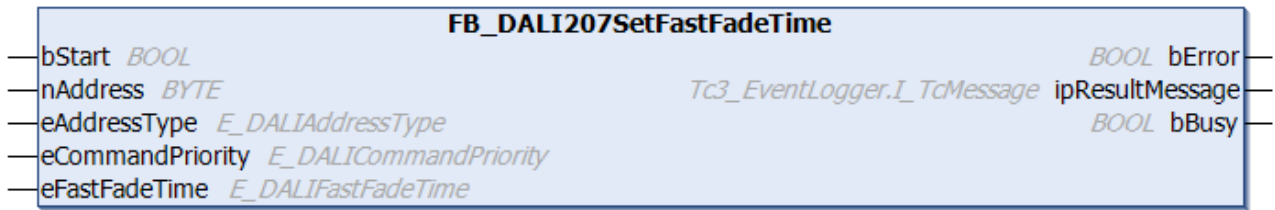
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	<a href="#">I_DALICommunication</a>	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.5.2.3 FB\_DALI207SetFastFadeTime



The function block saves the value *eFastFadeTime* in the variable *fastFadeTime* [▶ 330] of the DALI control gear.

The *fastFadeTime* is used in the function blocks *FB\_DALI102DirectArcPowerControl* [▶ 82] and *FB\_DALI102GoToScene* [▶ 87].

**i** Prior to the DALI command SET FAST FADE TIME, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also *FB\_DALI102EnableDeviceType* [▶ 130]).

**i** The function block sends the DALI command SET FAST FADE TIME twice, as required by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eFastFadeTime : E_DALIFastFadeTime := E_DALIFastFadeTime.Disabled;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eFastFadeTime	E_DALIFastFadeTime [▶ 861]	Fast fade time for changing the output value.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

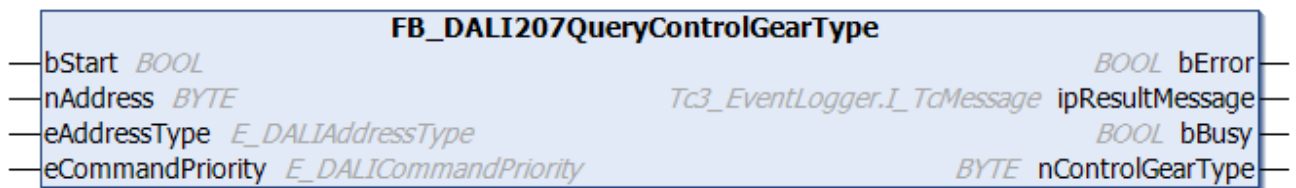
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.5.3 Query commands**

**4.1.2.5.3.1 FB\_DALI207QueryControlGearType**



The function block reads the variable `controlGearType` [[▶ 330](#)] from the DALI control gear.

**i** Prior to the DALI command QUERY CONTROL GEAR TYPE, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output `nControlGearType` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nControlGearType : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearType	BYTE	Outputs the value of the variable <i>controlGearType</i> (0...255).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.5.3.2 FB\_DALI207QueryDimmingCurve



The function block reads the set dimming curve from the DALI control gear (see [dimmingCurve](#) [▶ 331]).

**i** Prior to the DALI command QUERY DIMMING CURVE, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *eDimmingCurve* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  eDimmingCurve : E_DALIDimmingCurve;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eDimmingCurve	E_DALIDimmingCurve [▶ 858]	Returns the currently set dimming curve.

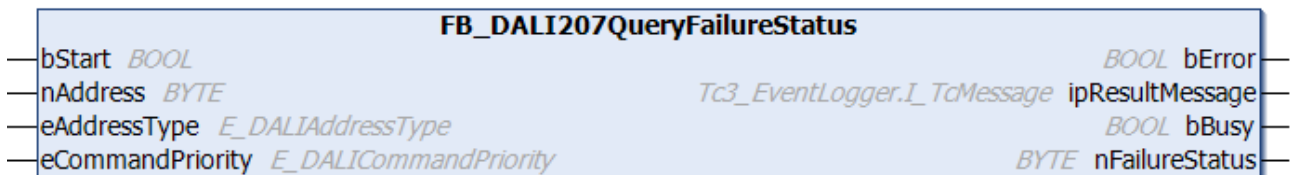
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.5.3.3 FB\_DALI207QueryFailureStatus**



The function block reads the failure status from the DALI control gear (see [failureStatus \[▶ 330\]](#)).

**i** Prior to the DALI command QUERY FAILURE STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nFailureStatus* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nFailureStatus  : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nFailureStatus	BYTE	Returns the failure status (see <u>failureStatus</u> [▶ 330]).

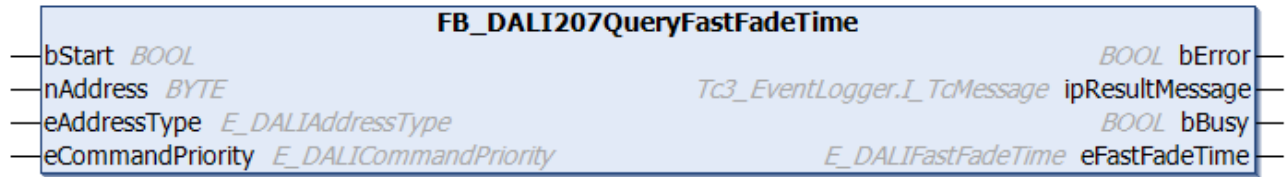
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.5.3.4 FB\_DALI207QueryFastFadeTime



The function block reads the variable `fastFadeTime` [▶ 330] from the DALI control gear.

The `fastFadeTime` is used in the function blocks `FB_DALI102DirectArcPowerControl` [▶ 82] and `FB_DALI102GoToScene` [▶ 87].

**i** Prior to the DALI command QUERY FAST FADE TIME, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [▶ 130]).

The output `eFastFadeTime` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

#### 🔧 Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
<code>bStart</code>	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
<code>nAddress</code>	BYTE	Address of a DALI control gear or a DALI group.
<code>eAddressType</code>	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
<code>eCommandPriority</code>	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### 🔧 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  eFastFadeTime   : E_DALIFastFadeTime;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eFastFadeTime	E_DALIFastFadeTime [▶ 861]	Output value of <i>fastFadeTime</i> .

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.5.3.5 FB\_DALI207QueryFeatures**



The function block reads the variable [features \[▶ 330\]](#) from the DALI control gear.



Prior to the DALI command QUERY FEATURES, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nFeatures* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nFeatures       : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nFeatures	BYTE	Returns information about the implemented features (see <a href="#">features [▶ 330]</a> ).



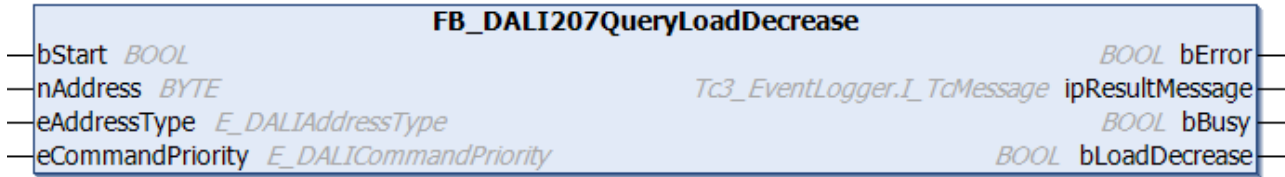
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.5.3.6 FB\_DALI207QueryLoadDecrease



The function block queries whether a significant load decrease (in comparison with the reference power of the system) has been detected.

**i** Prior to the DALI command QUERY LOAD DECREASE, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

The output *bLoadDecrease* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nLoadDecrease : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearType	BOOL	Detection of a significant load decrease.

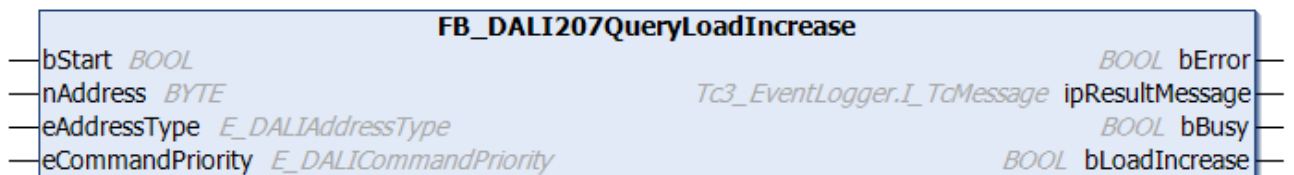
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.5.3.7 FB\_DALI207QueryLoadIncrease**



The function block queries whether a significant load increase (in comparison with the reference power of the system) has been detected.

**i** Prior to the DALI command QUERY LOAD INCREASE, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *bLoadIncrease* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bLoadIncrease   : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bLoadIncrease	BOOL	Detection of a significant load increase.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.5.3.8 FB\_DALI207QueryMinFastFadeTime



The function block reads the variable `minFastFadeTime` [▶ 329] from the DALI control gear.

**i** Prior to the DALI command MIN FAST FADE TIME, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

The output `eMinFastFadeTime` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  eMinFastFadeTime : E_DALIFastFadeTime;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
eMinFastFadeTime	E_DALIFastFadeTime [▶ 861]).	Outputs the minimum value of <i>fastFadeTime</i> .

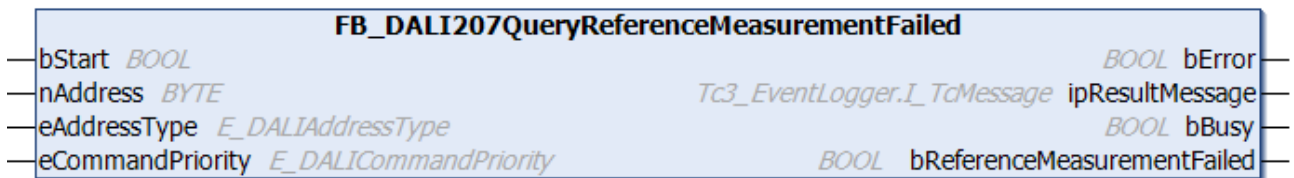
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.5.3.9 FB\_DALI207QueryReferenceMeasurementFailed**



The function block queries whether a started reference measurement has failed.

**i** Prior to the DALI command QUERY REFERENCE MEASUREMENT FAILED, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *bReferenceMeasurementFailed* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bReferenceMeasurementFailed : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bReferenceMeasurementFailed	BOOL	Indicates whether the reference measurement failed.



### Properties

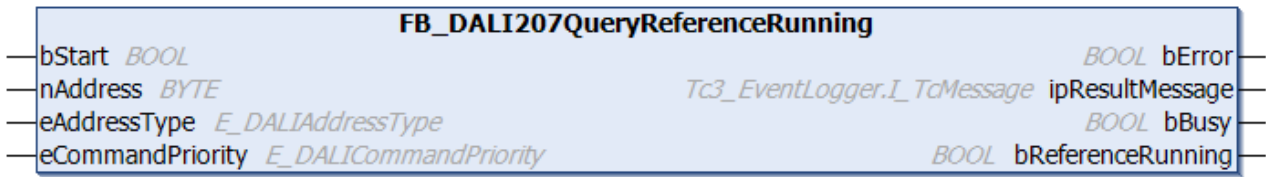
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0



4.1.2.5.3.10 FB\_DALI207QueryReferenceRunning



The function block queries whether a reference measurement of the system power is activated.

**i** Prior to the DALI command QUERY REFERENCE RUNNING, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

The output `bReferenceRunning` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  bReferenceRunning : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bReferenceRunning	BOOL	Indicates whether the reference measurement of the system performance is performed.

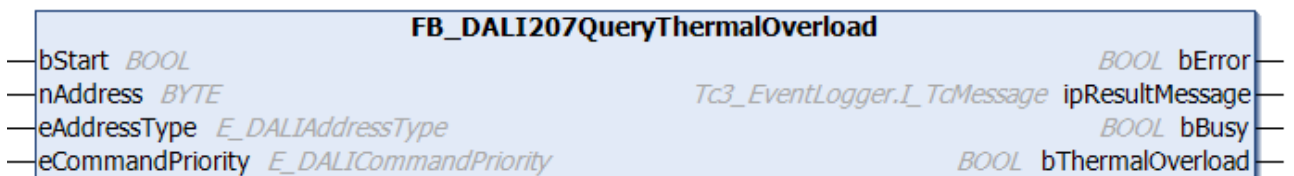
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.5.3.11 FB\_DALI207QueryThermalOverload**



The function block queries whether there is a thermal overload with reduction of the output level.

**i** Prior to the DALI command QUERY THERMAL OVERLOAD, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *bThermalOverload* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 🔌 Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bThermalOverload : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bThermalOverload	BOOL	Indicates whether thermal overload is present.

### 📄 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.5.3.12 FB\_DALI207QueryThermalShutdown



The function block queries whether a thermal shutdown has taken place.

**i** Prior to the DALI command QUERY THERMAL SHUTDOWN, the function block sends the DALI command ENABLE DEVICE TYPE 6, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

The output *bThermalShutdown* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  bThermalShutdown : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bThermalShutdown	BOOL	Indicates whether a thermal shutdown has taken place.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.5.4 Variables**

Name	Reset value	Scope	Size	Comment
<a href="#">minFastFadeTime [► 329]</a>	No change	<i>E_DALIFastFadeTime.T100ms</i> ... <i>E_DALIFastFadeTime.T700ms</i>	1 byte	Read only
<a href="#">fastFadeTime [► 330]</a>	<i>E_DALIFastFadeTime.Disabled</i>	<i>E_DALIFastFadeTime.Disabled, minFastFadeTime</i> ... <i>E_DALIFastFadeTime.T700ms</i>	1 byte	
<a href="#">controlGearType [► 330]</a>	No change	0...255	1 byte	Read only
<a href="#">features [► 330]</a>	No change	0...255	1 byte	Read only
<a href="#">failureStatus [► 330]</a>	No change	0...255	1 byte	Read only
<a href="#">dimmingCurve [► 331]</a>	<i>E_DALIDimmingCurve.Standard</i>	<i>E_DALIDimmingCurve.Standard, E_DALIDimmingCurve.Linear</i>	1 byte	

**minFastFadeTime**

The *minFastFadeTime* specifies the shortest *fadeTime* within which the output value of a DALI control gear behaves according to the selected dimming curve. Its value can lie between *E\_DALIFastFadeTime.T100ms* and *E\_DALIFastFadeTime.T700ms*.

The value of the variable can be read out with the function block [FB\\_DALI207QueryMinFastFadeTime \[► 322\]](#).

**fastFadeTime**

The *fastfadeTime* is used instead of the *fadeTime* if the *fadeTime* is equal to *E\_DALIFastFadeTime.Disabled*. The *fastFadeTime* can be programmed to *E\_DALIFastFadeTime.Disabled* or any value in the range between the value in the variable *minFastFadeTime* and *E\_DALIFastFadeTime.T700ms*. Programming the *fastFadeTime* to *E\_DALIFastFadeTime.Disabled* means the fastest possible change of the output value.

The variable has an effect on the DALI commands [FB\\_DALI102DirectArcPowerControl](#) [► 82], [FB\\_DALI102GoToScene](#) [► 87] and [FB\\_DALI102GoToLastActiveLevel](#) [► 85].

The value of the variable can be read with the function block [FB\\_DALI207QueryFastFadeTime](#) [► 316] and changed with the function block [FB\\_DALI207SetFastFadeTime](#) [► 310].

**controlGearType**

The variable contains information about the device type.

The value of the variable can be read out with the function block [FB\\_DALI207QueryControlGearType](#) [► 311].

Bit	Description
0	Reserve
1	LED module integrated
2	Reserve
3	Reserve
4	Reserve
5	Reserve
6	Reserve
7	Reserve

**features**

The variable contains information about the implemented optional properties.

The value of the variable can be read out with the function block [FB\\_DALI207QueryFeatures](#) [► 317].

Bit	Description
0	Reserve
1	Reserve
2	Detection of the load decrease can be queried
3	Detection of the load increase can be queried
4	Reserve
5	Thermal shutdown can be queried
6	Reduction of the output level due to thermal overload can be queried
7	Reserve

**failureStatus**

The variable contains information about the failure status.

The value of the variable can be read out with the function block [FB\\_DALI207QueryFailureStatus](#) [► 314].

Bit	Description
0	Reserve
1	Reserve
2	Load decrease
3	Load increase
4	Reserve
5	Thermal shutdown
6	Thermal overload with output level reduction
7	Reference measurement failed

### dimmingCurve

The variable specifies the type of dimming curve.

The value of the variable can be read out with the function block [FB\\_DALI207QueryDimmingCurve \[► 313\]](#) and changed with the function block [FB\\_DALI207SelectDimmingCurve \[► 308\]](#).

E_DALIDim- mingCurve	Description
Standard	Standard dimming curve
Linear	Linear dimming curve

## 4.1.2.6 Part 208 (device for switching functions)

### 4.1.2.6.1 Introduction

IEC 62386-208 describes DALI control gears with switching function.

DALI control gears with switching function have a digital output (e.g. a relay). The digital output is switched via the DALI commands for changing the output value (DAPC, OFF, RECALL MIN LEVEL, RECALL MAX LEVEL, UP, DOWN, STEP UP, STEP DOWN, ...). These DALI commands do not directly change the [actualLevel \[► 150\]](#) variable, but a virtual output value.

Four different threshold values are used to determine at which value of the virtual output value the digital output is set or reset.

If the digital output is active (TRUE), *actualLevel* = 254. If this is inactive (FALSE), *actualLevel* = 0. The value of the virtual output value cannot be queried.

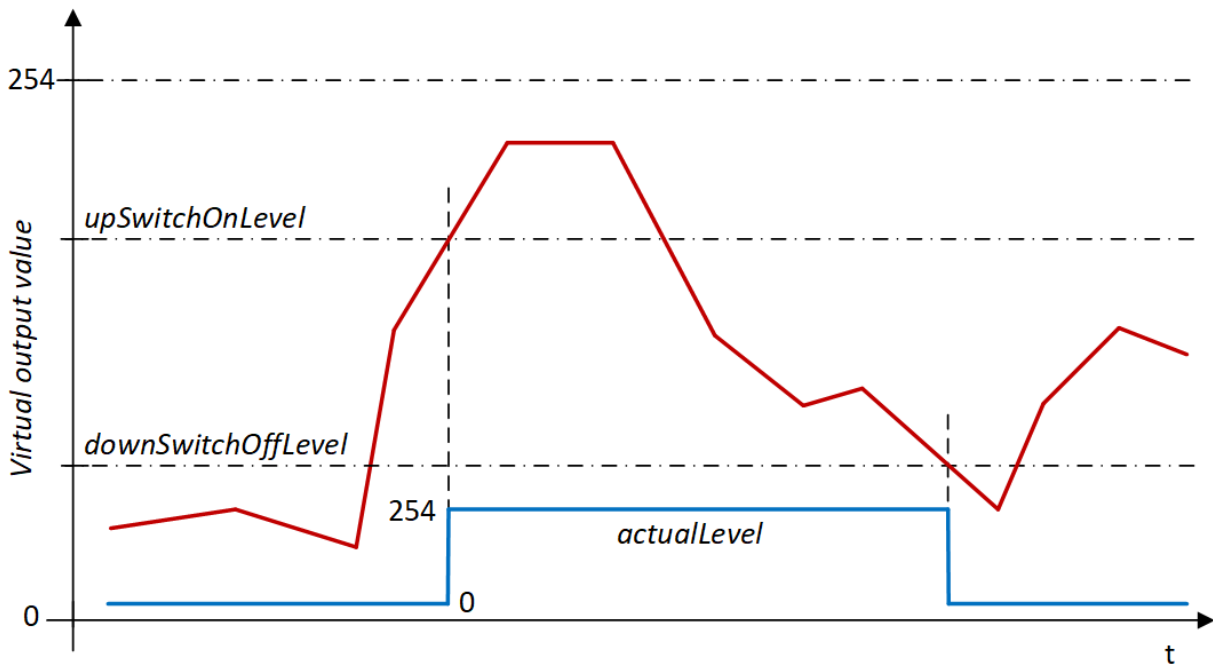
The default values of the [variables \[► 350\]](#) are set so that setting the virtual output value to 0 resets the digital output and a value of 1...254 sets the digital output. The value 255 (MASK) has no effect on the digital output.

#### Threshold values

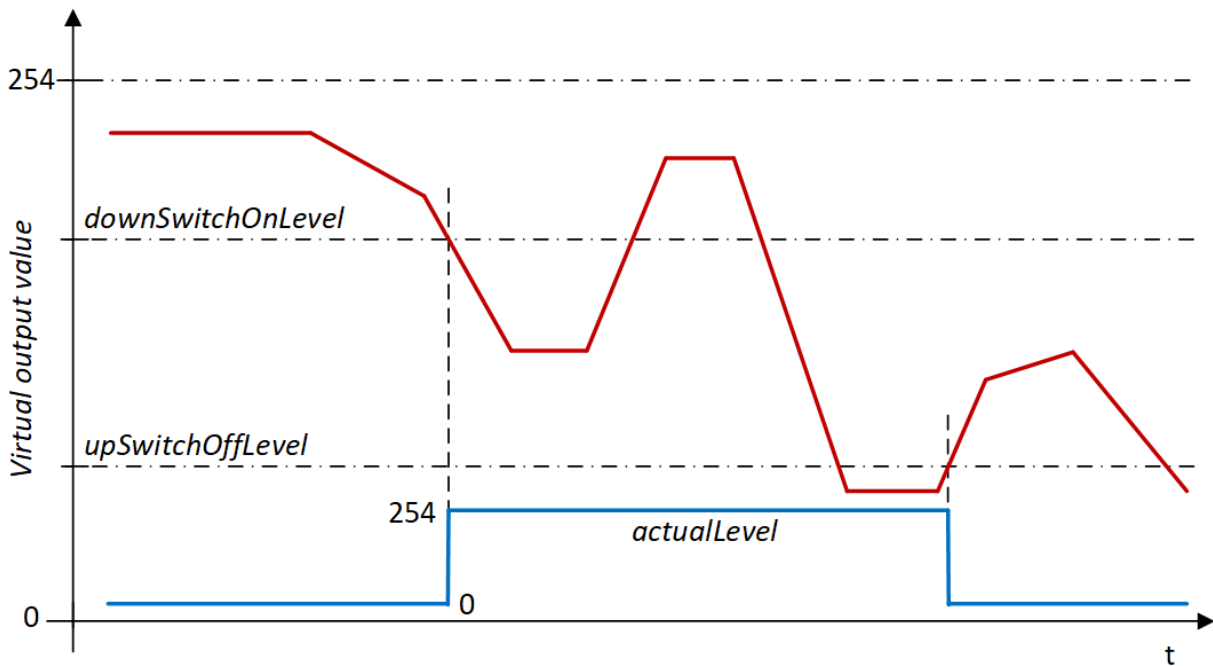
A total of four threshold values are available. Two threshold values ([upSwitchOnLevel \[► 350\]](#) and [upSwitchOffLevel \[► 350\]](#)) define the switching points at an increasing virtual output value. While the other two threshold values ([downSwitchOnLevel \[► 350\]](#) and [downSwitchOffLevel \[► 350\]](#)) determine when the output switches on a falling virtual output value. If a threshold value is not to have a function, it is set to 255 (MASK).

In most applications, it is not practical to use all four threshold values.

The following example shows how the digital output value behaves when the threshold values *downSwitchOffLevel* and *upSwitchOnLevel* are used. The two threshold values *downSwitchOnLevel* and *upSwitchOffLevel* have the value 255 (MASK) and are therefore without function.



The control direction can also be reversed by using the threshold values *downSwitchOnLevel* and *upSwitchOffLevel*. In this case, *downSwitchOffLevel* and *upSwitchOnLevel* are disabled.



Note that the default value of *minLevel* according to IEC 62386-208 is 254, which is the same as *maxLevel*. This means that the virtual output value can only assume the values 0 and 254. Depending on the threshold value, it may also be necessary to adjust *minLevel*.



### 4.1.2.6.2 Configuration commands

#### 4.1.2.6.2.1 FB\_DALI208SetDownSwitchOffThreshold



The function block stores the value *nDownSwitchOffThreshold* in the variable *downSwitchOffThreshold* [▶ 350] of the DALI control gear.

The output from the DALI control gear is reset as soon as the falling virtual output value reaches or falls below the threshold value *downSwitchOffThreshold*.

With the value *nDownSwitchOffThreshold* = 255 (MASK) the threshold is disabled.

The DALI command is only supported by the DALI control gear if bit 3 of *features* [▶ 350] is set.

- i** Prior to the DALI command SET DOWN SWITCH-OFF THRESHOLD, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

---

- i** The function block sends the DALI command SET DOWN SWITCH-OFF THRESHOLD twice, as foreseen by the DALI standard for certain DALI commands.

---

- i** The function block changes the DTR0 of all connected DALI control gears.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nDownSwitchOffThreshold : BYTE := 0;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

- i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDownSwitchOffThreshold	BYTE	Threshold value (0...255) for switching off the output with a falling virtual output value.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

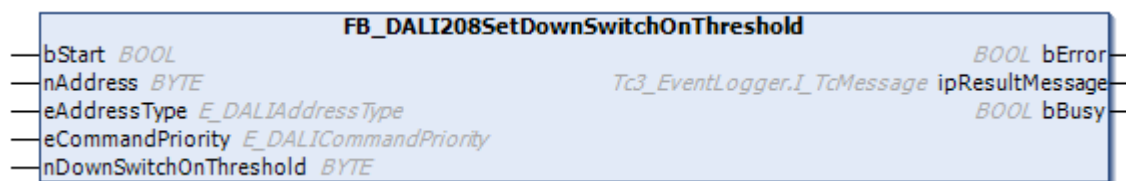
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.2.6.2.2 FB\_DALI208SetDownSwitchOnThreshold**



The function block stores the value *nDownSwitchOnThreshold* in the variable [downSwitchOnThreshold](#) [▶ 350] of the DALI control gear.

The output from the DALI control gear is set as soon as the falling virtual output value reaches or falls below the threshold value *downSwitchOnThreshold*.

With the value *nDownSwitchOnThreshold* = 255 (MASK) the threshold is disabled.

The DALI command is only supported by the DALI control gear if bit 3 of [features](#) [▶ 350] is set.

**i** Prior to the DALI command SET DOWN SWITCH-ON THRESHOLD, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also FB\_DALI102EnableDeviceType [▶ 130]).

**i** The function block sends the DALI command SET DOWN SWITCH-ON THRESHOLD twice, as foreseen by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nDownSwitchOnThreshold : BYTE := 255;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDownSwitchOnThreshold	BYTE	Threshold value (0...255) for switching on the output with a falling virtual output value.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

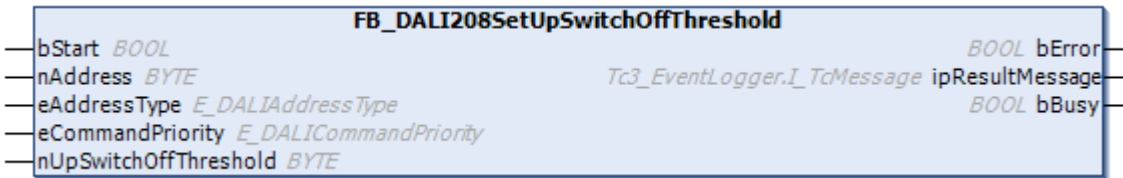
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.2.6.2.3 FB\_DALI208SetUpSwitchOffThreshold**



The function block stores the value *nUpSwitchOffThreshold* in the variable [upSwitchOffThreshold \[▶ 350\]](#) of the DALI control gear.

The output from the DALI control gear is reset as soon as the rising virtual output value reaches or exceeds the threshold value *upSwitchOffThreshold*.

With the value *nUpSwitchOffThreshold* = 255 (MASK) the threshold is disabled.

The DALI command is only supported by the DALI control gear if bit 3 of [features \[▶ 350\]](#) is set.

**i** Prior to the DALI command SET UP SWITCH-OFF THRESHOLD, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

**i** The function block sends the DALI command SET UP SWITCH-OFF THRESHOLD twice, as foreseen by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nUpSwitchOffThreshold : BYTE := 255;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nUpSwitchOffThreshold	BYTE	Threshold value (0...255) for switching off the output with a rising virtual output value.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

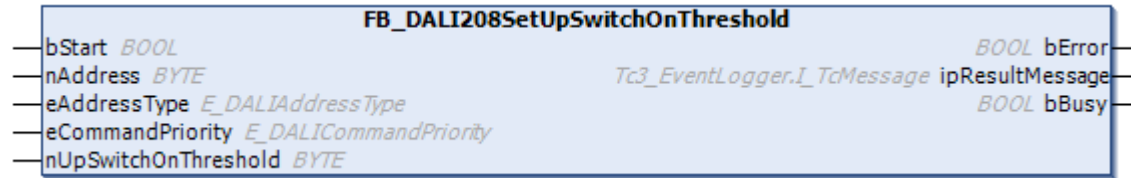
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.2.6.2.4 FB\_DALI208SetUpSwitchOnThreshold**



The function block stores the value *nUpSwitchOnThreshold* in the variable *upSwitchOnThreshold* [▶ 350] of the DALI control gear.

The output from the DALI control gear is set as soon as the rising virtual output value reaches or exceeds the threshold value *upSwitchOnThreshold*.

With the value *nUpSwitchOnThreshold* = 255 (MASK) the threshold is disabled.

The DALI command is only supported by the DALI control gear if bit 3 of *features* [▶ 350] is set.

**i** Prior to the DALI command SET UP SWITCH-ON THRESHOLD, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also *FB\_DALI102EnableDeviceType* [▶ 130]).

**i** The function block sends the DALI command SET UP SWITCH-ON THRESHOLD twice, as foreseen by the DALI standard for certain DALI commands.

**i** The function block changes the DTR0 of all connected DALI control gears.

**Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nUpSwitchOnThreshold : BYTE := 255;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nUpSwitchOnThreshold	BYTE	Threshold value (0...255) for switching on the output with a rising virtual output value.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

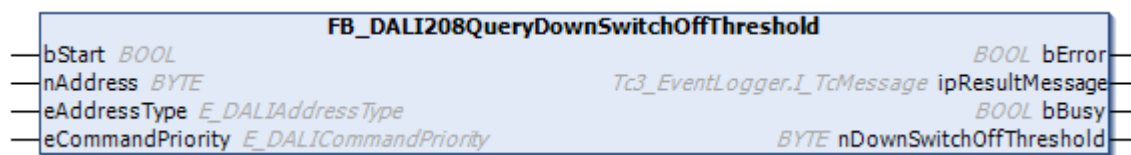
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.2.6.3 Query commands**

**4.1.2.6.3.1 FB\_DALI208QueryDownSwitchOffThreshold**



The function block reads the variable [downSwitchOffThreshold](#) [▶ 350] from the DALI control gear.

The output from the DALI control gear is reset as soon as the falling virtual output value reaches or falls below the threshold value *downSwitchOffThreshold*.



Prior to the DALI command QUERY DOWN SWITCH-OFF, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nDownSwitchOffThreshold* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nDownSwitchOffThreshold : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nDownSwitchOffThreshold	BYTE	Threshold value (0...255) for switching off the output with a falling virtual output value.



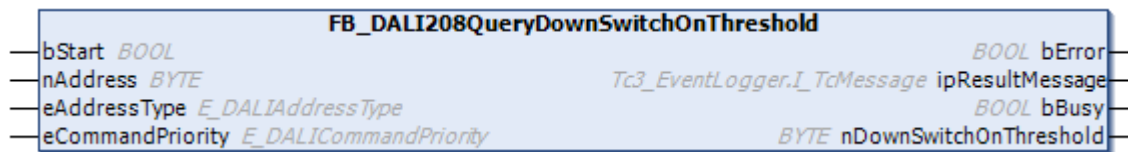
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

4.1.2.6.3.2 FB\_DALI208QueryDownSwitchOnThreshold



The function block reads the variable `downSwitchOnThreshold` [► 350] from the DALI control gear.

The output from the DALI control gear is set as soon as the falling virtual output value reaches or falls below the threshold value `downSwitchOnThreshold`.

**i** Prior to the DALI command QUERY DOWN SWITCH-ON, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[► 130\]](#)).

The output `nDownSwitchOnThreshold` only contains a valid value if the function block was executed without errors (`bError = FALSE`).

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nDownSwitchOnThreshold : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nDownSwitchOnThreshold	BYTE	Threshold value (0...255) for switching on the output with a falling virtual output value.

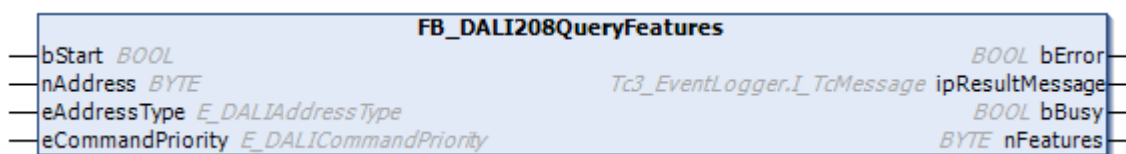
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.2.6.3.3 FB\_DALI208QueryFeatures**



The function block reads the variable [features](#) [▶ 350] from the DALI control gear.

**i** Prior to the DALI command QUERY FEATURES, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nFeatures* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart : BOOL;
  nAddress : BYTE;
```

```
eAddressType      : E_DALIAddressType := E_DALIAddressType.Short;
eCommandPriority  : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nFeatures       : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nFeatures	BYTE	Contains information about the implemented features (see <a href="#">features [▶ 350]</a> ).

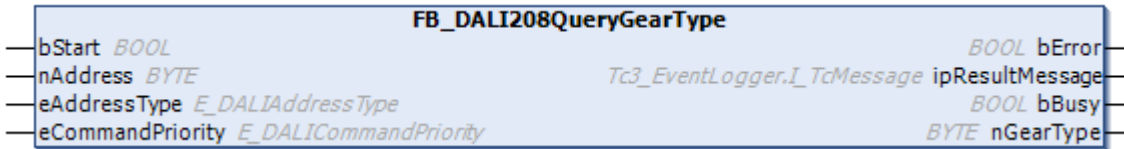
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

### 4.1.2.6.3.4 FB\_DALI208QueryGearType



The function block reads the variable `gearType` [▶ 351] from the DALI control gear.

**i** Prior to the DALI command QUERY GEAR TYPES, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [▶ 130]).

The output `nGearType` only contains a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nGearType   : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nGearType	BYTE	Contains information about the properties of the DALI control gear (see <a href="#">gearType</a> [▶ 351]).

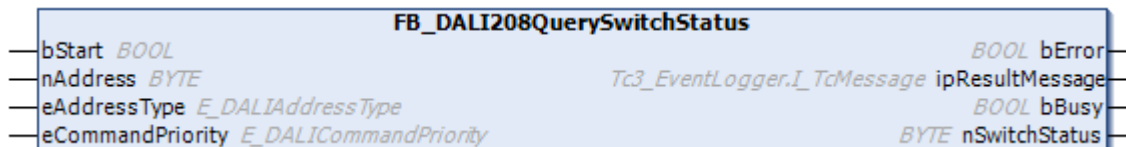
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.2.6.3.5 FB\_DALI208QuerySwitchStatus**



The function block reads the variable [switchStatus](#) [▶ 351] from the DALI control gear.

**i** Prior to the DALI command QUERY SWITCH STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nSwitchStatus* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bBusy            : BOOL;
  nSwitchStatus    : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nSwitchStatus	BYTE	Returns additional information on the current status of the DALI control gear (see <u>switchStatus</u> [▶ 351]).



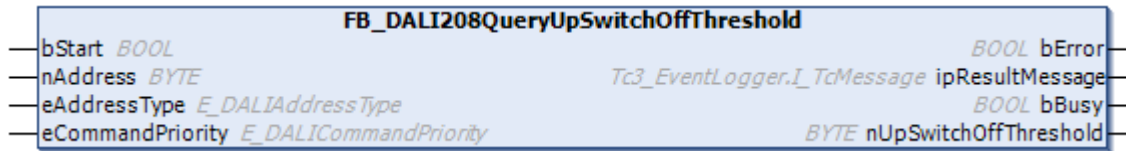
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

### 4.1.2.6.3.6 FB\_DALI208QueryUpSwitchOffThreshold



The function block reads the variable `upSwitchOffThreshold` [► 350] from the DALI control gear.

The output from the DALI control gear is reset as soon as the rising virtual output value reaches or exceeds the threshold value `upSwitchOffThreshold`.

The threshold value is disabled if a value of 255 (MASK) is returned.

**i** Prior to the DALI command QUERY UP SWITCH-OFF, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [► 130]).

The output `nUpSwitchOffThreshold` only contains a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nUpSwitchOffThreshold : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nUpSwitchOffThreshold	BYTE	Threshold value (1...255) for switching off with a rising virtual output value.

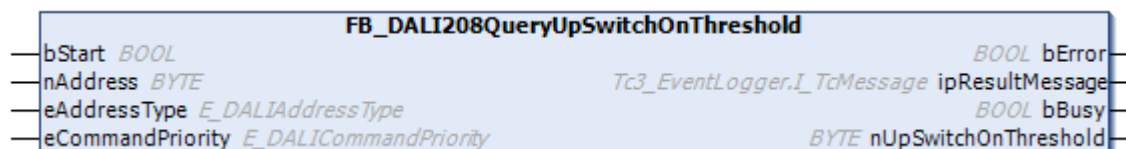
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.2.6.3.7 FB\_DALI208QueryUpSwitchOnThreshold**



The function block reads the variable [upSwitchOnThreshold](#) [▶ 350] from the DALI control gear.

The output from the DALI control gear is set as soon as the rising virtual output value reaches or exceeds the threshold value *upSwitchOnThreshold*.

The threshold value is disabled if a value of 255 (MASK) is returned.

**i** Prior to the DALI command QUERY UP SWITCH-ON, the function block sends the DALI command ENABLE DEVICE TYPE 7, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nUpSwitchOnThreshold* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 🔌 Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nUpSwitchOnThreshold : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nUpSwitchOnThreshold	BYTE	Threshold value (1...255) for switching on with a rising virtual output value.



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

#### 4.1.2.6.4 Variables

Name	Reset value	Scope	Size	Comment
<a href="#">upSwitchOnLevel</a> [▶ 350]	1	1...255	1 byte	
<a href="#">upSwitchOffLevel</a> [▶ 350]	255	1...255	1 byte	
<a href="#">downSwitchOnLevel</a> [▶ 350]	255	0...255	1 byte	
<a href="#">downSwitchOffLevel</a> [▶ 350]	0	0...255	1 byte	
<a href="#">features</a> [▶ 350]	No change	0...255	1 byte	Read only
<a href="#">gearType</a> [▶ 351]	No change	0...255	1 byte	Read only
<a href="#">switchStatus</a> [▶ 351]	No change	0...255	1 byte	Read only

##### upSwitchOnLevel

This variable is the threshold value (1...255) for switching on the output with a rising virtual output value. The threshold value is disabled with the value 255 (MASK).

If bit 3 of features is set, the variable *upSwitchOnLevel* can be set with the function block [FB\\_DALI208SetUpSwitchOnThreshold](#) [▶ 338].

The variable can be read with the function block [FB\\_DALI208QueryUpSwitchOnThreshold](#) [▶ 348].

##### upSwitchOffLevel

This variable is the threshold value (1...255) for switching off the output with a rising virtual output value. The threshold value is disabled with the value 255 (MASK).

If bit 3 of features is set, the variable *upSwitchOffLevel* can be set with the function block [FB\\_DALI208SetUpSwitchOffThreshold](#) [▶ 336].

The variable can be read with the function block [FB\\_DALI208QueryUpSwitchOffThreshold](#) [▶ 347].

##### downSwitchOnLevel

This variable is the threshold value (0...255) for switching on the output with a falling virtual output value. The threshold value is disabled with the value 255 (MASK).

If bit 3 of features is set, the variable *downSwitchOnLevel* can be set with the function block [FB\\_DALI208SetDownSwitchOnThreshold](#) [▶ 334].

The variable can be read with the function block [FB\\_DALI208QueryDownSwitchOnThreshold](#) [▶ 341].

##### downSwitchOffLevel

This variable is the threshold value (0...255) for switching off the output with a falling virtual output value. The threshold value is disabled with the value 255 (MASK).

If bit 3 of features is set, the variable *downSwitchOffLevel* can be set with the function block [FB\\_DALI208SetDownSwitchOffThreshold](#) [▶ 333].

The variable can be read with the function block [FB\\_DALI208QueryDownSwitchOffThreshold](#) [▶ 339].

##### features

The variable contains information about the implemented properties of the DALI control gear.

The variable can be read with the function block [FB\\_DALI208QueryFeatures](#) [▶ 342].

Bit	Description
0	reserve
1	reserve
2	reserve
3	Changeable threshold values.
4	reserve
5	reserve
6	reserve
7	Addressing of the physical selection is supported.

**gearType**

The variable contains information about the properties of the DALI control gear.

The variable can be read with the function block [FB\\_DALI208QueryGearType](#) [► 344].

Bit	Description
0	The output is an electronic switch.
1	The output is a relay with make contact.
2	The output is a relay with break contact.
3	The output has electronic voltage protection.
4	The input inrush current is limited.
5	Reserve
6	Reserve
7	Reserve

**switchStatus**

The variable returns additional information on the current state of the DALI control gear.

The variable can be read with the function block [FB\\_DALI208QuerySwitchStatus](#) [► 345].

Bit	Description
0	reserve
1	reserve
2...3	Last threshold value that was reached or exceeded: 00: upSwitchOnThreshold 01: upSwitchOffThreshold 10: downSwitchOnThreshold 11: downSwitchOffThreshold
4	reserve
5	reserve
6	reserve
7	reserve

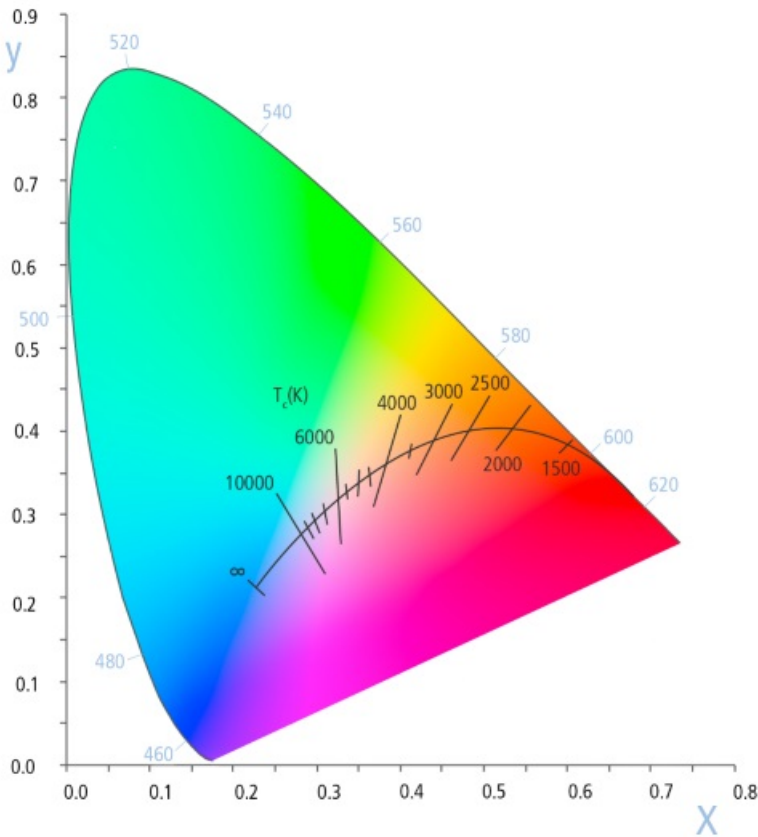
**4.1.2.7 Part 209 (color/color temperature control)**

**4.1.2.7.1 Introduction**

IEC 62386-209 describes DALI control gears for color or color temperature control. Currently, the Tc3\_DALI library contains only function blocks for color temperature setting  $T_C$  (Tunable-White), therefore the further explanations refer to this aspect.

**Representation of the color temperature**

The black-body line (BBL) or Planck curve runs within the CIE standard color chart, which represents the possible color space. Values on this curve correspond to a white hue specified in Kelvin. The higher the value, the colder the light is perceived.



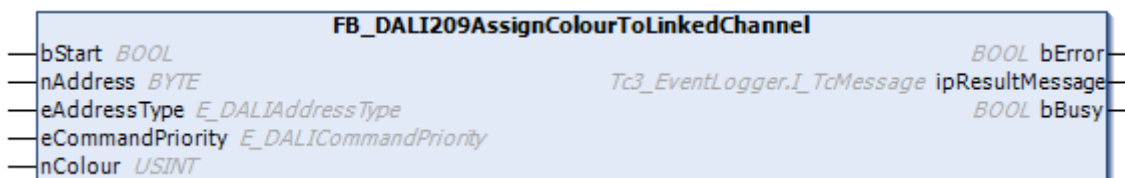
Light sources from this range combine LEDs of different color temperatures on one board. This allows the lighting effect of, for example, Human Centric Lighting (HCL) to be adjusted throughout the day. It may be necessary to use different color temperatures for the presentation of different goods (e.g. food, colorful or bright backgrounds).

**Mirek and Kelvin**

In IEC 62386, the Mirek unit is used to specify the color temperature  $T_C$ . The function blocks of the Tc3\_DALI library also use this unit. The color temperature values can be converted using the functions [KELVIN TO MIREK \[► 414\]](#) and [MIREK TO KELVIN \[► 414\]](#).

**4.1.2.7.2 Configuration commands**

**4.1.2.7.2.1 FB\_DALI209AssignColourToLinkedChannel**



Linked output channels are assigned to the defined color (see table).

The linked channels are specified by bits 0 to 5 of the variable [temporaryRGBWAFControl \[► 420\]](#). The channel assignment is not changed if [temporaryRGBWAFControl \[► 420\]](#) contains the value 255 (MASK). All TEMPORARY COLOUR SETTINGS are set to MASK after the use of this command.

Value	Description
0	No color assigned
1	Red
2	Green
3	Blue
4	White
5	Amber
6	Freely selectable color



The function block changes the DTR0 of all connected DALI control gears.



The function block supports the following color representations:

- RGBWAF

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nColour         : USINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0..63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nColour	USINT	Color value that is assigned to a channel (see table).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

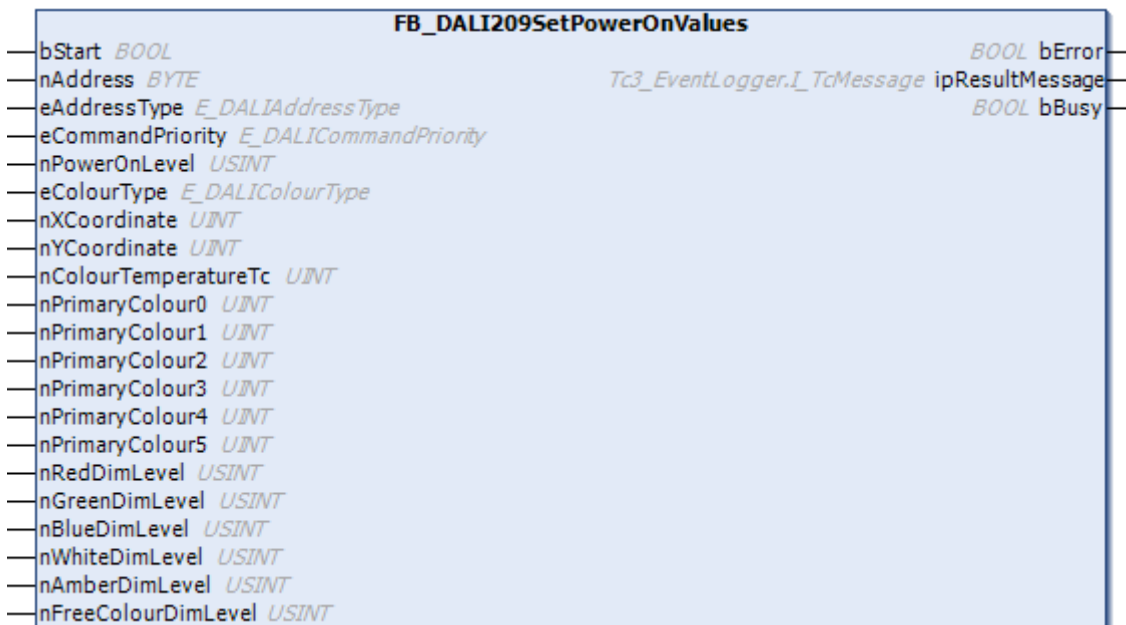
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.2.2 FB\_DALI209SetPowerOnValues**



The function block saves the values *nPowerOnLevel* and *eColourType* in the variables [powerOnLevel](#) [▶ 150] and [powerOnColourType](#) [▶ 419] of the DALI control gear. In addition, depending on the value of *eColourType*, the values of the inputs are written to the corresponding variables.



Prior to the DALI command SET POWER ON LEVEL, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nPowerOnLevel   : USINT := 254;
  eColourType     : E_DALIColourType := E_DALIColourType.MASK;
  nXCoordinate    : UINT;
  nYCoordinate    : UINT;
  nColourTemperatureTc : UINT := 65534;
  nPrimaryColour0 : UINT;
  nPrimaryColour1 : UINT;
  nPrimaryColour2 : UINT;
  nPrimaryColour3 : UINT;
  nPrimaryColour4 : UINT;
  nPrimaryColour5 : UINT;
  nRedDimLevel    : USINT;
  nGreenDimLevel  : USINT;
  nBlueDimLevel   : USINT;
  nWhiteDimLevel  : USINT;
  nAmberDimLevel  : USINT;
  nFreeColourDimLevel : USINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nPowerOnLevel	USINT	Switch-on value (0...255)
eColourType	E_DALIColourType	Defines the color setting with which the function block is operated (see <a href="#">colourType</a> [▶ 419]). MASK prevents the color representation from being changed.
nXCoordinate	UINT	Input of the x-value of the color representation according to xy color type (see <a href="#">X-coordinate</a> [▶ 418]) in the color setting Xy coordinates (see <a href="#">color representation</a> [▶ 418]).
nYCoordinate	UINT	Input of the y-value of the color representation according to xy color type (see <a href="#">Y-coordinate</a> [▶ 419]) in the color setting Xy coordinates (see <a href="#">color representation</a> [▶ 418]).
nColourTemperature Tc	UINT	Input of the desired color temperature (see <a href="#">colourTemperatureTc</a> [▶ 419]) in the setting Color temperature Tc (see <a href="#">color representation</a> [▶ 418]).
nPrimaryColourN	UINT	Input of the color representation on the corresponding output channel on the operating device in the color setting Primary color N (see <a href="#">color representation</a> [▶ 418]).
nRedDimLevel	USINT	Setting of the red dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nGreenDimLevel	USINT	Setting of the green dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nBlueDimLevel	USINT	Setting of the blue dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nWhiteDimLevel	USINT	Setting of the white dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nAmberDimLevel	USINT	Setting of the amber dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nFreeColourDimLevel	USINT	Setting of the dimming level of the freely selectable color (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).

## Outputs

```

VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR

```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.2.7.2.3 FB\_DALI209SetScene



The function block can be used to assign a specific lighting value to a scene.

**i** Prior to the DALI command SET SCENE, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see [FB\\_DALI102EnableDeviceType](#) [▶ 130]) and SET TEMPORARY COLOUR TEMPERATURE TC.



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.

**Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nSceneLevel     : USINT;
  nScene          : USINT;
  eColourType     : E_DALIColourType := E_DALIColourType.MASK;
  nXCoordinate    : UINT;
  nYCoordinate    : UINT;
  nColourTemperatureTc : UINT := 65534;
  nPrimaryColour0 : UINT;
  nPrimaryColour1 : UINT;
  nPrimaryColour2 : UINT;
  nPrimaryColour3 : UINT;
  nPrimaryColour4 : UINT;
  nPrimaryColour5 : UINT;
  nRedDimLevel    : USINT;
  nGreenDimLevel  : USINT;
  nBlueDimLevel   : USINT;
  nWhiteDimLevel  : USINT;
  nAmberDimLevel  : USINT;
  nFreeColourDimLevel : USINT;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nSceneLevel	USINT	Output value of the desired scene (0...255).
nScene	USINT	Scene whose output value is to be changed (0...15).
eColourType	E_DALIColourType	Defines the color setting with which the function block is operated (see <a href="#">colourType</a> [▶ 419]). MASK prevents the color representation from being changed.
nXCoordinate	UINT	Input of the x-value of the color representation according to xy color type (see <a href="#">X-coordinate</a> [▶ 418]) in the color setting Xy coordinates (see <a href="#">color representation</a> [▶ 418]).
nYCoordinate	UINT	Input of the y-value of the color representation according to xy color type (see <a href="#">Y-coordinate</a> [▶ 419]) in the color setting Xy coordinates (see <a href="#">color representation</a> [▶ 418]).
nColourTemperature Tc	UINT	Input of the desired color temperature (see <a href="#">colourTemperatureTc</a> [▶ 419]) in the setting Color temperature Tc (see <a href="#">color representation</a> [▶ 418]).
nPrimaryColourN	UINT	Input of the color representation on the corresponding output channel on the operating device in the color setting Primary color N (see <a href="#">color representation</a> [▶ 418]).
nRedDimLevel	USINT	Setting of the red dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nGreenDimLevel	USINT	Setting of the green dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nBlueDimLevel	USINT	Setting of the blue dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nWhiteDimLevel	USINT	Setting of the white dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nAmberDimLevel	USINT	Setting of the amber dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nFreeColourDimLevel	USINT	Setting of the dimming level of the freely selectable color (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).

 **Outputs**

```

VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.7.2.4 FB\_DALI209SetSystemFailureValues**

**FB\_DALI209SetSystemFailureValues**

<code>— bStart <i>BOOL</i></code>			<code><i>BOOL</i> bError</code>
<code>— nAddress <i>BYTE</i></code>	<code><i>Tc3_EventLogger.I_TcMessage</i></code>		<code>ipResultMessage</code>
<code>— eAddressType <i>E_DALIAddressType</i></code>			<code><i>BOOL</i> bBusy</code>
<code>— eCommandPriority <i>E_DALICommandPriority</i></code>			
<code>— nSystemFailureLevel <i>USINT</i></code>			
<code>— eColourType <i>E_DALIColourType</i></code>			
<code>— nXCoordinate <i>UINT</i></code>			
<code>— nYCoordinate <i>UINT</i></code>			
<code>— nColourTemperatureTc <i>UINT</i></code>			
<code>— nPrimaryColour0 <i>UINT</i></code>			
<code>— nPrimaryColour1 <i>UINT</i></code>			
<code>— nPrimaryColour2 <i>UINT</i></code>			
<code>— nPrimaryColour3 <i>UINT</i></code>			
<code>— nPrimaryColour4 <i>UINT</i></code>			
<code>— nPrimaryColour5 <i>UINT</i></code>			
<code>— nRedDimLevel <i>USINT</i></code>			
<code>— nGreenDimLevel <i>USINT</i></code>			
<code>— nBlueDimLevel <i>USINT</i></code>			
<code>— nWhiteDimLevel <i>USINT</i></code>			
<code>— nAmberDimLevel <i>USINT</i></code>			
<code>— nFreeColourDimLevel <i>USINT</i></code>			

The function block writes the color representation and light intensity (*nSystemFailureLevel*) of the light source to be used in the case of a system failure to the DALI control gear.

**i** Prior to the DALI command SET SYSTEM FAILURE LEVEL, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see [FB\\_DALI102EnableDeviceType](#) [▶ 130]) and SET TEMPORARY COLOUR TEMPERATURE TC.



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.

**Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nSystemFailureLevel : USINT := 254;
  eColourType     : E_DALIColourType := E_DALIColourType.MASK;
  nXCoordinate    : UINT;
  nYCoordinate    : UINT;
  nColourTemperatureTc : UINT := 65534;
  nPrimaryColour0  : UINT;
  nPrimaryColour1  : UINT;
  nPrimaryColour2  : UINT;
  nPrimaryColour3  : UINT;
  nPrimaryColour4  : UINT;
  nPrimaryColour5  : UINT;
  nRedDimLevel     : USINT;
  nGreenDimLevel   : USINT;
  nBlueDimLevel    : USINT;
  nWhiteDimLevel   : USINT;
  nAmberDimLevel   : USINT;
  nFreeColourDimLevel : USINT;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nSystemFailureLevel	USINT	Output value in the case of a system failure (0...255).
eColourType	E_DALIColourType	Defines the color setting with which the function block is operated (see <a href="#">colourType</a> [▶ 419]). MASK prevents the color representation from being changed.
nXCoordinate	UINT	Input of the x-value of the color representation according to xy color type (see <a href="#">X-coordinate</a> [▶ 418]) in the color setting Xy coordinates (see <a href="#">color representation</a> [▶ 418]).
nYCoordinate	UINT	Input of the y-value of the color representation according to xy color type (see <a href="#">Y-coordinate</a> [▶ 419]) in the color setting Xy coordinates (see <a href="#">color representation</a> [▶ 418]).
nColourTemperatureTc	UINT	Input of the desired color temperature (see <a href="#">colourTemperatureTc</a> [▶ 419]) in the setting Color temperature Tc (see <a href="#">color representation</a> [▶ 418]).
nPrimaryColourN	UINT	Input of the color representation on the corresponding output channel on the operating device in the color setting Primary color N (see <a href="#">color representation</a> [▶ 418]).
nRedDimLevel	USINT	Setting of the red dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nGreenDimLevel	USINT	Setting of the green dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nBlueDimLevel	USINT	Setting of the blue dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nWhiteDimLevel	USINT	Setting of the white dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nAmberDimLevel	USINT	Setting of the amber dimming level (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).
nFreeColourDimLevel	USINT	Setting of the dimming level of the freely selectable color (see <a href="#">DimLevel</a> [▶ 419]) in the color setting RGBWAF (see <a href="#">color representation</a> [▶ 418]).

## Outputs

```

VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

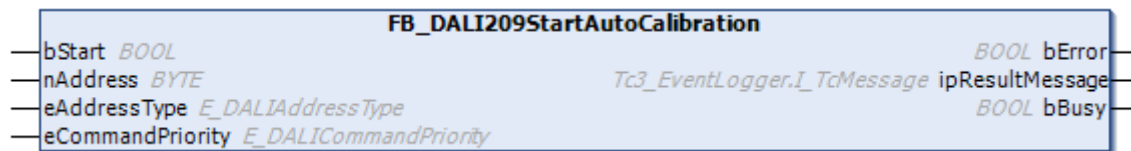
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.7.2.5 FB\_DALI209StartAutoCalibration**



The calibration procedure is started in order to measure the x-coordinate, the y-coordinate and the TY value of all supported primary colors.

The command starts a 15-minute timer or initiates it again. Bit 2 of the [colourStatus \[► 420\]](#) variable is 1 as long as the timer is active (see [FB\\_DALI209QueryColourStatus \[► 398\]](#)). On expiry of the timer the last color representation, the last color value and the last lamp power level are directly saved again.

During the timer period the DALI control gear carries out a calibration procedure in order to measure the x-coordinate, the y-coordinate and the TY value of all supported primary colors. Whilst the calibration procedure is running, the DALI control gear does not react to any commands apart from TERMINATE, QUERY COLOUR STATUS and START AUTO CALIBRATION. In addition, bit 3 in the variable [colourStatus \[► 420\]](#) is set to 0 at the start of the calibration. The TERMINATE command ends the procedure and stops the timer.

If the calibration was successful, bit 3 in [colourStatus \[► 420\]](#) is set to 1 and the timer is stopped. If the calibration was not successful, then the last successful calibration data are restored if the DALI control gear is able to do so. Bit 3 of [colorStatus \[► 420\]](#) is then set to 1. The ability to restore the last successful calibration data is a feature of the operating device (see [FB\\_DALI209QueryGearFeaturesStatus \[► 402\]](#) command).

Due to the fact that the calibration can take longer than 15 minutes, the status of the automatic calibration should be checked periodically using the QUERY COLOR STATUS command and the calibration timer restarted with the START AUTO CALIBRATION command. The calibration procedure is started in order to measure the x-coordinate, the y-coordinate and the TY value of all supported primary colors.

The command starts a 15-minute timer or initiates it again. Bit 2 of the `colourStatus` [▶ 420] variable is 1 as long as the timer is active (see `FB_DALI209QueryColourStatus` [▶ 398]). On expiry of the timer the last color representation, the last color value and the last lamp power level are directly saved again.

During the timer period the DALI control gear carries out a calibration procedure in order to measure the x-coordinate, the y-coordinate and the TY value of all supported primary colors. Whilst the calibration procedure is running, the DALI control gear does not react to any commands apart from TERMINATE, QUERY COLOUR STATUS and START AUTO CALIBRATION. In addition, bit 3 in the variable `colourStatus` [▶ 420] is set to 0 at the start of the calibration. The TERMINATE command ends the procedure and stops the timer.

If the calibration was successful, bit 3 in `colourStatus` [▶ 420] is set to 1 and the timer is stopped. If the calibration was not successful, then the last successful calibration data are restored if the DALI control gear is able to do so. Bit 3 of `colorStatus` [▶ 420] is then set to 1. The ability to restore the last successful calibration data is a feature of the operating device (see `FB_DALI209QueryGearFeaturesStatus` [▶ 402] command).

Due to the fact that the calibration can take longer than 15 minutes, the status of the automatic calibration should be checked periodically using the QUERY COLOR STATUS command and the calibration timer restarted with the START AUTO CALIBRATION command (if necessary).



The function block supports the following color representations:

- xy coordinates
- Color temperature Tc
- Primary (color) N
- RGBWAF

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

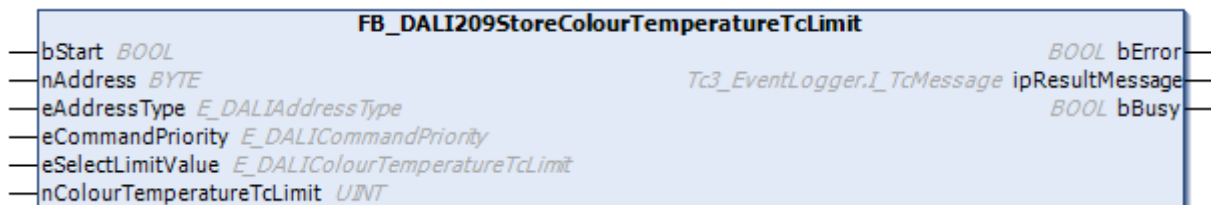
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.2.6 FB\_DALI209StoreColourTemperatureTcLimit**



Depending on the specification of *eSelectLimitValue*, the function block saves the value *nColourTemperatureTcLimit* in the variables *colourTemperatureTcCoolest*, *colourTemperatureTcWarmest*, *colourTemperatureTcPhysicalCoolest* or *colourTemperatureTcPhysicalWarmest* of the DALI control gear.

The values can be read with function block [FB\\_DALI209QueryColourValue \[► 401\]](#).

**i** Prior to the DALI command STORE COLOUR TEMPERATURE Tc LIMIT, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[► 130\]](#)).

**i** The function block changes the DTR0, DTR1 and DTR2 of all connected DALI control gears.

**i** The function block sends the DALI command STORE COLOUR TEMPERATURE Tc LIMIT twice, as foreseen by the DALI standard for certain DALI commands.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eSelectLimitValue : E_DALIColourTemperatureTcLimit;
  nColourTemperatureTcLimit : UINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eSelectLimitValue	E_DALIColourTemperatureTcLimit [▶ 855]	Specifies the limit value to be set.
nColourTemperatureTcLimit	UINT	The value that is written into the selected variable.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

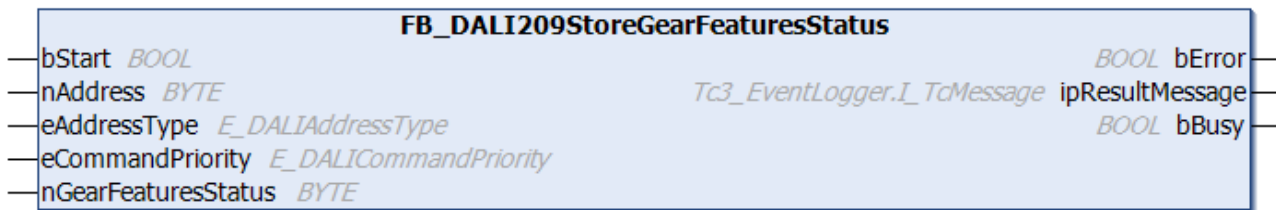
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.7.2.7 FB\_DALI209StoreGearFeaturesStatus



The function block saves the value *nGearFeaturesStatus* in the variable [gearFeatures/Status \[► 420\]](#) of the DALI control gear.

The value can be read with the function block [FB\\_DALI209QueryGearFeaturesStatus \[► 402\]](#).

If Bit 0 is set to 1, all commands for controlling the lamp power – with the exception of ENABLE DAPC SEQUENCE – automatically initiate a color transition.

**i** Prior to the DALI command STORE GEAR FEATURES/STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[► 130\]](#)).

**i** The function block changes the DTR0 of all connected DALI control gears.

**i** The function block sends the DALI command STORE GEAR FEATURES/STATUS twice, as foreseen by the DALI standard for certain DALI commands.

 Inputs

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nGearFeaturesStatus : BYTE;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nGearFeaturesStatus	BYTE	Value written to the variable <code>gearFeatures/Status</code> [▶ 420].

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.7.2.8 FB\_DALI209StoreTYPrimaryN



The function block stores the value `nTYPrimaryN` in the variable `tyPrimaryN` [▶ 419] of the DALI control gear.

The value is expressed in units of 0.5 lm, which results in a possible range of  $TY_{min} = 0 \text{ lm}$  to  $TY_{max} = 32767 \text{ lm}$ . A value of 65535 (MASK) means "unknown". The `nPrimaryColour` parameter specifies the primary color and must be within the range of 0 to 5, depending on the available number of primary colors. The command is ignored for every other value.



The function block changes the DTR0, DTR1 and DTR2 of all connected DALI control gears.



The function block supports the following color representations:

- Primary (color) N

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nPrimaryColour  : USINT;
  nTYPrimaryN     : UINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nPrimaryColour	USINT	Primary color (0...5).
nTyPrimaryN	UINT	Value written to variable <code>tyPrimaryN</code> [▶ 419].

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

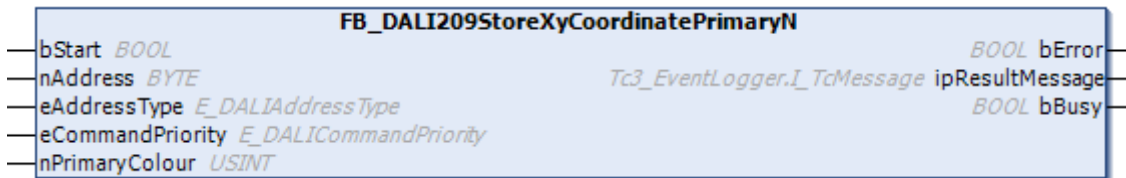
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.2.9 FB\_DALI209StoreXyCoordinatePrimaryN**



The function block copies the value from the variables [temporaryXCoordinate](#) [▶ 418] and [temporaryYCoordinate](#) [▶ 419] to the variables [xCoordinatePrimaryN](#) [▶ 419] and [yCoordinatePrimaryN](#) [▶ 419].

The *nPrimaryColour* parameter specifies the primary color and must be within the range of 0 to 5, depending on the available number of primary colors. The command is ignored for every other value.

This command can be used to store the current xy coordinates associated with the primary color. xy coordinates outside the color space chromaticity diagram are not meaningful and should therefore be avoided.



The function block changes the DTR2 of all connected DALI control gears.



The function block supports the following color representations:

- Primary (color) N

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nPrimaryColour  : USINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nPrimaryColour	USINT	Primary color (0...5).

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

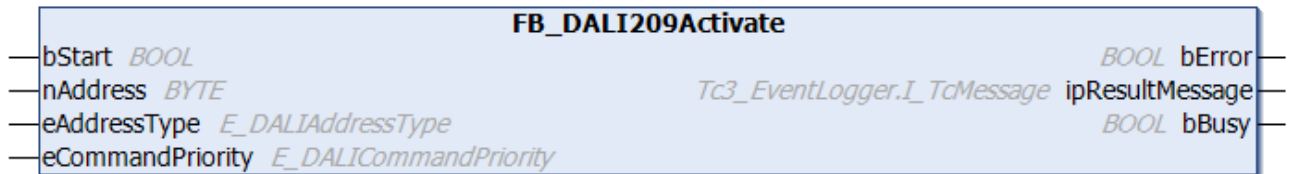
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.3 Control commands**

**4.1.2.7.3.1 FB\_DALI209Activate**



The function block causes the buffered values of the temporary registers to be transferred to the lamp. It ends a running cross-fade and starts a new cross-fade for the respective color/color temperature.

**i** Prior to the DALI command ACTIVATE, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶\_130]).

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶_854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶_856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

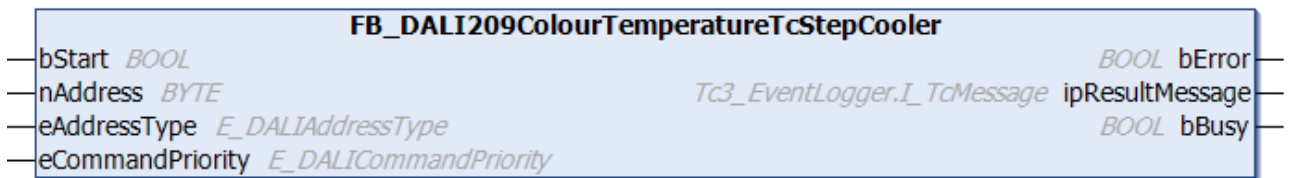
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.7.3.2 FB\_DALI209ColourTemperatureTcStepCooler**



Due to the function block the value [colourTemperatureTc](#) [▶ 419] is reduced by 1 Mirek without cross-fading. If the [colourTemperatureTc](#) [▶ 419] value already has the same value as [colourTemperatureTcCoolest](#) [▶ 419], no change takes place.

Bit 1 (Colour temperature Tc out of range) is set in [colourStatus](#) [▶ 420] if the color temperature cannot be reached by the DALI control gear. This command is executed by the DALI control gear only if bit 5 (Colour type color temperature Tc active) is set in the variable [colourStatus](#) [▶ 420].

The functions [KELVIN TO MIREK](#) [▶ 414] and [MIREK TO KELVIN](#) [▶ 414] are available for converting from or to Kelvin.

**i** Prior to the DALI command COLOUR TEMPERATURE Tc STEP COOLER, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

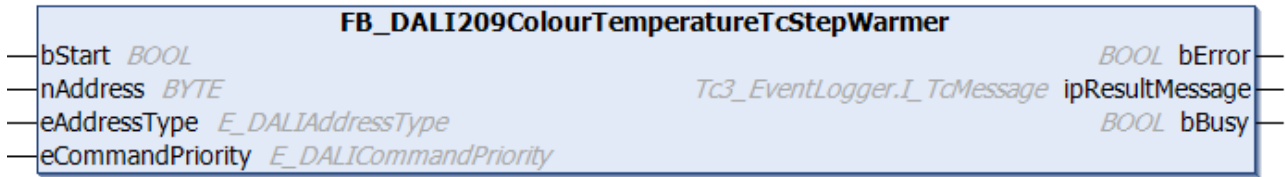
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.7.3.3 FB\_DALI209ColourTemperatureTcStepWarmer



Due to the function block the value `colourTemperatureTc` [▶ 419] is increased by 1 Mirek without cross-fading. If the `colourTemperatureTc` [▶ 419] value already has the same value as `colourTemperatureTcWarmest` [▶ 419], no change takes place.

Bit 1 (Colour temperature Tc out of range) is set in `colourStatus` [▶ 420] if the color temperature cannot be reached by the DALI control gear. This command is executed by the DALI control gear only if bit 5 (Colour type color temperature Tc active) is set in the variable `colourStatus` [▶ 420].

The functions `KELVIN_TO_MIREK` [▶ 414] and `MIREK_TO_KELVIN` [▶ 414] are available for converting from or to Kelvin.

**i** Prior to the DALI command COLOUR TEMPERATURE Tc STEP WARMER, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also `FB_DALI102EnableDeviceType` [▶ 130]).

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

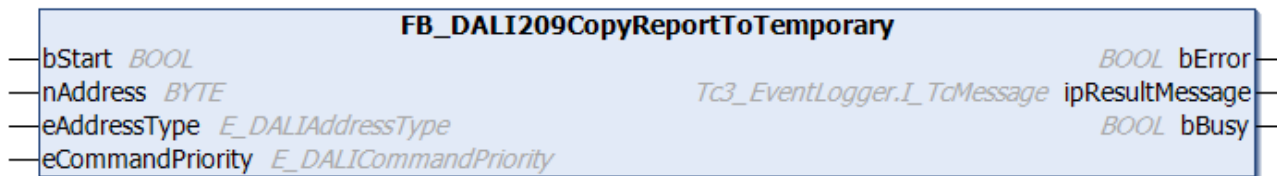
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.7.3.4 FB\_DALI209CopyReportToTemporary**



The function block copies the contents of the variables with the color settings to the variables for the temporary color settings.

**i** Prior to the DALI command COPY REPORT TO TEMPORARY, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application-related extension commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.7.3.5 FB\_DALI209SetColourTemperatureTc



The function block saves the value *nColourTemperatureTc* in the variable *colourTemperatureTc* [▶ 419] of the DALI control gear and causes the set color temperature to be applied to the lamp by executing the ACTIVATE command.

The value can be read with function block [FB\\_DALI209QueryColourValue](#) [▶ 401].

The value is expressed in units of 1 Mirek. A value of 0 is ignored and therefore not saved. The color temperature Tc can vary from 1 Mirek (1000000 K) to 65534 Mirek (15.26 K).

The functions [KELVIN TO MIREK](#) [▶ 414] and [MIREK TO KELVIN](#) [▶ 414] are available for converting from or to Kelvin.

**i** Prior to the DALI command SET TEMPORARY COLOUR TEMPERATURE Tc, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]). To activate the color temperature, the function block sends the command ACTIVATE (see [FB\\_DALI209Activate](#) [▶ 372]).

**i** The function block changes the DTR0 and the DTR1 of all connected DALI control gears.

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nColourTemperatureTc : UINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nColourTemperatureTc	UINT	The value written to the variable <i>ColourTemperatureTc</i> and transferred directly to the output value (0...65534 Mirek, 65535).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.7.3.6 FB\_DALI209SetTemporaryColourTemperatureTc**



The function block saves the value *nTemporaryColourTemperatureTc* in the variable [temporaryTemperatureTc](#) [▶ 419] of the DALI control gear.

The value can be read with function block [FB\\_DALI209QueryColourValue](#) [▶ 401].

The value is expressed in units of 1 Mirek. A value of 0 is ignored and therefore not saved. The color temperature Tc can vary from 1 Mirek (1000000 K) to 65534 Mirek (15.26 K).

The functions [KELVIN TO MIREK](#) [▶ 414] and [MIREK TO KELVIN](#) [▶ 414] are available for converting from or to Kelvin.



Prior to the DALI command SET TEMPORARY COLOUR TEMPERATURE Tc, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).





The function block changes the DTR0 and the DTR1 of all connected DALI control gears.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nTemporaryColourTemperatureTc : UINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nTemporaryColourTemperatureTc	UINT	The value that is written to the variable <i>temporaryColourTemperatureTc</i> <a href="#">▶ 419</a> (0...65534 Mirek, 65535).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.7.3.7 FB\_DALI209SetTemporaryPrimaryNDimLevel



The function block saves the value *nTemporaryPrimaryNDimLevel* in the variable *temporaryPrimaryNDimLevel* [[▶ 419](#)] of the DALI control gear.

The value can be read with function block [FB\\_DALI209QueryColourValue](#) [[▶ 401](#)].

The value is expressed in steps of 1 / 65536. The maximum value of the variable *temporaryPrimaryNDimLevel* is 0.99997 and is mapped on a linear scale.



The function block changes the DTR0, DTR1 and DTR2 of all connected DALI control gears.



The function block supports the following color representations:

- Primary (color) N

 Inputs

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nPrimaryColour  : USINT;
  nTemporaryPrimaryNDimLevel : UINT;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [ <a href="#">▶ 854</a> ]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [ <a href="#">▶ 856</a> ]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nPrimaryColour	USINT	Primary color (0...5).
nTemporaryPrimaryNDimLevel	UINT	The value that is written to the variable <code>temporaryPrimaryNDimLevel</code> [▶ 419].

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <code>ipResultMessage</code> . The output is set to FALSE again as soon as <code>bBusy</code> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <code>bBusy</code> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

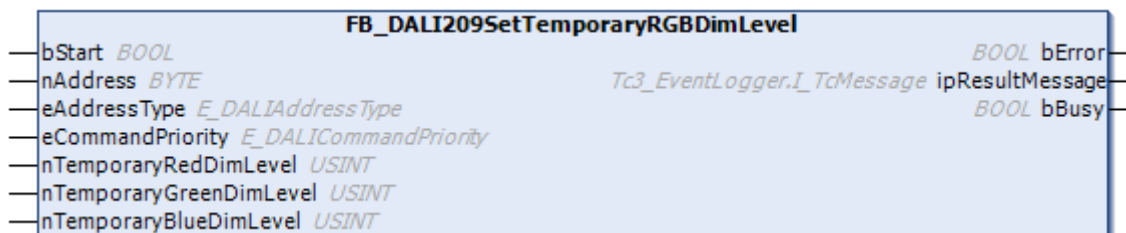
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.3.8 FB\_DALI209SetTemporaryRGBDimLevel**



The function block saves the values `nTemporaryRedDimLevel`, `nTemporaryGreenDimLevel` and `nTemporaryBlueDimLevel` in the variables `temporaryRedDimLevel` [▶ 419], `temporaryGreenDimLevel` [▶ 419] and `temporaryBlueDimLevel` [▶ 419] of the DALI control gear.

The values can be read with function block [FB\\_DALI209QueryColourValue](#) [▶ 401].



The function block changes the DTR0, DTR1 and DTR2 of all connected DALI control gears.



The function block supports the following color representations:

- RGBWAF

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nTemporaryRedDimLevel : USINT;
  nTemporaryGreenDimLevel : USINT;
  nTemporaryBlueDimLevel : USINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nTemporaryRedDimLevel	USINT	The value that is written to the variable <a href="#">temporaryRedDimLevel</a> [▶ 419].
nTemporaryGreenDimLevel	USINT	The value that is written to the variable <a href="#">temporaryGreenDimLevel</a> [▶ 419].
nTemporaryBlueDimLevel	USINT	The value that is written to the variable <a href="#">temporaryBlueDimLevel</a> [▶ 419].

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

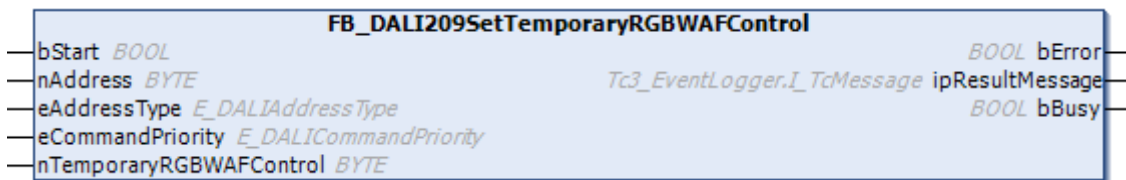
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.3.9 FB\_DALI209SetTemporaryRGBWAFControl**



The function block saves the value *nTemporaryRGBWAFControl* in the variable [temporaryRGBWAFControl](#) [▶ 420] of the DALI control gear.

The value can be read with function block [FB\\_DALI209QueryColourValue](#) [▶ 401]. The *nTemporaryRGBWAFControl* input thereby contains the new assignment.



The function block changes the DTR0 of all connected DALI control gears.



The function block supports the following color representations:

- RGBWAF

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nTemporaryRGBWAFControl : BYTE;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nTemporaryRGBWAFControl	BYTE	Contains the channel assignment (see <a href="#">temporaryRGBWAFControl</a> [▶ 420]).

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

### 4.1.2.7.3.10 FB\_DALI209SetTemporaryWAFDimLevel



The function block saves the values *nTemporaryWhiteDimLevel*, *nTemporaryAmberDimLevel* and *nTemporaryFreeColourDimLevel* in the variables [temporaryWhiteDimLevel \[▶ 419\]](#), [temporaryAmberDimLevel \[▶ 419\]](#) and [temporaryFreeColourDimLevel \[▶ 419\]](#) of the DALI control gear.

The values can be read with function block [FB\\_DALI209QueryColourValue \[▶ 401\]](#).



The function block changes the DTR0, DTR1 and DTR2 of all connected DALI control gears.



The function block supports the following color representations:

- RGBWAF

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nTemporaryWhiteDimLevel : USINT;
  nTemporaryAmberDimLevel : USINT;
  nTemporaryFreeColourDimLevel : USINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nTemporaryWhiteDimLevel	USINT	The value that is written to the variable <a href="#">temporaryWhiteDimLevel [▶ 419]</a> .
nTemporaryAmberDimLevel	USINT	The value that is written to the variable <a href="#">temporaryAmberDimLevel [▶ 419]</a> .
nTemporaryFreeColourDimLevel	USINT	The value that is written to the variable <a href="#">temporaryFreeColourDimLevel [▶ 419]</a> .

**🔌 Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

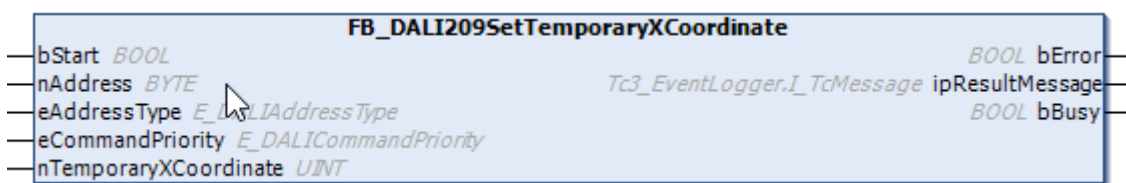
**📄 Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.3.11 FB\_DALI209SetTemporaryXCoordinate**



The function block saves the value *nTemporaryXCoordinate* in the variable [temporaryXCoordinate \[▶ 418\]](#) of the DALI control gear.

The value can be read with function block [FB\\_DALI209QueryColourValue \[▶ 401\]](#).



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.



The function block supports the following color representations:

- xy coordinates
- Primary (color) N

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nTemporaryXCoordinate : UINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#"> &gt; 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#"> &gt; 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nTemporaryXCoordinate	UINT	The value that is written to the variable <a href="#">temporaryXCoordinate  &gt; 418</a> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation  &gt; 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages  &gt; 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

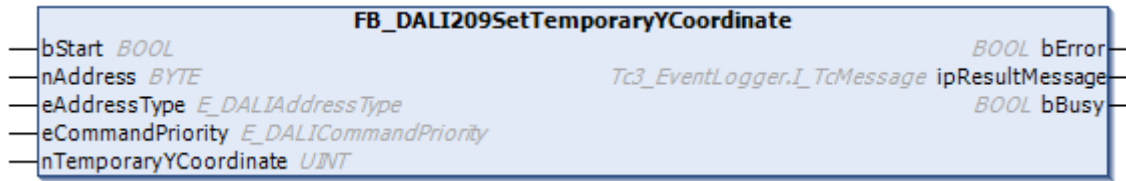
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block  &gt; 892</a> ).



Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

4.1.2.7.3.12 FB\_DALI209SetTemporaryYCoordinate



The function block saves the value *nTemporaryYCoordinate* in the variable temporaryYCoordinate [▶ 419] of the DALI control gear.

The value can be read with function block FB\_DALI209QueryColourValue [▶ 401].



The function block changes the DTR0 and the DTR1 of all connected DALI control gears.



The function block supports the following color representations:

- xy coordinates
- Primary (color) N

Inputs

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority := E_DALICommandPriority.MiddleLow;
  nTemporaryYCoordinate : UINT;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<u>E_DALIAddressType</u> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<u>E_DALICommandPriority</u> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nTemporaryYCoordinate	UINT	The value that is written to the variable <u>temporaryYCoordinate</u> [▶ 419].

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

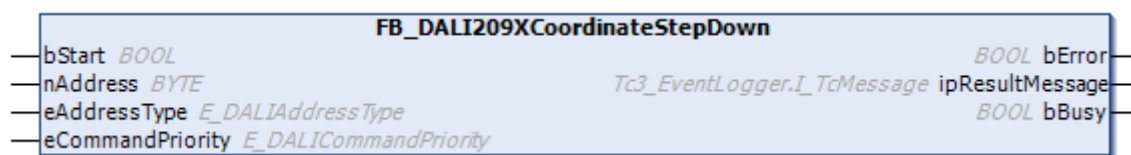
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.3.13 FB\_DALI209XCoordinateStepDown**



The function block reduces the variable [xCoordinate](#) [▶ 418] by 256 steps (256 / 65536) without cross-fading.

If the new color value does not correspond to a color that can be achieved by the DALI control gear, this must be indicated by bit 0 of [colourStatus](#) [▶ 420] (xy-coordinate color point lies outside the valid range). The command is executed only if bit 4 of [colourStatus](#) [▶ 420] (color representation xy-coordinate active) is set.



The function block supports the following color representations:

- xy coordinates

**Inputs**

```
VAR_INPUT
  bStart : BOOL;
  nAddress : BYTE;
```

```
eAddressType      : E_DALIAddressType := E_DALIAddressType.Short;
eCommandPriority  : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

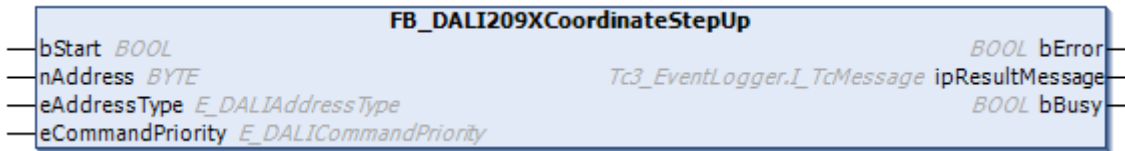
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

### 4.1.2.7.3.14 FB\_DALI209XCoordinateStepUp



The function block increases the variable `xCoordinate` [▶ 418] by 256 steps (256 / 65536) without cross-fading.

If the new color value does not correspond to a color that can be achieved by the DALI control gear, this must be indicated by bit 0 of `colourStatus` [▶ 420] (xy-coordinate color point lies outside the valid range). The command is executed only if bit 4 of `colourStatus` [▶ 420] (color representation xy-coordinate active) is set.



The function block supports the following color representations:

- xy coordinates

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

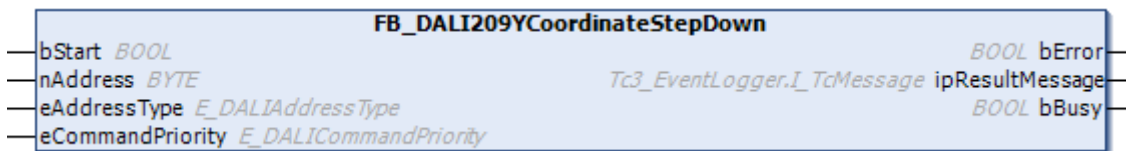
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.3.15 FB\_DALI209YCoordinateStepDown**



The function block reduces the variable [yCoordinate](#) [▶ 419] by 256 steps (256 / 65536) without cross-fading.

If the new color value does not correspond to a color that can be achieved by the DALI control gear, this must be indicated by bit 0 of [colourStatus](#) [▶ 420] (xy-coordinate color point lies outside the valid range). The command is executed only if bit 4 of [colourStatus](#) [▶ 420] (color representation xy-coordinate active) is set.



The function block supports the following color representations:

- xy coordinates

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType := E_DALIAddressType.Short;
  eCommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy            : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



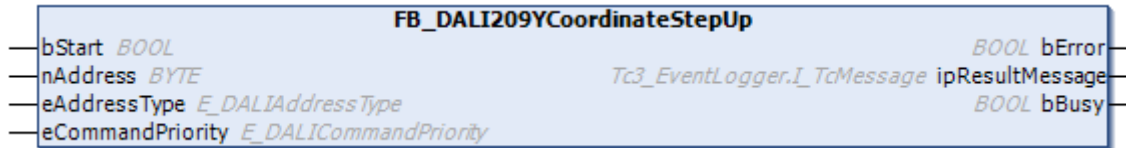
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

### 4.1.2.7.3.16 FB\_DALI209YCoordinateStepUp



The function block increases the variable `yCoordinate` [▶ 419] by 256 steps (256 / 65536) without cross-fading.

If the new color value does not correspond to a color that can be achieved by the DALI control gear, this must be indicated by bit 0 of `colourStatus` [▶ 420] (xy-coordinate color point lies outside the valid range). The command is executed only if bit 4 of `colourStatus` [▶ 420] (color representation xy-coordinate active) is set.



The function block supports the following color representations:

- xy coordinates

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

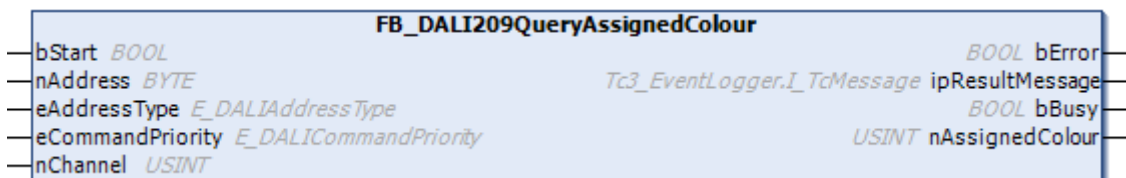
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.4 Query commands**

**4.1.2.7.4.1 FB\_DALI209QueryAssignedColour**



The function block reads the variable [assignedColour \[▶ 419\]](#) from the DALI control gear.

This contains the color assigned to the specified output channel (0...5). If a non-existent channel number is specified, 255 (MASK) is returned.



The function block changes the DTR0 of all connected DALI control gears.



The function block supports the following color representations:

- RGBWAF

 **Inputs**

```
VAR_INPUT
    bStart      : BOOL;
    nAddress    : BYTE;
    eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
```



```
eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
nChannel        : USINT;
```

END\_VAR

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nChannel	USINT	Channel number (0 - 5).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nAssignedColour : USINT;
```

END\_VAR

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nAssignedColour	USINT	Assigned color of the channel (see table above).

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

### 4.1.2.7.4.2 FB\_DALI209QueryColourStatus



The function block reads the variable `colourStatus` [► 420] from the DALI control gear.

**i** Prior to the DALI command QUERY COLOUR STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [► 130]).

The output `nColourStatus` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nColourStatus : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nColourStatus	BYTE	Output of the status information (see <a href="#">colourStatus [▶ 420]</a> ).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.7.4.3 FB\_DALI209QueryColourTypeFeatures**



The function block reads the variable [colourTypeFeatures \[▶ 420\]](#) from the DALI control gear.

This contains the color representations supported by the DALI control gear.

**i** Prior to the DALI command QUERY COLOUR TYPE FEATURES, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType \[▶ 130\]](#)).

The output *nColourTypeFeatures* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nColourTypeFeatures : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nColourTypeFeatures	BYTE	Information about the color representations supported by the DALI control gear (see <a href="#">colourTypeFeatures [▶ 420]</a> ).



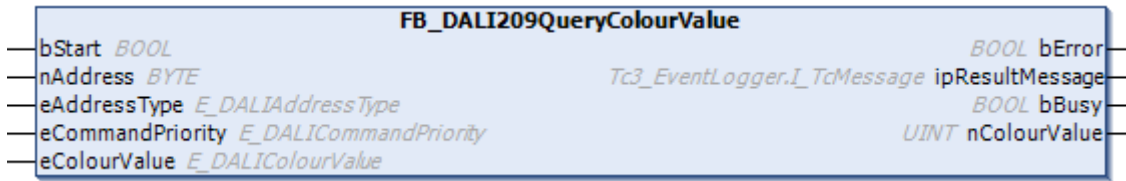
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.7.4.4 FB\_DALI209QueryColourValue



The function block reads the specified variable (color value) from the DALI control gear. The value to be read is defined by *eColourValue*.

Certain variables can be read directly by DALI commands (e.g. [FB\\_DALI209QueryColourStatus](#) [▶ 398]). Further details on the variables can be found in section [Variables](#) [▶ 415].

The response must be 255 (MASK) if the DALI control gear does not know the coordinates or if the primary color is not present.

**i** Prior to the DALI command QUERY COLOUR VALUE, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

**i** The function block changes the DTR0 of all connected DALI control gears.

The output *nColourValue* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  eColourValue    : E_DALIColourValue;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
eColourValue	<a href="#">E_DALIColourValue</a> [▶ 855]	Variable to be read from the DALI control gear.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nColourValue   : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nColourValue	UINT	Contains the value of the read variable.

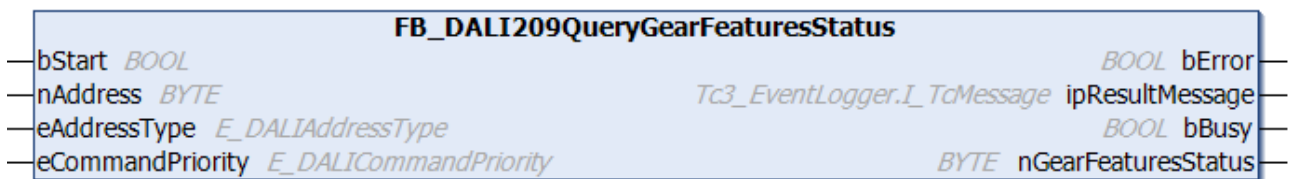
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.7.4.5 FB\_DALI209QueryGearFeaturesStatus**



The function block reads the variable [gearFeatures/Status](#) [▶ 420] from the DALI control gear.

**i** Prior to the DALI command QUERY GEAR FEATURES/STATUS, the function block sends the DALI command ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see also [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

The output *nGearFeaturesStatus* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart : BOOL;
  nAddress : BYTE;
```

```
eAddressType      : E_DALIAddressType := E_DALIAddressType.Short;
eCommandPriority  : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">[► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nGearFeatureStatus : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nGearFeatureStatus	BYTE	Output of the status information (see <a href="#">gearFeatures/Status [► 420]</a> ).

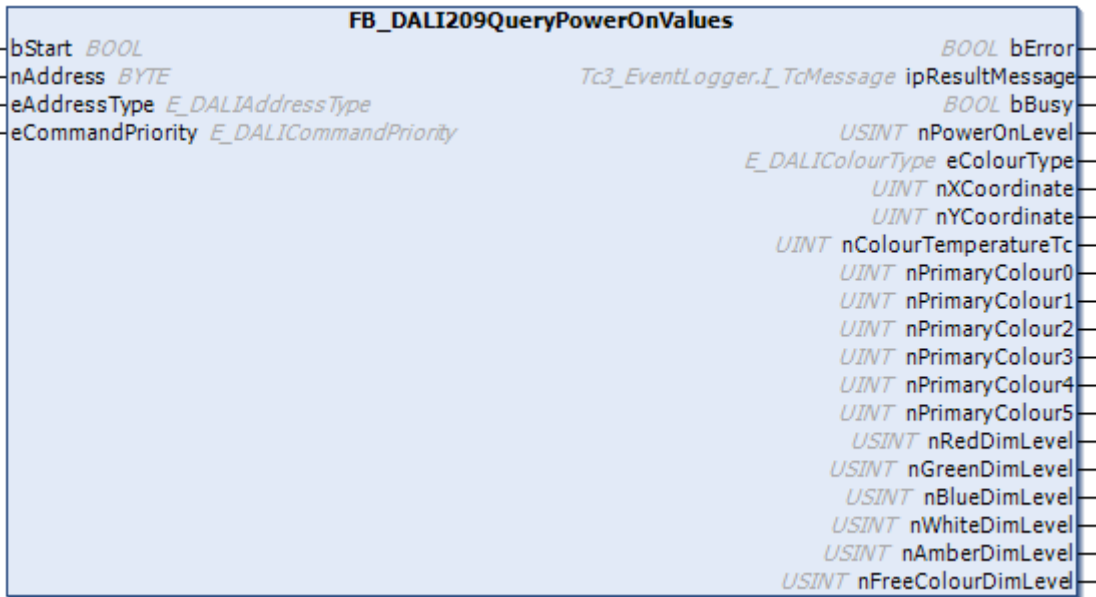
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.7.4.6 FB\_DALI209QueryPowerOnValues



The function block reads the variables `powerOnLevel` [► 150] and the supported values of the color settings from the DALI control gear.

**i** In addition to the DALI command QUERY POWER ON LEVEL, the function block sends the DALI commands QUERY COLOUR VALUE and ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see `FB_DALI102EnableDeviceType` [► 130]).

**i** The function block changes the DTR0 of all connected DALI control gears.

The outputs only contain valid values if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.



 **Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage      : I_TcMessage;
  bBusy                : BOOL;
  nPowerOnLevel        : USINT;
  eColourType          : E_DALIColourType;
  nXCoordinate         : UINT;
  nYCoordinate         : UINT;
  nColourTemperatureTc : UINT;
  nPrimaryColour0      : UINT;
  nPrimaryColour1      : UINT;
  nPrimaryColour2      : UINT;
  nPrimaryColour3      : UINT;
  nPrimaryColour4      : UINT;
  nPrimaryColour5      : UINT;
  nRedDimLevel         : USINT;
  nGreenDimLevel       : USINT;
  nBlueDimLevel        : USINT;
  nWhiteDimLevel       : USINT;
  nAmberDimLevel       : USINT;
  nFreeColourDimLevel  : USINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nPowerOnLevel	USINT	Output of the switch-on value (see <a href="#">powerOnLevel [► 150]</a> ) from the DALI control gear (0...255).
eColourType	E_DALIColourType	Specifies the color setting with which the DALI control gear is operated (see <a href="#">colourType [► 419]</a> ).
nXCoordinate	UINT	Output of the x-value of the color representation according to xy color type (see <a href="#">X-coordinate [► 418]</a> ) in the color setting Xy coordinates (see <a href="#">color representation [► 418]</a> ).
nYCoordinate	UINT	Output of the y-value of the color representation according to xy color type (see <a href="#">Y-coordinate [► 419]</a> ) in the color setting Xy coordinates (see <a href="#">color representation [► 418]</a> ).
nColourTemperature Tc	UINT	Output of the color temperature (see <a href="#">colourTemperatureTc [► 419]</a> ) in the setting Color temperature Tc (see <a href="#">color representation [► 418]</a> ).
nPrimaryColourN	UINT	Output of the color representation on the corresponding output channel on the operating device in the color setting Primary color N (see <a href="#">color representation [► 418]</a> ).
nRedDimLevel	USINT	Output of the red dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nGreenDimLevel	USINT	Output of the green dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nBlueDimLevel	USINT	Output of the blue dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nWhiteDimLevel	USINT	Output of the white dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nAmberDimLevel	USINT	Output of the amber dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nFreeColourDimLevel	USINT	Output of the dimming level of the freely selectable color (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).

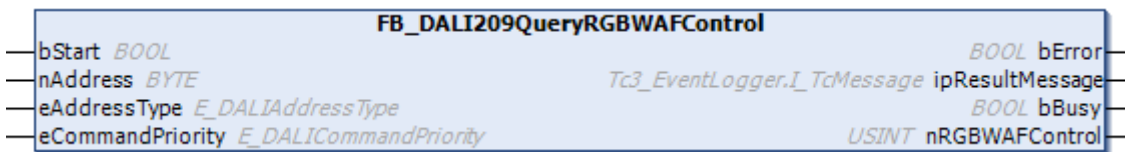
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.2.7.4.7 FB\_DALI209QueryRGBWAFControl



The function block reads the variable [RGBWAFControl \[► 420\]](#) from the DALI control gear.

If an output channel or a color is not supported, then the corresponding bit is FALSE.



The function block supports the following color representations:

- RGBWAF

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
```

```
bBusy          : BOOL;
nRGBWAFColour : USINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRGBWAFColour	USINT	Information about the channel assignment (see <a href="#">RGBWAFControl</a> [▶ 420]).

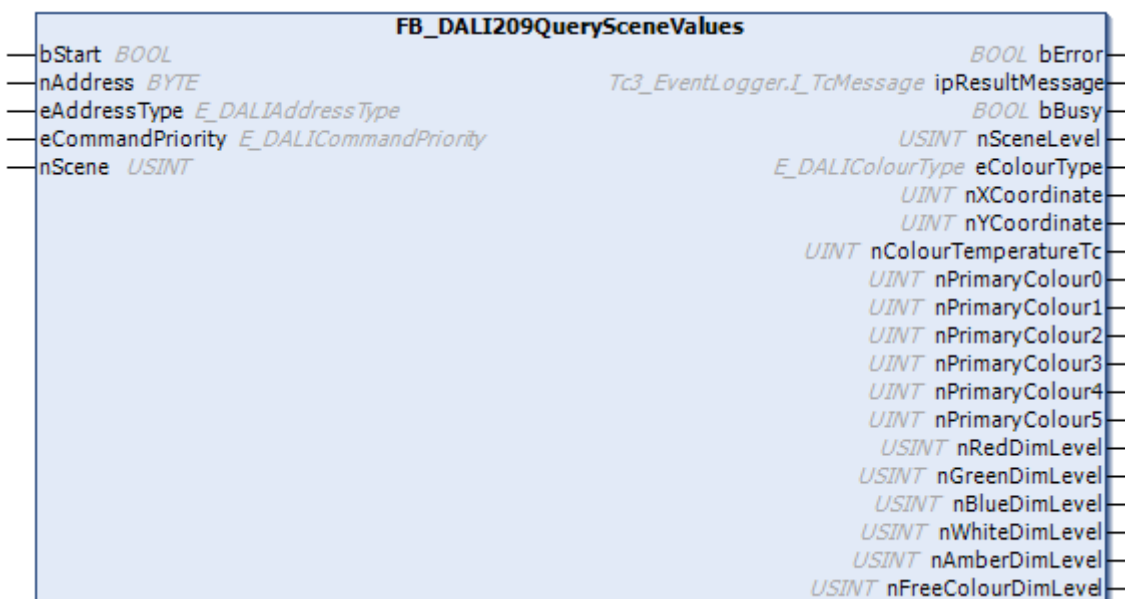
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.55	Tc3_DALI from v3.18.1.0

**4.1.2.7.4.8 FB\_DALI209QuerySceneValues**



The function block reads the output values from the DALI control gear for the specified scene *nScene*.

**i** In addition to the DALI command QUERY SCENE LEVEL, the function block sends the DALI commands QUERY COLOUR VALUE and ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see [FB\\_DALI102EnableDeviceType](#) [▶ 130]).

**i** The function block changes the DTR0 of all connected DALI control gears.

The outputs only contain valid values if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nScene         : USINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** *eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nScene	USINT	Scene whose output value is to be read (0...15).

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nSceneLevel     : USINT;
  eColourType     : E_DALIColourType;
  nXCoordinate    : UINT;
  nYCoordinate    : UINT;
  nColourTemperatureTc : UINT;
  nPrimaryColour0 : UINT;
  nPrimaryColour1 : UINT;
  nPrimaryColour2 : UINT;
  nPrimaryColour3 : UINT;
  nPrimaryColour4 : UINT;
  nPrimaryColour5 : UINT;
  nRedDimLevel    : USINT;
  nGreenDimLevel  : USINT;
  nBlueDimLevel   : USINT;
  nWhiteDimLevel  : USINT;
  nAmberDimLevel  : USINT;
  nFreeColourDimLevel : USINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nSceneLevel	USINT	Output value of the desired scene (0...255).
eColourType	E_DALIColourType	Specifies the color setting with which the DALI control gear is operated (see <a href="#">colourType [► 419]</a> ).
nXCoordinate	UINT	Output of the x-value of the color representation according to xy color type (see <a href="#">X-coordinate [► 418]</a> ) in the color setting Xy coordinates (see <a href="#">color representation [► 418]</a> ).
nYCoordinate	UINT	Output of the y-value of the color representation according to xy color type (see <a href="#">Y-coordinate [► 419]</a> ) in the color setting Xy coordinates (see <a href="#">color representation [► 418]</a> ).
nColourTemperature Tc	UINT	Output of the color temperature (see <a href="#">colourTemperatureTc [► 419]</a> ) in the setting Color temperature Tc (see <a href="#">color representation [► 418]</a> ).
nPrimaryColourN	UINT	Output of the color representation on the corresponding output channel on the operating device in the color setting Primary color N (see <a href="#">color representation [► 418]</a> ).
nRedDimLevel	USINT	Output of the red dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nGreenDimLevel	USINT	Output of the green dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nBlueDimLevel	USINT	Output of the blue dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nWhiteDimLevel	USINT	Output of the white dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nAmberDimLevel	USINT	Output of the amber dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nFreeColourDimLevel	USINT	Output of the dimming level of the freely selectable color (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).

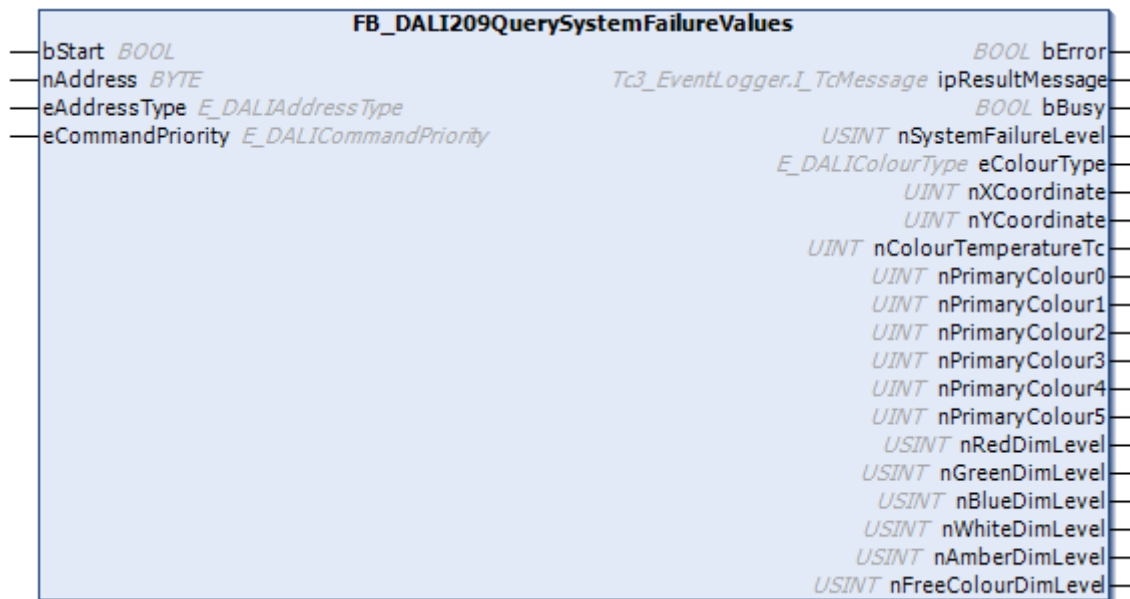
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.2.7.4.9 FB\_DALI209QuerySystemFailureValues



The function block reads the variable `systemFailureLevel` [[▶ 150](#)] and the corresponding color/color temperature settings from the DALI control gear.

**i** In addition to the DALI command QUERY SYSTEM FAILURE LEVEL, the function block sends the DALI commands QUERY COLOUR VALUE and ENABLE DEVICE TYPE 8, as is necessary for application extended commands (see [FB\\_DALI102EnableDeviceType](#) [[▶ 130](#)]).

**i** The function block changes the DTR0 of all connected DALI control gears.

The outputs only contain valid values if the function block was executed without errors (`bError = FALSE`).

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">[► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

## Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nSystemFailureLevel : USINT;
  eColourType     : E_DALIColourType;
  nXCoordinate    : UINT;
  nYCoordinate    : UINT;
  nColourTemperatureTc : UINT;
  nPrimaryColour0 : UINT;
  nPrimaryColour1 : UINT;
  nPrimaryColour2 : UINT;
  nPrimaryColour3 : UINT;
  nPrimaryColour4 : UINT;
  nPrimaryColour5 : UINT;
  nRedDimLevel    : USINT;
  nGreenDimLevel  : USINT;
  nBlueDimLevel   : USINT;
  nWhiteDimLevel  : USINT;
  nAmberDimLevel  : USINT;
  nFreeColourDimLevel : USINT;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nSystemFailureLevel	USINT	Output value in the case of a system failure (0...255).
eColourType	E_DALIColourType	Specifies the color setting with which the DALI control gear is operated (see <a href="#">colourType [► 419]</a> ).
nXCoordinate	UINT	Output of the x-value of the color representation according to xy color type (see <a href="#">X-coordinate [► 418]</a> ) in the color setting Xy coordinates (see <a href="#">color representation [► 418]</a> ).
nYCoordinate	UINT	Output of the y-value of the color representation according to xy color type (see <a href="#">Y-coordinate [► 419]</a> ) in the color setting Xy coordinates (see <a href="#">color representation [► 418]</a> ).
nColourTemperatureTc	UINT	Output of the color temperature (see <a href="#">colourTemperatureTc [► 419]</a> ) in the setting Color temperature Tc (see <a href="#">color representation [► 418]</a> ).
nPrimaryColourN	UINT	Output of the color representation on the corresponding output channel on the operating device in the color setting Primary color N (see <a href="#">color representation [► 418]</a> ).
nRedDimLevel	USINT	Output of the red dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nGreenDimLevel	USINT	Output of the green dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nBlueDimLevel	USINT	Output of the blue dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nWhiteDimLevel	USINT	Output of the white dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nAmberDimLevel	USINT	Output of the amber dimming level (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).
nFreeColourDimLevel	USINT	Output of the dimming level of the freely selectable color (see <a href="#">DimLevel [► 419]</a> ) in the color setting RGBWAF (see <a href="#">color representation [► 418]</a> ).

 **Properties**

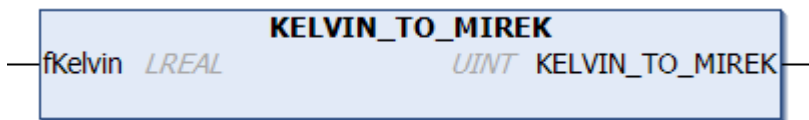
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.2.7.5 Help Functions**

**4.1.2.7.5.1 KELVIN\_TO\_MIREK**



Conversion of the color temperature from Kelvin to Mirek.

Mirek is a color temperature unit used by most DALI commands. The return value of the function is limited to the value range 0...65535 (see table).

Mirek = 1000000 / (color temperature in Kelvin).

Kelvin	Mirek
0.0	65535
15.0	65535
16.0	62500
1000.0	1000
10000.0	100
1000000.0	1
1000001.0	0

 **Inputs**

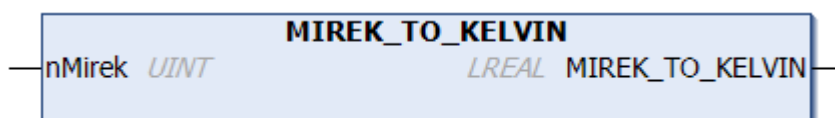
```
VAR_INPUT
  fKelvin    : LREAL;
END_VAR
```

Name	Type	Description
fKelvin	LREAL	Color temperature in Kelvin

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.7.5.2 MIREK\_TO\_KELVIN**



Conversion of the color temperature from Mirek to Kelvin.

Mirek is a color temperature unit used by most DALI commands. The return value of the function is limited to the value range 0...1000001 (see table).

$Kelvin = 1000000 / (\text{color temperature in Mirek})$ .

Mirek	Kelvin
0	1000001.0
1	1000000.0
100	10000.0
1000	1000.0
10000	100.0
65534	15.259
65535	0.0

 **Inputs**

```
VAR_INPUT
  nMirek      : UINT;
END_VAR
```

Name	Type	Description
nMirek	UINT	Color temperature in Mirek

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.7.6 Variables**

Every DALI control gear for color/color temperature control has a certain number of variables (parameters ) from which it is possible to read a variety of information or to modify individual parameters.

Some variables can be read directly by DALI commands (e.g. [FB\\_DALI209QueryColourStatus](#) [[▶ 398](#)]). The function block [FB\\_DALI209QueryColourValue](#) [[▶ 401](#)] can be used to read out further variables.

Name	Reset value	Scope	Size	Necessary color representation [► 418]	Comment
<a href="#">temporaryXCoordinate</a> [► 418]	65535	0...65535	2 bytes	0, 2	
reportXCoordinate	65535	0...65535	2 bytes	0	
<a href="#">xCoordinate</a> [► 418]	No change	0...65535	2 bytes	0	
<a href="#">temporaryYCoordinate</a> [► 419]	65535	0...65535	2 bytes	0, 2	
reportYCoordinate	65535	0...65535	2 bytes	0	
<a href="#">yCoordinate</a> [► 419]	No change	0...65535	2 bytes	0	
<a href="#">temporaryColourTemperatureTc</a> [► 419]	65535	1...65535	2 bytes	1	
reportColourTemperatureTc	65535	1...65535	2 bytes	1	
<a href="#">colourTemperatureTc</a> [► 419]	No change	1...65535	2 bytes	1	
<a href="#">colourTemperatureTcCoolest</a>	<i>colourTemperatureTcPhysicalCoolest</i>	<i>colourTemperatureTcPhysicalCoolest</i> ... <i>colourTemperatureTcWarmest</i> , 65535	2 bytes	1	
<a href="#">colourTemperatureTcWarmest</a>	<i>colourTemperatureTcPhysicalWarmest</i>	<i>colourTemperatureTcCoolest</i> ... <i>colourTemperatureTcPhysicalWarmest</i> , 65535	2 bytes	1	
<a href="#">colourTemperatureTcPhysicalCoolest</a>	No change	1 ... <i>colourTemperatureTcPhysicalWarmest</i> , 65535	2 bytes	1	
<a href="#">colourTemperatureTcPhysicalWarmest</a>	No change	<i>colourTemperatureTcPhysicalCoolest</i> ... 65534, 65535	2 bytes	1	
<a href="#">temporaryPrimaryNDimLevel</a> [► 419]	65535	0...65535	2 bytes	2	
reportPrimaryNDimLevel	65535	0...65535	2 bytes	2	
<a href="#">primaryNDimLevel</a> [► 419]	No change	0...65535	2 bytes	2	
<a href="#">xCoordinatePrimaryN</a> [► 419]	No change	0...65535	2 bytes	0, 2	Read only
<a href="#">yCoordinatePrimaryN</a> [► 419]	No change	0...65535	2 bytes	0, 2	Read only
<a href="#">tyPrimaryN</a> [► 419]	No change	0...65535	2 bytes	0, 2	Read only

Name	Reset value	Scope	Size	Necessary color representation [▶ 418]	Comment
<a href="#">temporaryRedDimLevel [▶ 419]</a>	255	0...255	1 byte	3	
reportRedDimLevel	255	0...255	1 byte	3	
<a href="#">redDimLevel [▶ 419]</a>	No change	0...255	1 byte	3	
<a href="#">temporaryGreenDimLevel [▶ 419]</a>	255	0...255	1 byte	3	
reportGreenDimLevel	255	0...255	1 byte	3	
<a href="#">greenDimLevel [▶ 419]</a>	No change	0...255	1 byte	3	
<a href="#">temporaryBlueDimLevel [▶ 419]</a>	255	0...255	1 byte	3	
reportBlueDimLevel	255	0...255	1 byte	3	
<a href="#">blueDimLevel [▶ 419]</a>	No change	0...255	1 byte	3	
<a href="#">temporaryWhiteDimLevel [▶ 419]</a>	255	0...255	1 byte	3	
reportWhiteDimLevel	255	0...255	1 byte	3	
<a href="#">whiteDimLevel [▶ 419]</a>	No change	0...255	1 byte	3	
<a href="#">temporaryAmberDimLevel [▶ 419]</a>	255	0...255	1 byte	3	
reportAmberDimLevel	255	0...255	1 byte	3	
<a href="#">amberDimLevel [▶ 419]</a>	No change	0...255	1 byte	3	
<a href="#">temporaryFreeColourDimLevel [▶ 419]</a>	255	0...255	1 byte	3	
reportFreeColourDimLevel	255	0...255	1 byte	3	
<a href="#">freeColourDimLevel [▶ 419]</a>	No change	0...255	1 byte	3	
<a href="#">temporaryRGBWAFControl [▶ 420]</a>	255	0...255	1 byte	3	
reportRGBWAFControl	255	0...255	1 byte	3	
<a href="#">RGBWAFControl [▶ 420]</a>	No change	0...255	1 byte	3	
<a href="#">assignedColour [▶ 419]</a>	16#0102_0304_0506	16#0000_0000_0000 ... 16#0606_0606_0606	6 bytes	3	Read only MSB: channel 0 LSB: channel 5

Name	Reset value	Scope	Size	Necessary color representation [▶ 418]	Comment
<a href="#">temporaryColourType</a> [▶ 419]	255	16#10, 16#20, 16#40, 16#80, 16#FF	1 byte	0, 1, 2, 3	
<a href="#">reportColourType</a> [▶ 419]	255	16#10, 16#20, 16#40, 16#80, 16#FF	1 byte	0, 1, 2, 3	
<a href="#">scene0-15ColourType</a> [▶ 419]	65535	16#10, 16#20, 16#40, 16#80, 16#FF	1 byte	0, 1, 2, 3	Read only
<a href="#">scene0-15ColourValue</a>	65535	0...65535	32 bytes... 192 bytes	0, 1, 2, 3	Read only
<a href="#">powerOnColourType</a> [▶ 419]	Manufacturer-dependent	16#10, 16#20, 16#40, 16#80, 16#FF	1 byte	0, 1, 2, 3	Read only
<a href="#">powerOnColourValue</a>	Manufacturer-dependent	0...65535	2 bytes... 12 bytes	0, 1, 2, 3	Read only
<a href="#">systemFailureColourType</a> [▶ 419]	Manufacturer-dependent	16#10, 16#20, 16#40, 16#80, 16#FF	1 byte	0, 1, 2, 3	Read only
<a href="#">systemFailureColourValue</a>	Manufacturer-dependent	0...65535	2 bytes... 12 bytes	0, 1, 2, 3	Read only
<a href="#">gearFeatures/Status</a> [▶ 420]	2#XX00_0001	2#XX00_0000, 2#XX00_0001	1 byte	0, 1, 2, 3	
<a href="#">colourStatus</a> [▶ 420]	No change	0...255	1 byte	0, 1, 2, 3	
<a href="#">colourTypeFeatures</a> [▶ 420]	No change	0...255	1 byte	0, 1, 2, 3	Read only

X: undetermined

In the case of 1-byte values the value 255 is also called MASK.

In the case of 2-byte values the value 65,535 is also called MASK.

### Necessary color representation

Specifies the color representation that the DALI control gear must support so that it contains the appropriate variables:

Value	Description
0	xy coordinates
1	Color temperature Tc
2	Primary (color) N
3	RGBWAF

### xCoordinate / temporaryXCoordinate

Chromaticity coordinate x in the xy color type.

The x-Coordinate is specified in [Mirek](#) [▶ 414].

It can vary from 0 Mirek (0 K) to 65534 Mirek (15.26 K).

At a value of 65535 (MASK), the value is not defined.

**yCoordinate / temporaryYCoordinate**

Chromaticity coordinate y in the xy color type.

The *y-Coordinate* is specified in [Mirek \[▶ 414\]](#).

It can vary from 0 Mirek (0 K) to 65534 Mirek (15.26 K).

At a value of 65535 (MASK), the value is not defined.

**colourTemperatureTc / temporaryColourTemperatureTc**

The variable *colourTemperatureTc / temporaryColourTemperatureTc* defines the color temperature. It is adjustable in the range between *colourTemperatureTcWarmest* and *colourTemperatureTcCoolest*.

The *colourTemperatureTc* is specified in [Mirek \[▶ 414\]](#).

It can vary from 1 Mirek (1000000 K) to 65534 Mirek (15.26 K).

**colourTemperatureTcCoolest / colourTemperatureTcWarmest**

**dimLevel / temporaryDimLevel**

Setting of the dimming level of the corresponding color (red, green, blue, white, amber or a freely selectable color) from 0 to 255.

**primaryNDimLevel / temporary primaryNDimLevel**

**assignedColour**

The association between output channel and color is defined in the variable *assignedColour*. Each byte contains the color of the corresponding channel. The function block [FB\\_DALI209QueryAssignedColour \[▶ 396\]](#) can be used to read out the value.

Value	Description
0	No color assigned
1	Red
2	Green
3	Blue
4	White
5	Amber
6	Freely selectable color

**colourType**

The variable *colourType* defines the color representations supported by the DALI control gear. The values can be read out with the function block [FB\\_DALI209QueryColourValue \[▶ 401\]](#).

<b>E DALIColour-Type [▶ 855]</b>	Description
XyCoordinate	xy coordinates
ColourTemperatureTc	Color temperature Tc
PrimaryNDimLevel	Primary (color) N
RGBWAFControl	RGBWAF
MASK	No color change

**colourStatus**

*colourStatus* contains information about the current status of the DALI control gear.

The function block [FB\\_DALI209QueryColourStatus \[► 398\]](#) can be used to read out the values

Bit	Description
0	xy coordinate color point is outside the valid range.
1	Color temperature Tc lies outside the valid range
2	Automatic calibration is active.
3	Automatic calibration was successful.
4	Color representation xy-coordinate active.
5	Color representation color temperature Tc active.
6	Color representation primary N active.
7	Color representation RGBWAF active.

**gearFeatures / Status**

*gearFeatures / Status* contains information about the current status of the DALI control gear.

The values can be read with the function block [FB\\_DALI209QueryGearFeaturesStatus \[► 402\]](#) and written with the function block [FB\\_DALI209StoreGearFeaturesStatus \[► 367\]](#).

Bit	Description
0	Automatic activation
1...5	reserved
6	Automatic calibration is supported
7	Restoration of the automatic calibration is supported

**RGBWAFControl / temporaryRGBWAFControl**

*RGBWAFControl* contains further information about the assignment between output channel and color.

Bit	Description
0	Output channel 0 / red
1	Output channel 1 / green
2	Output channel 2 / blue
3	Output channel 3 / white
4	Output channel 4 / amber
5	Output channel 5 / free selectable color
6...7	00 = channel control 01 = color control 10 = standardized color control 11 = reserved

**colourTypeFeatures**

*colourTypeFeatures* defines the color representations supported by the DALI control gear.

The values can be read out with the function block [FB\\_DALI209QueryColourTypeFeatures \[► 399\]](#).



Bit	Description
0	The DALI control gear supports color representation through xy coordinates.
1	The DALI control gear supports color representation through color temperature Tc.
2...4	Number of primary colors supported by the DALI control gear. A value of 0 means that this color representation by primary colors is not supported.
5...7	Number of RGBWAF channels supported by the DALI control gear. A value of 0 means that this color representation by RGBWAF is not supported.

### 4.1.2.8 Part 251 (memory bank 1 extension)

#### 4.1.2.8.1 Introduction

In specifications that go beyond the current DALI-2 standard, the [DALI Alliance](#) (DiiA) describes additional DALI parts for the operation of intelligent luminaires.

These new parts are combined under the term D4i.

The current Tc3\_DALI library supports Parts 251 (DT50), 252 (DT51) and 253 (DT52).

These parts contain functions that can be forwarded to the higher-level DALI network for evaluation and control.

They describe the location, type and format of the data in the memory banks of the DALI control gears. In addition, they describe the exchange of data with DALI control devices.

#### DALI Part 251 – memory bank 1 extension

Part 251 extends previously unoccupied areas of memory bank 1 for vendor-specific information. This information does not change the functionality of the control gear.

The data stored here are directly available to the DALI network for technical management, e.g. in large properties.

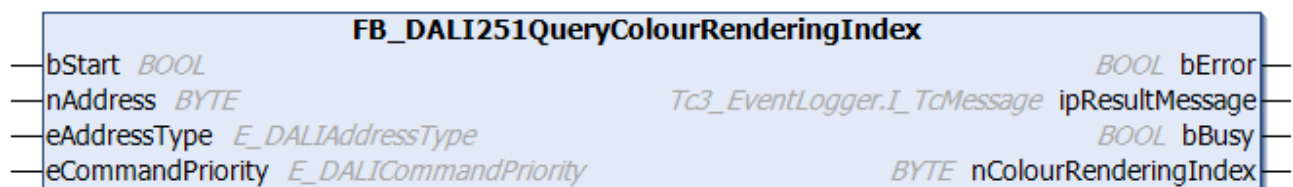
Part 251 provides information about the inventory, such as device number, date of manufacture and color of the device. Information on rated power, rated illuminance, rated voltage, minimum power consumption, color rendering index and correlated color temperature are also available.

The values in Part 251 are read-only.

The entire contents of memory bank 1 are read via the function block [FB\\_DALI102ReadMemoryBank](#) [▶ 47].

Part 251 also provides the function blocks described below for reading the individual parameters.

#### 4.1.2.8.2 FB\_DALI251QueryColourRenderingIndex



The function block reads the color rendering index from the DALI control gear.

The output `nColourRenderingIndex` contains a valid value only if the function block was executed without errors (`bError` = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
```

```
eAddressType      : E_DALIAddressType := E_DALIAddressType.Short;
eCommandPriority  : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 🔌 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nColourRenderingIndex : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nColourRenderingIndex	BYTE	Outputs the color rendering index (0...100).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.8.3 FB\_DALI251QueryCorrelatedColourTemperature



The function block reads the correlated color temperature from the DALI control gear.

The output *nCorrelatedColourTemperature* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nCorrelatedColourTemperature : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nCorrelatedColourTemperature	UINT	Outputs the correlated color temperature [K] (0...17000). If the value 16#FFFE is returned, then Part 209 is not implemented in the DALI control gear.

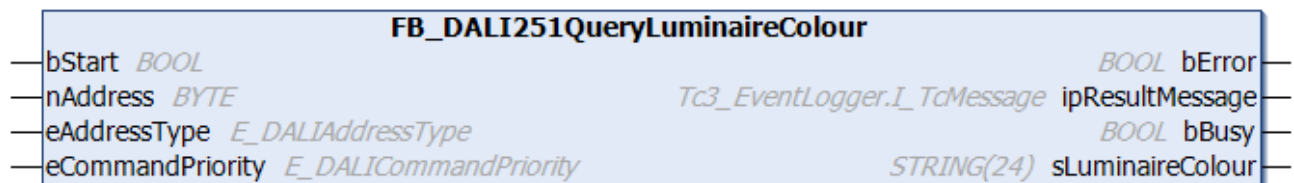
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.8.4 FB\_DALI251QueryLuminaireColour**



The function block reads the luminaire color from the DALI control gear.

The output *sLuminaireColour* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 🔌 Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  sLuminaireColour : STRING(24);
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
sLuminaireColour	STRING	Outputs the luminaire color.



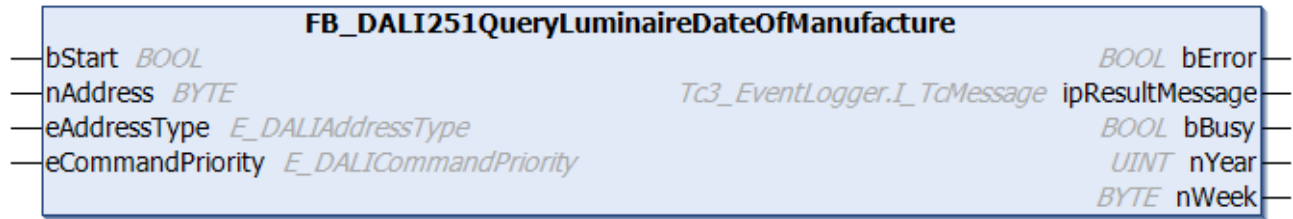
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.8.5 FB\_DALI251QueryLuminaireDateOfManufacture



The function block reads the year and week of production from the DALI control gear.

The outputs *nYear* and *nWeek* only contain a valid value if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nYear      : UINT;
  nWeek      : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nYear	UINT	Year of production (2000...2099).
nWeek	BYTE	Week of production (1...53).

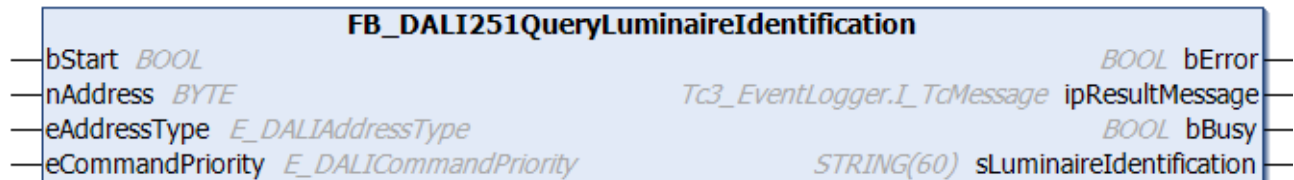
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.8.6 FB\_DALI251QueryLuminaireIdentification**



The function block reads the identification designation of the luminaire from the DALI control gear.

The output *sLuminaireIdentification* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  sLuminaireIdentification : STRING(60);
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
sLuminaireIdentification	STRING	Identification designation



### Properties

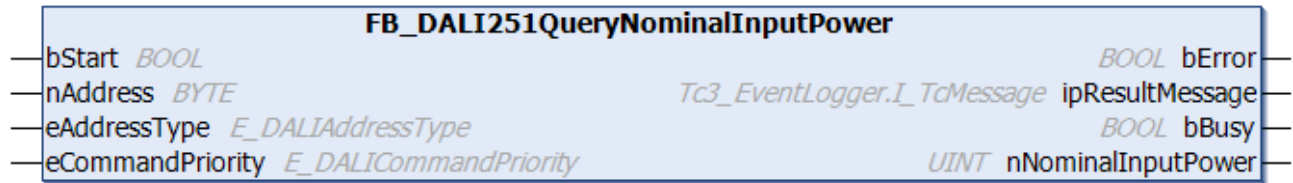
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block ▶ 892</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0



4.1.2.8.7 FB\_DALI251QueryNominalInputPower



The function block reads the nominal input power of the luminaire from the DALI control gear.

The output *nNominalInputPower* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nNominalInputPower : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nNominalInputPower	UINT	Nominal output [W] (0...65534).

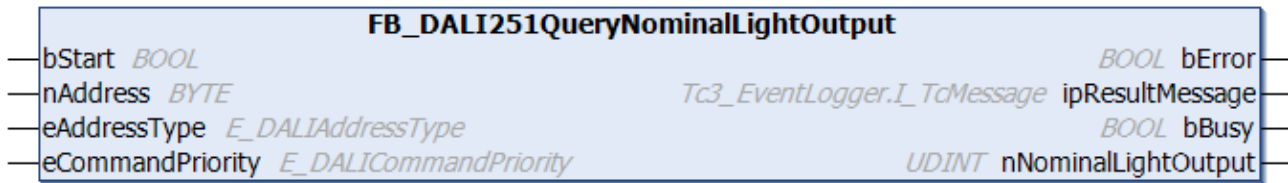
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.8.8 FB\_DALI251QueryNominalLightOutput**



The function block reads the nominal light output of the luminaire from the DALI control gear.

The output *nNominalLightOutput* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType [► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority [► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nNominalLightOutput : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nNominalLightOutput	UDINT	Nominal illuminance [Lm] (0...16777214).

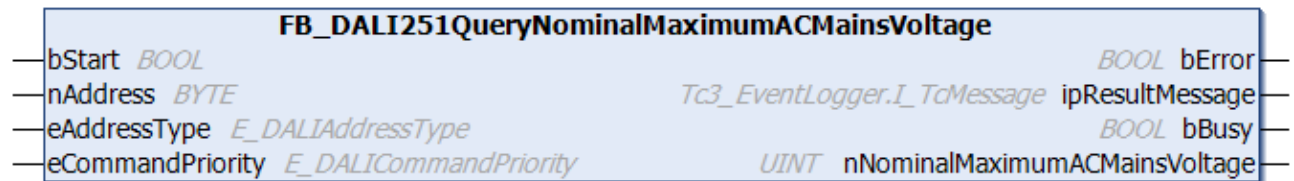
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.8.9 FB\_DALI251QueryNominalMaximumACMainsVoltage**



The function block reads the maximum nominal input voltage of the DALI control gear.

The output *nNominalMaximumACMainsVoltage* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nNominalMaximumACMainsVoltage : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nNominalMaximumACMainsVoltage	UINT	Maximum nominal input voltage [V] (90...480).



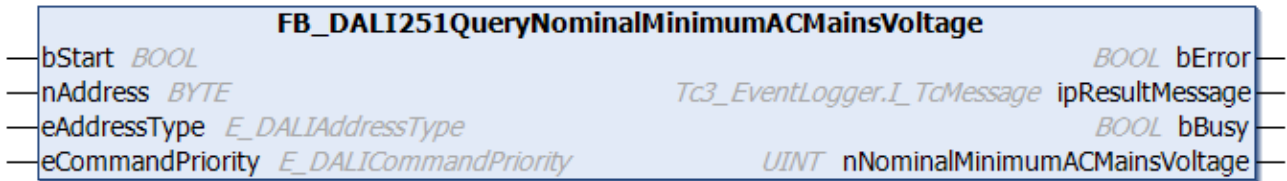
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.8.10 FB\_DALI251QueryNominalMinimumACMainsVoltage



The function block reads the minimum nominal input voltage of the DALI control gear.

The output *nNominalMinimumACMainsVoltage* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nNominalMinimumACMainsVoltage : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nNominalMinimumACMainsVoltage	UINT	Minimum nominal input voltage [V] (90...480).

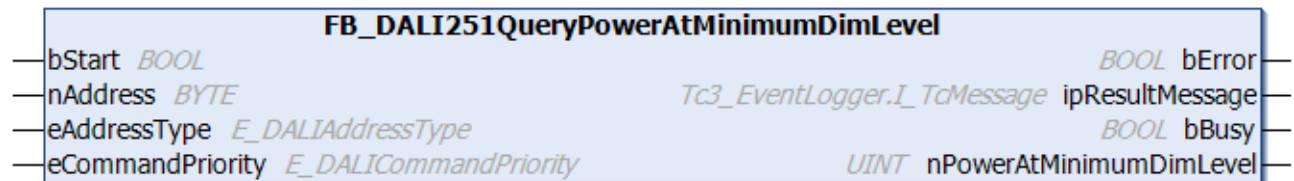
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.8.11 FB\_DALI251QueryPowerAtMinimumDimLevel**



The function block reads the power consumption at the minimum dimming value of the DALI control gear.

The output *nPowerAtMinimumDimLevel* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nPowerAtMinimumDimLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nPowerAtMinimumDimLevel	UINTE	Power consumption [W] at the minimum dimming value (0...65534).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.9 Part 252 (energy reporting)

#### 4.1.2.9.1 Introduction

In specifications that go beyond the current DALI-2 standard, the DALI Alliance (DiiA) describes additional DALI parts for the operation of intelligent luminaires.

These new parts are combined under the term D4i.

The current Tc3\_DALI library supports Parts 251 (DT50), 252 (DT51) and 253 (DT52).

These parts contain functions that can be forwarded to the higher-level DALI network for evaluation and control.

They describe the location, type and format of the data in the memory banks of the DALI control gears. In addition, they describe the exchange of data with DALI control devices.

#### Part 252 – Energy reporting

Part 252 describes additional memory banks in which energy and consumption data can be read.

Memory bank 202 contains information about active energy and active power. This information is obligatory and is provided by DALI control gears that support Part 252.

Memory bank 203 contains information on apparent energy and apparent power.

Memory bank 204 contains information on output energy and output power.

The information from memory banks 203 and 204 is optional and does not need to be provided by DALI control gears that support Part 252.

The values in Part 252 are read-only.

The entire contents of memory banks 202, 203 and 204 are read via the function block FB\_DALI102ReadMemoryBank [► 47].

Part 252 also provides the function blocks described below for reading the individual parameters. The actual value of the read parameter is calculated from the read output value and the scaling factor. The function blocks output both values for further calculation.

#### 4.1.2.9.2 FB\_DALI252QueryActiveEnergy



The function block reads the active energy from the DALI control gear.

It is calculated as follows:

$$E = nActiveEnergy * 10^{nScaleFactorForActiveEnergy}$$

The outputs `nActiveEnergy` and `nScaleFactorForActiveEnergy` only contain a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
```



```
eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">[► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">[► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
bError : BOOL;
ipResultMessage : I_TcMessage;
bBusy : BOOL;
nActiveEnergy : ULINT;
nScaleFactorForActiveEnergy : SINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nActiveEnergy	ULINT	Active energy [Wh] (0...281474976710653).
nScaleFactorForActiveEnergy	SINT	Scaling factor for the active energy (-6...6).

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.9.3 FB\_DALI252QueryActiveEnergyLoadside**



The function block reads the load-side active energy from the DALI control gear.

It is calculated as follows:

$$E_{Loadside} = nActiveEnergyLoadside * 10^{nScaleFactorForLoadsideEnergy}$$

The outputs *nActiveEnergyLoadside* and *nScaleFactorForLoadsideEnergy* only contain a valid value if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nActiveEnergyLoadside : ULINT;
  nScaleFactorForLoadsideEnergy: SINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nActiveEnergyLoads ide	ULINT	Load-side active energy [Wh] (0...281474976710653).
nScaleFactorForLoa dsideEnergy	SINT	Scaling factor for the load-side active energy (-6...6).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunicati on	I_DALICo mmunicati on	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.9.4 FB\_DALI252QueryActivePower**



The function block reads the active power from the DALI control gear.

It is calculated as follows:

$$P = nActivePower * 10^{nScaleFactorForActivePower}$$

The outputs *nActivePower* and *nScaleFactorForActivePower* only contain a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nActivePower    : UDINT;
  nScaleFactorForActivePower : SINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nActivePower	UDINT	Active power [W] (0...4294967293).
nScaleFactorForActivePower	SINT	Scaling factor for the active power (-6...6).

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.9.5 FB\_DALI252QueryActivePowerLoadside



The function block reads the load-side active power from the DALI control gear.

It is calculated as follows:

$$P_{Loadside} = nActivePowerLoadside * 10^{nScaleFactorForLoadsidePower}$$

The outputs *nActivePowerLoadside* and *nScaleFactorForLoadsidePower* only contain a valid value if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
    nActivePowerLoadside : UDINT;
    nScaleFactorForLoadsidePower : SINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nActivePowerLoadside	UDINT	Load-side active power [W] (0...4294967293).
nScaleFactorForLoadsidePower	SINT	Scaling factor for the load-side active power (-6...6).

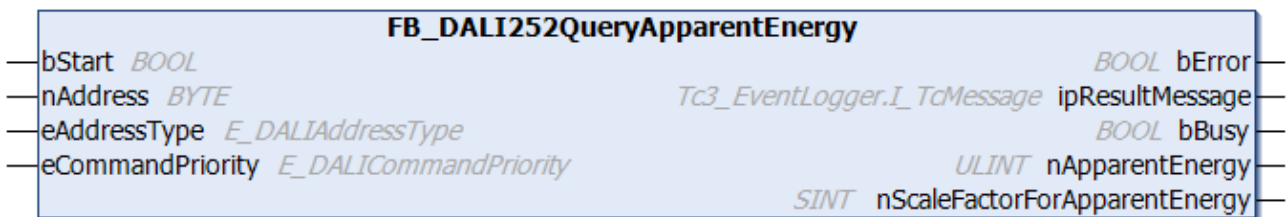
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.9.6 FB\_DALI252QueryApparentEnergy**



The function block reads the apparent energy from the DALI control gear.

It is calculated as follows:

$$ApparentEnergy = nApparentEnergy * 10^{nScaleFactorForApparentEnergy}$$

The outputs *nApparentEnergy* and *nScaleFactorForApparentEnergy* only contain a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nApparentEnergy : ULINT;
  nScaleFactorForApparentEnergy: SINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nApparentEnergy	ULINT	Apparent energy [VAh] (0...281474976710653).
nScaleFactorForApparentEnergy	SINT	Scaling factor for the apparent energy (-6...6).



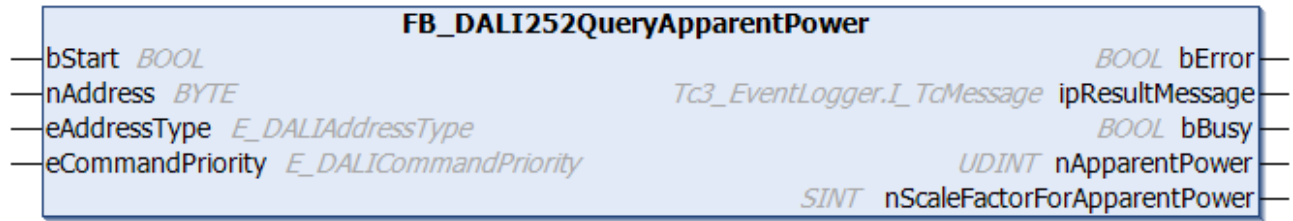
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.9.7 FB\_DALI252QueryApparentPower



The function block reads the apparent power from the DALI control gear.

It is calculated as follows:

$$S = nApparentPower * 10^{nScaleFactorForApparentPower}$$

The outputs `nApparentPower` and `nScaleFactorForApparentPower` only contain a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```
VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<u>E_DALIAddressType</u> ▶ 854	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<u>E_DALICommandPriority</u> ▶ 856	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
    nApparentPower  : UDINT;
    nScaleFactorForApparentPower : SINT;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nApparentPower	ULINT	Apparent power [VA] (0...4294967293).
nScaleFactorForApparentPower	SINT	Scaling factor for the apparent power (-6...6).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10 Part 253 (diagnostics and maintenance)**

**4.1.2.10.1 Introduction**

In specifications that go beyond the current DALI-2 standard, the [DALI Alliance](#) (DiiA) describes additional DALI parts for the operation of intelligent luminaires.

These new parts are combined under the term D4i.

The current Tc3\_DALI library supports Parts 251 (DT50), 252 (DT51) and 253 (DT52).

These parts contain functions that can be forwarded to the higher-level DALI network for evaluation and control.

They describe the location, type and format of the data in the memory banks of the DALI control gears. In addition, they describe the exchange of data with DALI control devices.

**Part 253 – Diagnostics and maintenance**

Part 252 writes additional memory banks in which diagnostic and maintenance data can be read and partly specified.

Memory bank 205 contains information on diagnostics and maintenance of the DALI control gear.

Memory bank 206 contains information on the diagnostics and maintenance of the light source.

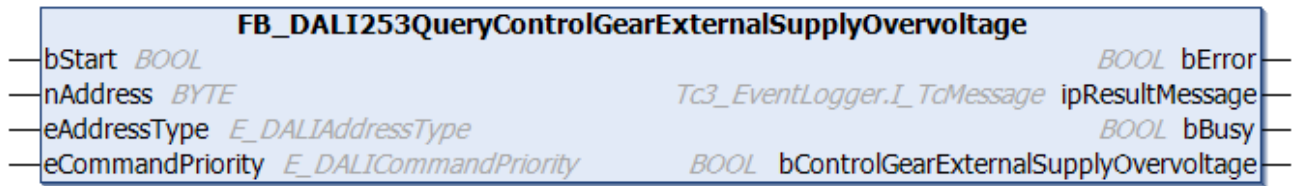
Memory bank 207 contains maintenance information on the luminaire.

This information is obligatory and is provided by DALI control gears that support Part 253.

The entire contents of memory banks 205, 206 and 207 are read via the function block `FB_DALI102ReadMemoryBank` [▶ 47].

Part 253 also provides the function blocks described below for reading the individual parameters.

### 4.1.2.10.2 FB\_DALI253QueryControlGearExternalSupplyOvervoltage



The function block indicates whether an overvoltage has been detected in the external supply voltage of the DALI control gear.

The output `bControlGearExternalSupplyOvervoltage` only contains a valid value if the function block was executed without errors (`bError = FALSE`).

#### Inputs

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

**i** `eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```

VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  bControlGearExternalSupplyOvervoltage : BOOL;
END_VAR
  
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bControlGearExternalSupplyOvervoltage	BOOL	The output is set as soon as an overvoltage has been detected in the external supply voltage of the DALI control gear.

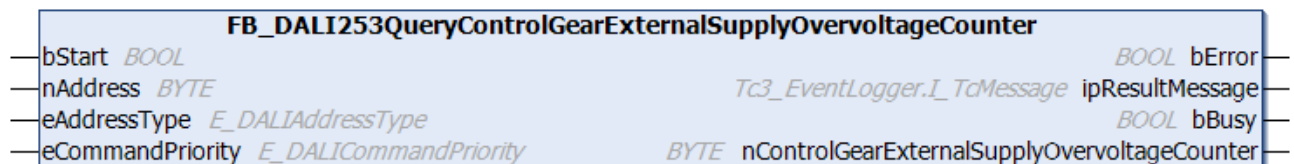
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.3 FB\_DALI253QueryControlGearExternalSupplyOvervoltageCounter**



The function block reads the counter for the detected overvoltage events of the external supply voltage of the DALI control gear.

The output *nControlGearExternalSupplyOvervoltageCounter* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nControlGearExternalSupplyOvervoltageCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearExternalSupplyOvervoltageCounter	BYTE	Number of detected overvoltage events of the external supply voltage (0...253).

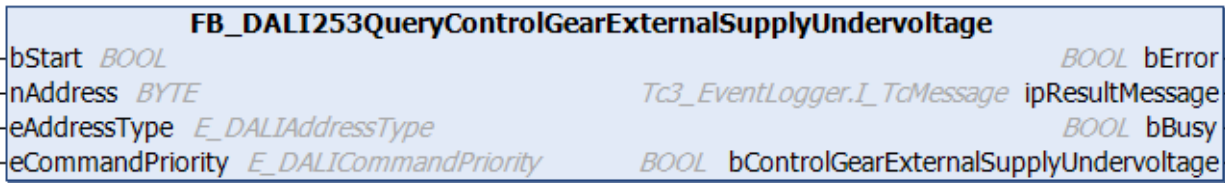
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.4 FB\_DALI253QueryControlGearExternalSupplyUndervoltage



The function block indicates whether an undervoltage has been detected in the external supply voltage of the DALI control gear.

The output *bControlGearExternalSupplyUndervoltage* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  bControlGearExternalSupplyUndervoltage : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bControlGearExternalSupplyOvervoltage	BOOL	The output is set as soon as an overvoltage has been detected in the external supply voltage of the DALI control gear.

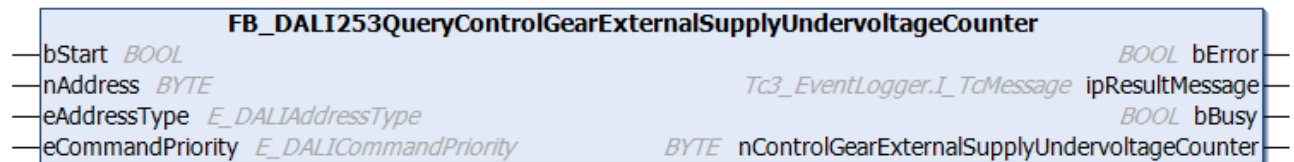
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.5 FB\_DALI253QueryControlGearExternalSupplyUndervoltageCounter**



The function block reads the counter for the detected undervoltage events of the external supply voltage of the DALI control gear.

The output *nControlGearExternalSupplyUndervoltageCounter* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nControlGearExternalSupplyUndervoltageCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearExternalSupplyUndervoltageCounter	BYTE	Number of detected undervoltage events of the external supply voltage (0...253).

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.6 FB\_DALI253QueryControlGearExternalSupplyVoltage



The function block reads the effective value of the external supply voltage from the DALI control gear.

The output *fControlGearExternalSupplyVoltage* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  fControlGearExternalSupplyVoltage : LREAL;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
fControlGearExternalSupplyVoltage	LREAL	RMS value of the external supply voltage [Vrms] (0.0... 6553.3).

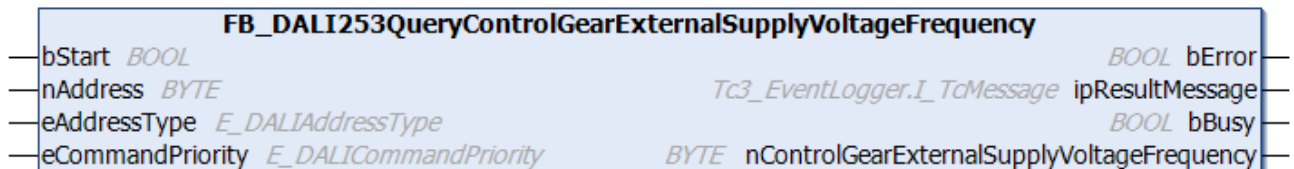
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.7 FB\_DALI253QueryControlGearExternalSupplyVoltageFrequency**



The function block reads the frequency of the external supply voltage from the DALI control gear.

The output *fControlGearExternalSupplyVoltageFrequency* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  fControlGearExternalSupplyVoltageFrequency : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
fControlGearExternalSupplyVoltageFrequency	BYTE	Frequency of the external supply voltage [Hz] (0...253).

### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.8 FB\_DALI253QueryControlGearOperatingTime



The function block reads the operating time of the DALI control gear.

The output *nControlGearOperatingTime* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
  nControlGearOperatingTime : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearOperatingTime	UDINT	Operating time [s] (0...4294967293)

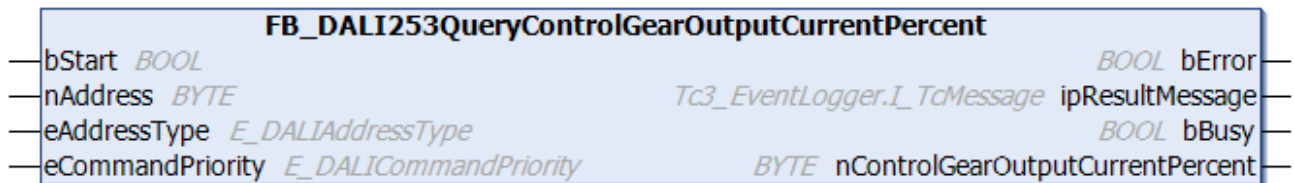
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.9 FB\_DALI253QueryControlGearOutputCurrentPercent**



The function block reads the output current as a percentage of the nominal current from the DALI control gear.

The output *nControlGearOutputCurrentPercent* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nControlGearOutputCurrentPercent : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearOutputCurrentPercent	BYTE	Percentage output current [%] (0...100)



**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.10 FB\_DALI253QueryControlGearOutputPowerLimitation



The function block indicates whether the limit value of the output power of the DALI control gear was exceeded.

The output *bControlGearOutputPowerLimitation* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  bControlGearOutputPowerLimitation : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bControlGearOutputPowerLimitation	BOOL	The output is set as soon as the output power of the DALI control gear is higher than the set limit value.

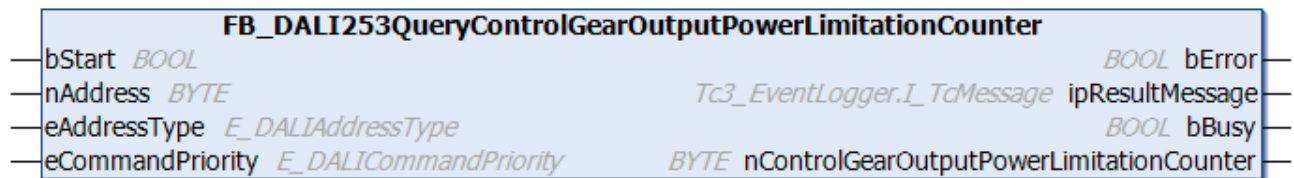
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.11 FB\_DALI253QueryControlGearOutputPowerLimitationCounter**



The function block reads the counter of the number of exceedances of the output power limit value at the DALI control gear.

The output *nControlGearOutputPowerLimitationCounter* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nControlGearOutputPowerLimitationCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearOutputPowerLimitationCounter	BYTE	Number of times the output power was exceeded (0...253).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0



4.1.2.10.12 FB\_DALI253QueryControlGearOverallFailureCondition



The function block outputs the status of `controlGearFailure` [▶ 153] at the DALI control gear.

The output `bControlGearOverallFailureCondition` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  bControlGearOverallFailureCondition : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
ControlGearOverallFailureCondition	BOOL	The output is set as soon as an error has been detected in the DALI control gear.

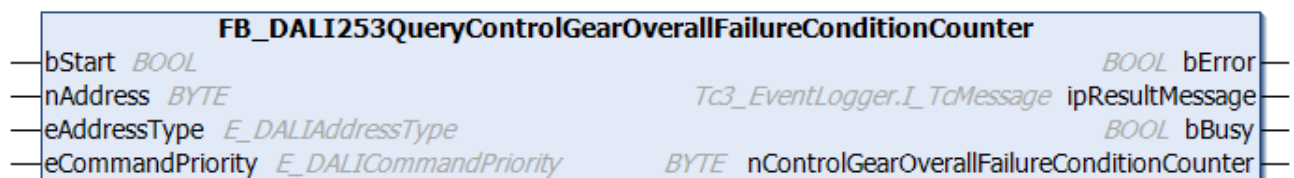
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.13 FB\_DALI253QueryControlGearOverallFailureConditionCounter**



The function block reads the counter of the error events that have occurred in the DALI control gear.

The output *nControlGearOverallFailureConditionCounter* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nControlGearOverallFailureConditionCounter: BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearOverallFailureConditionCounter	BYTE	Number of errors detected (0...253).



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.14 FB\_DALI253QueryControlGearPowerFactor



The function block outputs the power factor of the DALI control gear.

The output *fControlGearPowerFactor* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType ▶ 854	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority ▶ 856	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  fControlGearPowerFactor : LREAL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
fControlGearPowerFactor	LREAL	Power factor (0.0...1.0)

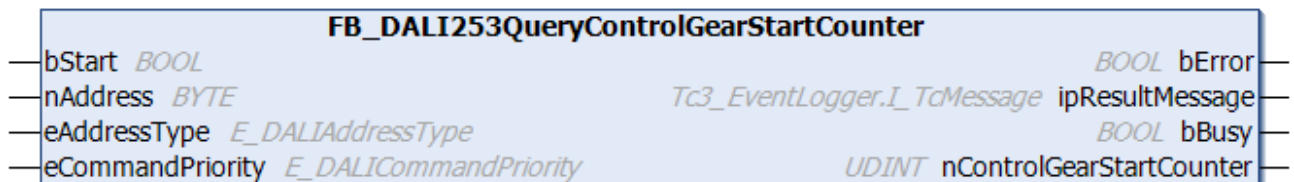
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.15 FB\_DALI253QueryControlGearStartCounter**



The function block outputs the number of starts of the DALI control gear. The value is incremented after switching on the external power supply.

The output *nControlGearStartCounter* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nControlGearStartCounter : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearStartCounter	UDINT	Number of starts of the DALI control gear (0...16777213).



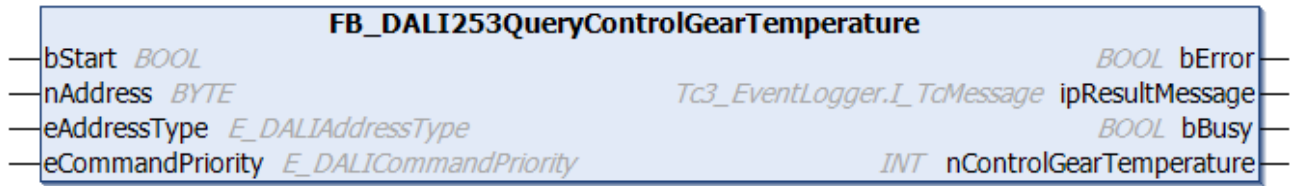
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.16 FB\_DALI253QueryControlGearTemperature



The function block outputs the internal temperature of the DALI control gear.

The output *nControlGearTemperature* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType ▶ 854	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority ▶ 856	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nControlGearTemperature : INT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearTemperature	INT	Internal temperature [°C] (-60... 193)

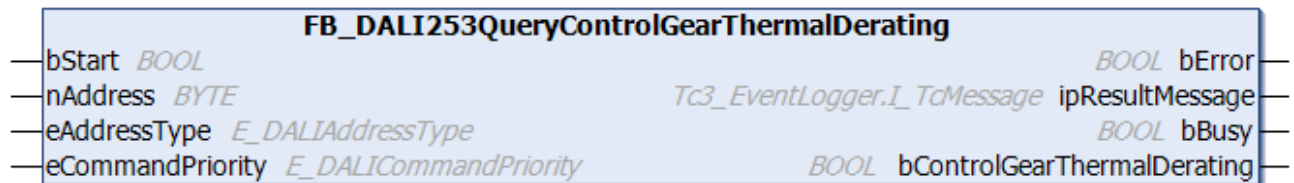
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.17 FB\_DALI253QueryControlGearThermalDerating**



The function block indicates whether the internal temperature is higher than the specified limit value in the DALI control gear.

The output *bControlGearThermalDerating* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 🔌 Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  bControlGearThermalDerating : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bControlGearThermalDerating	BOOL	The output is set as soon as the internally measured temperature of the DALI control gear is higher than the specified limit value.

### 📄 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.18 FB\_DALI253QueryControlGearThermalDeratingCounter



The function block reads the counter that counts the number of times the temperature limit value inside the DALI control gear is exceeded.

The output *nControlGearThermalDeratingCounter* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nControlGearThermalDeratingCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearThermalDeratingCounter	BYTE	Number of times the internal temperature was exceeded (0...253)

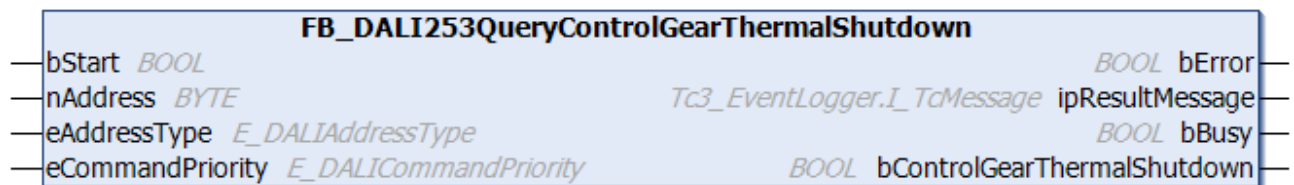
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.19 FB\_DALI253QueryControlGearThermalShutdown**



The function block indicates whether there is a thermal shutdown of the DALI control gear.

The output *bControlGearThermalShutdown* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  bControlGearThermalShutdown : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bControlGearThermalShutdown	BOOL	The output is set as soon as the condition for a thermal shutdown of the DALI control gear is satisfied.



**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.20 FB\_DALI253QueryControlGearThermalShutdownCounter



The function block reads the counter of the number of thermal shutdowns of the DALI control gear.

The output *nControlGearThermalShutdownCounter* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType ▶ 854	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority ▶ 856	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nControlGearThermalShutdownCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nControlGearThermalShutdownCounter	BYTE	Number of thermal shutdowns (0...253)

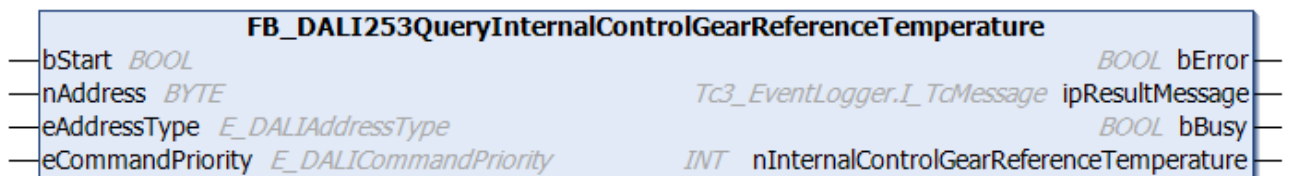
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.21 FB\_DALI253QueryInternalControlGearReferenceTemperature**



The function block outputs the internal reference temperature of the DALI control gear.

The output *nInternalControlGearReferenceTemperature* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nInternalControlGearReferenceTemperature: INT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nInternalControlGearReferenceTemperature	INT	Internal reference temperature [°C] (-60...193)

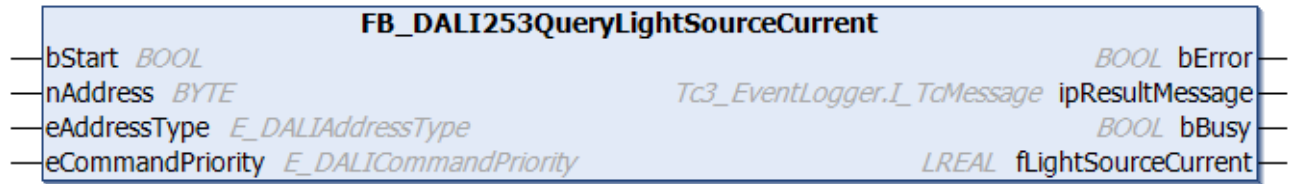
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.22 FB\_DALI253QueryLightSourceCurrent



The function block reads the current consumption of the light source.

The output *fLightSourceCurrent* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  fLightSourceCurrent : LREAL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
fLightSourceCurrent	LREAL	Current consumption [A] (0.0...65.533)



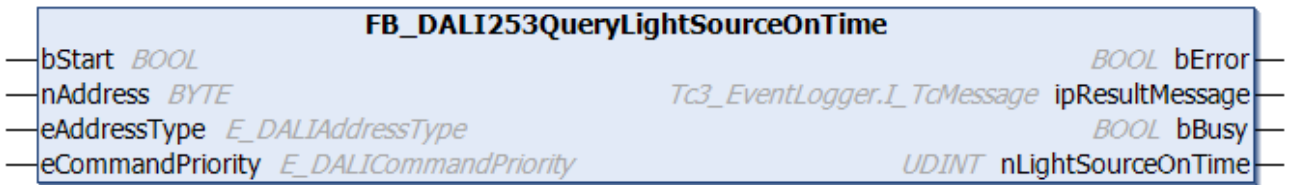
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.23 FB\_DALI253QueryLightSourceOnTime



The function block reads the non-resettable operating hour counter of the light source. If the light source is switched on, the non-resettable operating hour counter is incremented by one every second.

The non-resettable operating hour counter cannot be reset or changed.

The output *nLightSourceOnTime* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[► 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[► 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

 Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
```

```
bBusy : BOOL;
nLightSourceOnTime : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceOnTime	UDINT	Current value of the non-resettable operating hours counter [s] (0...4294967293)

 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.24 FB\_DALI253QueryLightSourceOnTimeResettable



The function block reads the resettable operating hour counter of the light source. If the light source is switched on, the resettable operating hour counter is incremented by one every second. The resettable operating hours counter can be changed by the function block [FB\\_DALI253SetLightSourceOnTimeResettable \[▶ 505\]](#).

The output *nLightSourceOnTimeResettable* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 Inputs

```
VAR_INPUT
bStart : BOOL;
nAddress : BYTE;
eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nLightSourceOnTimeResettable : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceOnTimeResettable	UDINT	Current value of the resettable operating hours counter of the light source [s] (0...4294967293)



**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.25 FB\_DALI253QueryLightSourceOpenCircuit



The function block indicates whether there is an open circuit to the light source.

The output *bLightSourceOpenCircuit* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  bLightSourceOpenCircuit : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bLightSourceOpenCircuit	BOOL	The output is set as soon as an open circuit to the light source is detected.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.26 FB\_DALI253QueryLightSourceOpenCircuitCounter**



The function block reads the counter of the number of open circuits to the light source.

The output *nLightSourceOpenCircuitCounter* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nLightSourceOpenCircuitCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceOpenCircuitCounter	BYTE	Number of open circuits (0...253)



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.27 FB\_DALI253QueryLightSourceOverallFailureCondition



The function block outputs the status of `lampFailure` [▶ 153] at the DALI control gear.

The output `bLightSourceOverallFailureCondition` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  bLightSourceOverallFailureCondition : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bLightSourceOverallFailureCondition	BOOL	The output is set as soon as a light source error has been detected in the DALI control gear.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.28 FB\_DALI253QueryLightSourceOverallFailureConditionCounter**



The function block reads the counter of the light source error events.

The output *nLightSourceOverallFailureConditionCounter* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 🔌 Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nLightSourceOverallFailureConditionCounter: BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceOverallFailureConditionCounter	BYTE	Number of errors detected (0...253)



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.29 FB\_DALI253QueryLightSourceShortCircuit



The function block indicates whether there is a short-circuit at the light source.

The output *bLightSourceShortCircuit* contains a valid value only if the function block was executed without errors (*bError* = FALSE)

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  bLightSourceShortCircuit : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bLightSourceShortCircuit	UDINT	The output is set as soon as a short circuit is detected at the light source.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.30 FB\_DALI253QueryLightSourceShortCircuitCounter**



The function block reads the counter that counts the number of light source short-circuits.

The output *nLightSourceShortCircuitCounter* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nLightSourceShortCircuitCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceShortCircuitCounter	BYTE	Number of short-circuits that have occurred (0...253)



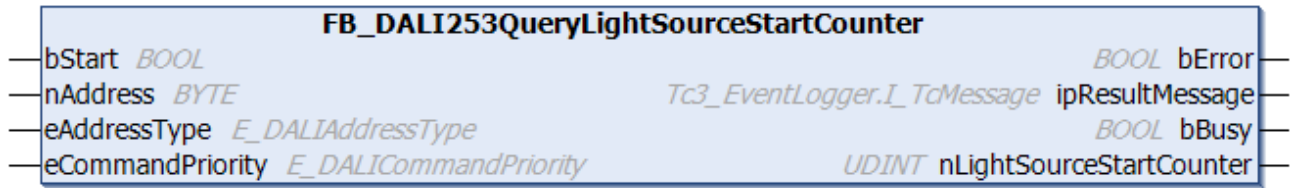
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.31 FB\_DALI253QueryLightSourceStartCounter



The function block reads the non-resettable counter for the switching on of the light source. The non-resettable counter is incremented by one each time the light source changes from the Off state to the On state. The non-resettable counter cannot be reset or changed.

The output *nLightSourceStartCounter* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

Inputs

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.

*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nLightSourceStartCounter : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceStartCounter	BOOL	Current value of the non-resettable counter for the switching on of the light source (0...16777213).

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.32 FB\_DALI253QueryLightSourceStartCounterResettable**



The function block reads the resettable counter for the switching on of the light source. The resettable counter is incremented by one each time the light source changes from the Off state to the On state.

The resettable counter is writable by the function block [FB\\_DALI253SetLightSourceStartCounterResettable](#) [▶ 507].

The output *nLightSourceStartCounterResettable* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nLightSourceStartCounterResettable : UDINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceStartCounterResettable	UDINT	Current value of the resettable counter for the switching on of the light source (0...16777213).

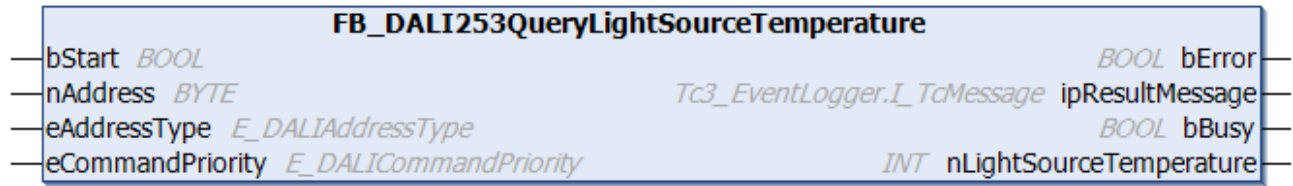
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.33 FB\_DALI253QueryLightSourceTemperature



The function block indicates the temperature of the light source.

The output *nLightSourceTemperature* only contains a valid value if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
    nLightSourceTemperature : INT;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceTemperature	INT	Temperature of the light source [°C] (-60...193)

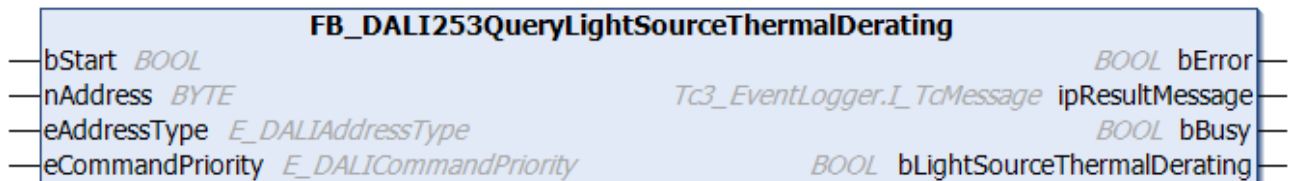
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.34 FB\_DALI253QueryLightSourceThermalDerating**



The function block indicates whether the light source temperature is higher than the specified limit value.

The output *bLightSourceThermalDerating* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  bLightSourceThermalDerating : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bLightSourceThermalDerating	BOOL	The output is set as soon as the temperature measured at the light source is higher than the specified limit value.



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block ▶ 892</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.35 FB\_DALI253QueryLightSourceThermalDeratingCounter



The function block reads the counter that counts the number of times the light source temperature exceeded the limit value.

The output *nLightSourceThermalDeratingCounter* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType ▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority ▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy      : BOOL;
  nLightSourceThermalDeratingCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceThermalDeratingCounter	BYTE	Number of times the light source temperature was exceeded (0...253)

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.36 FB\_DALI253QueryLightSourceThermalShutdown**



The function block indicates whether there is a thermal shutdown on the DALI control gear due to the light source.

The output *bLightSourceThermalShutdown* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  bLightSourceThermalShutdown : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <u>error evaluation</u> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <u>runtime messages</u> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bLightSourceThermalShutdown	BOOL	The output is set as soon as the condition for a thermal shutdown of the light source is satisfied.

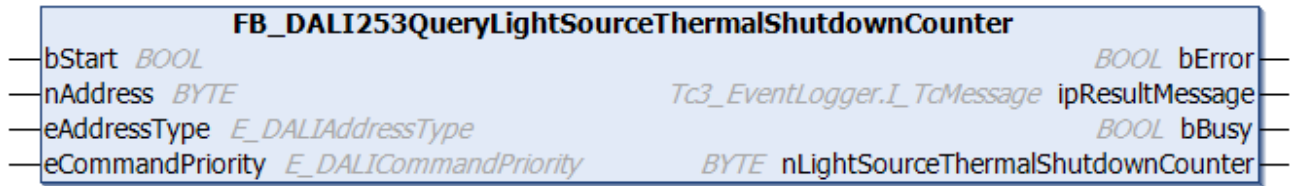
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.37 FB\_DALI253QueryLightSourceThermalShutdownCounter



The function block reads the counter that counts the number of thermal shutdowns of the light source.

The output *nLightSourceThermalShutdownCounter* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nLightSourceThermalShutdownCounter : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nLightSourceThermalShutdownCounter	BYTE	Number of thermal shutdowns (0...253)

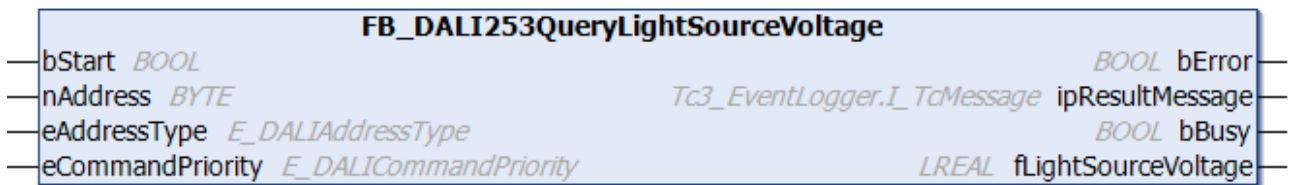
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.38 FB\_DALI253QueryLightSourceVoltage**



The function block reads the output voltage at the DALI control gear.

The output *fLightSourceVoltage* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
  
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  fLightSourceVoltage   : LREAL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
fLightSourceVoltage	LREAL	Output voltage [V] (0.0...6553.3)

### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0



4.1.2.10.39 FB\_DALI253QueryRatedMedianUsefulLifeOfLuminaire



The function block indicates the estimated average lifetime of the complete luminaire, including all individual components.

The function block `FB_DALI253SetRatedMedianUsefulLifeOfLuminaire` [► 508] writes the value to the DALI control gear.

The output `nRatedMedianUsefulLifeOfLuminaire` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...15). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



`eCommandPriority` has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nRatedMedianUsefulLifeOfLuminaire : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRatedMedianUsefulLifeOfLuminaire	BYTE	Estimated average service life [1000 h]. The value lies within the range 0...253 as a multiple of the step size of 1000 h.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.40 FB\_DALI253QueryRatedMedianUsefulLightSourceStarts**



The function block indicates the average number of possible starts of the complete luminaire, including all individual components.

The function block [FB\\_DALI253SetRatedMedianUsefulLightSourceStarts \[► 510\]](#) writes the value to the DALI control gear.

The output *nRatedMedianUsefulLightSourceStarts* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

### 🔌 Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nRatedMedianUsefulLightSourceStarts : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRatedMedianUsefulLightSourceStarts	UINT	Number of average possible starts [100]. The value lies within the range 0...65533 as a multiple of the step size of 100.



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.41 FB\_DALI253SetInternalControlGearReferenceTemperature



The function block writes the internal reference temperature to the DALI control gear.

The function block [FB\\_DALI253QueryInternalControlGearReferenceTemperature](#) [▶ 474] reads the value from the DALI control gear.

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
w;
  nInternalControlGearReferenceTemperature : INT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nInternalControlGearReferenceTemperature	INT	Internal reference temperature [°C] (-60...193)

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.42 FB\_DALI253SetLightSourceOnTimeResettable**



The function block sets the resettable operating hour counter of the light source to any desired value. If the light source is switched on, the resettable operating hour counter is incremented by one every second.

The resettable operating hour counter can be read using the function block [FB\\_DALI253QueryLightSourceOnTimeResettable \[▶ 478\]](#).

 **Inputs**

```

VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
    nLightSourceOnTimeResettable : UDINT;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nLightSourceOnTimeResettable	UDINT	New value of the resettable operating hours counter [s] (0...4294967293)

### Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.10.43 FB\_DALI253SetLightSourceStartCounterResettable



The function block sets the resettable counter for the switching on of the light source to any desired value. The resettable counter is incremented by one each time the light source changes from the Off state to the On state.

The resettable counter can be read using the function block FB\_DALI253QueryLightSourceStartCounterResettable [► 490].

Inputs

```
VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
    nLightSourceStartCounterResettable : UDINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nLightSourceStartCounterResettable	UDINT	New value of the resettable counter for the switching on of the light source (0...16777213)

Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

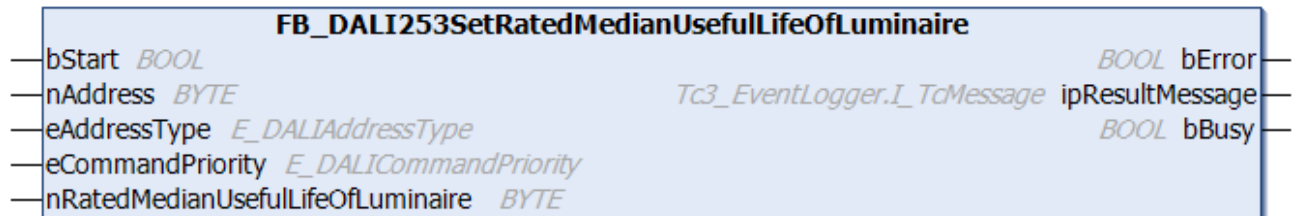
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.10.44 FB\_DALI253SetRatedMedianUsefulLifeOfLuminaire**



The function block writes the estimated average lifetime of the complete luminaire to the DALI control gear.

The function block [FB\\_DALI253QueryRatedMedianUsefulLifeOfLuminaire \[► 501\]](#) reads the value from the DALI control gear.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nRatedMedianUsefulLifeOfLuminaire : BYTE;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nRatedMedianUsefulLifeOfLuminaire	BYTE	Estimated average service life [1000 h]. The input value is between 0 and 253, as a multiple of the step size of 1000 h.

### 🔌 Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.



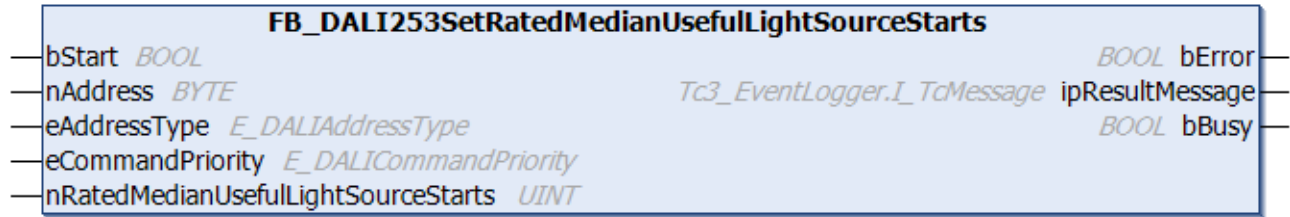
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.1.2.10.45 FB\_DALI253SetRatedMedianUsefulLightSourceStarts



The function block writes the average number of possible starts of the complete luminaire to the DALI control gear.

The function block `FB_DALI253QueryRatedMedianUsefulLightSourceStarts` [▶ 502] reads the value from the DALI control gear.

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nRatedMedianUsefulLightSourceStarts : UINT;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...15). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



*eCommandPriority* has no function with the KL6811. The DALI command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nRatedMedianUsefulLightSourceStarts	UINT	Number of average possible starts. The input value is between 0 and 65533, as a multiple of the step size of 100.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.11 Part 301 (push button)**

**4.1.2.11.1 Introduction**

IEC 62386-301 describes DALI control devices that are defined as push buttons.

DALI control devices according to IEC 62386-301 correspond to instance type 1. The type of an instance can be queried with the function block [FB\\_DALI103QueryInstanceType \[▶ 201\]](#).

**Input value**

The current state (*inputValue*) of the push button can be queried via the function block [FB\\_DALI103QueryInputValue \[▶ 194\]](#). The value is 16#FF when the push button is actuated. The value is 16#00 when it is not actuated.

**Timer**

IEC 62386-301 defines specific timers with which the sending of the different events can be influenced.

The respective time is calculated from

$$Time = T_{incr} \times Multiplier$$

Time	Multiplier	Preset value of the multiplier	T <sub>incr</sub> incremental value	T <sub>default</sub> default value	T <sub>min</sub> lowest possible set value	T <sub>max</sub> highest possible set value
T <sub>short</sub>	<i>tShort</i>	Max (25, <i>tShortMin</i> )	20 ms	Max (25, <i>tShortMin</i> ) * 20 ms	<i>tShortMin</i> * 20 ms	5100 ms
T <sub>double</sub>	<i>tDouble</i>	0	20 ms	<i>tDoubleMin</i> * 20 ms	<i>tDoubleMin</i> * 20 ms	2000 ms
T <sub>repeat</sub>	<i>tRepeat</i>	8	20 ms	160 ms	100 ms	2000 ms
T <sub>stuck</sub>	<i>tStuck</i>	20	1 s	20 s	5 s	255 s

*tShortMin*: Vendor-specific (see [tShortMin \[► 531\]](#)).

*tDoubleMin*: Vendor-specific (see [tDoubleMin \[► 532\]](#)).



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.

### Event filter

DALI control devices are capable of sending different events. The event filter consists of an 8-bit value, where each individual bit defines whether an event is disabled or enabled. The event is enabled if the corresponding bit is set, otherwise it is disabled.

The function block [FB\\_DALI103QueryEventFilter \[► 188\]](#) can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter \[► 170\]](#) can be used to write to it.

The event filter (see [eventFilter \[► 531\]](#)) must be configured accordingly in order to trigger time-controlled events.

Bit	Event	Description
0	<i>Button Released</i>	The push button was released.
1	<i>Button Pressed</i>	The push button was pressed.
2	<i>Short Press</i>	The push button was pressed and quickly released again without being pressed again (short button press).
3	<i>Double Press</i>	The push button was pressed, released and quickly pressed again (double button press).
4	<i>Long Press Start</i>	The push button was pressed without being released in between times (long button press).
5	<i>Long Press Repeat</i>	If the push button was pressed for a long period ( <i>Long Press Start</i> ) without being released in between times, this event is sent at regular intervals.
6	<i>Long Press Stop</i>	This event is sent if the push button is released after a long button press.
7	<i>Button Free / Button Stuck</i>	If the push button is pressed for a very long time, it is assumed that it has stuck and the <i>Button Stuck</i> event is sent. <i>Button Free</i> is sent when the push button is detected as not pressed again.



Release only those events that you need for the application. If too many events are sent, this can have a negative effect on the behavior of the application.

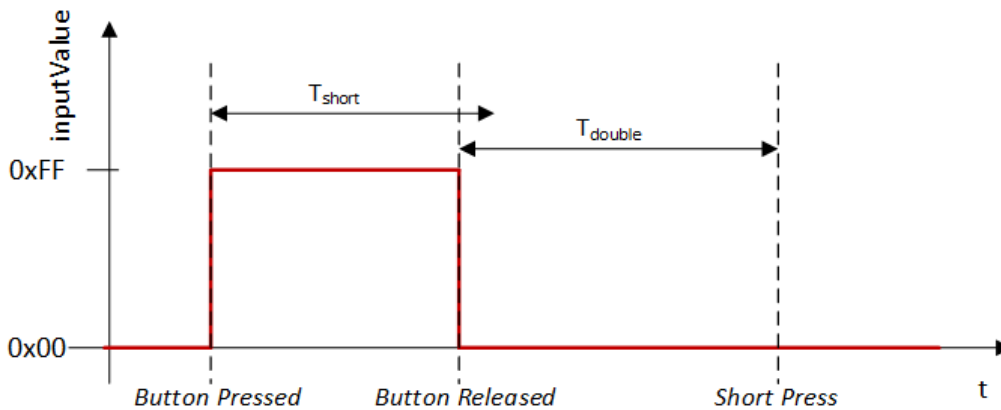
### Input Notification event

With instance type 1 it is possible to release different events with the help of the event filter. All events are sent from the DALI control device by a *Input Notification* and can be received with [FB\\_DALIGetInputNotification \[► 841\]](#).

The possible events (*Event Name* column) and their binary representation (*Event Information* column) are listed in the following table.

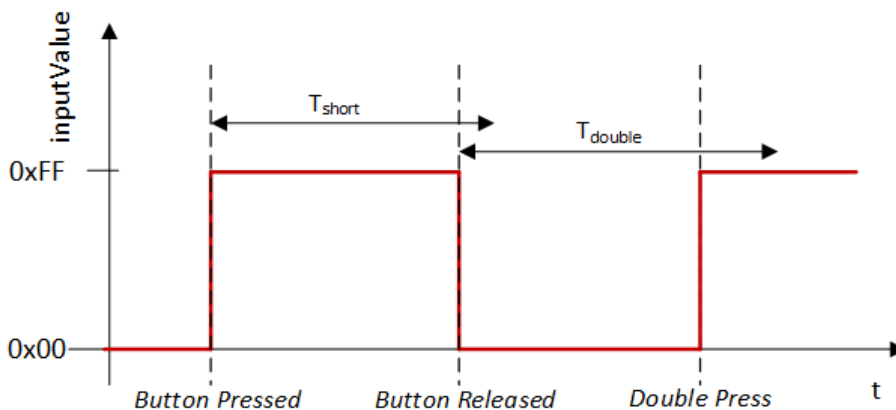
Event Name	Event Information	Description
<i>Button Released</i>	2#00_0000_0000	The push button was released.
<i>Button Pressed</i>	2#00_0000_0001	The push button was pressed.
<i>Short Press</i>	2#00_0000_0010	The push button was pressed and quickly released again without being pressed again (short button press).
<i>Double Press</i>	2#00_0000_0101	The push button was pressed, released and quickly pressed again (double button press).
<i>Long Press Start</i>	2#00_0000_1001	The push button was pressed without being released in between times.
<i>Long Press Repeat</i>	2#00_0000_1011	If the push button was pressed for a long period ( <i>Long Press Start</i> ) without being released in between times, this event is sent at regular intervals.
<i>Long Press Stop</i>	2#00_0000_1100	This event is sent if the push button is released after a long button press.
<i>Button Free</i>	2#00_0000_1110	Button Free is sent when the push button is detected as not pressed again following the <i>Button Stuck</i> event.
<i>Button Stuck</i>	2#00_0000_1111	If the push button is pressed for a very long time, it is assumed that it has stuck and the <i>Button Stuck</i> event is sent.

**Short Press event**



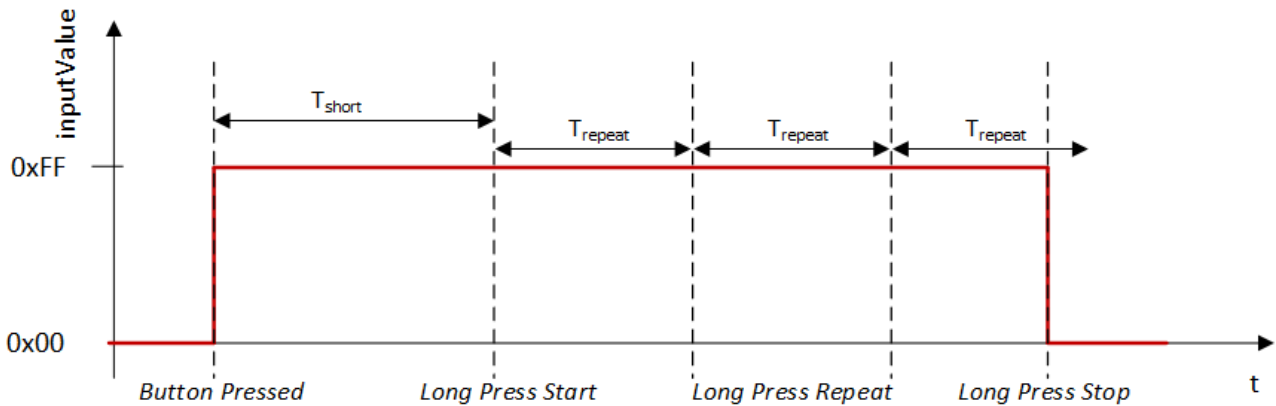
The *Short Press* event is triggered when the push button is released again within the time  $T_{short}$  and not pressed again for at least the time  $T_{double}$ . If the push button is pressed again within the time  $T_{double}$ , a new *Short Press* event can be triggered.

**Double Press event**



The *Double Press* event is triggered when the push button is released within the time  $T_{short}$  and pressed again before expiry of the time  $T_{double}$ .

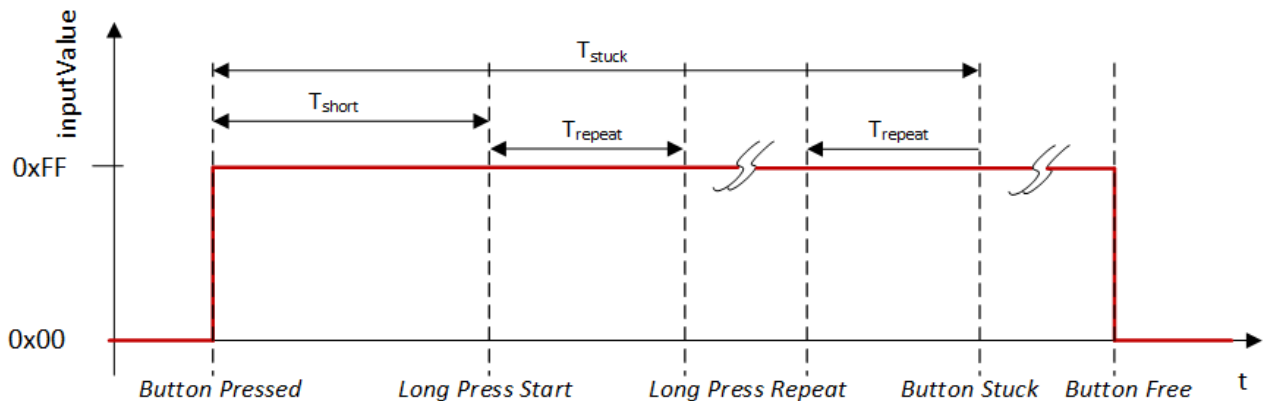
**Long Press Start, Long Press Repeat and Long Press Stop events**



The *Long Press* event is triggered when the push button still remains pressed after expiry of the time  $T_{short}$ . The time  $T_{repeat}$  subsequently starts. The *Long Press Repeat* event is triggered if this time has expired. If the push button remains pressed, the time  $T_{repeat}$  starts again and the *Long Press Repeat* event is triggered after each expiry.

The *Long Press Stop* event is triggered by releasing the push button.

**Button Free and Button Stuck events**

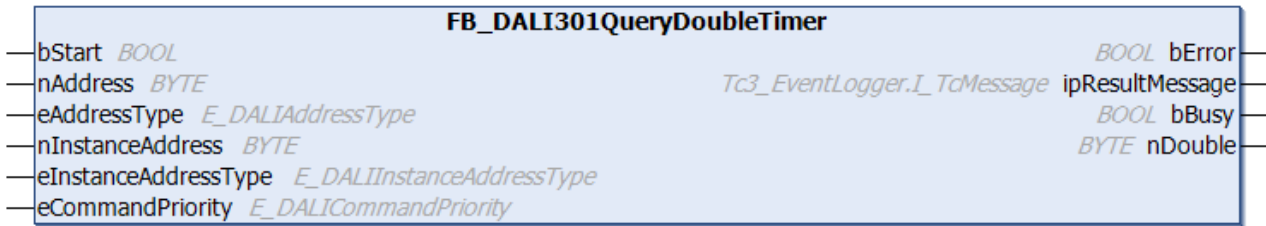


The *Button Stuck* event is triggered as soon as the push button was pressed for the time  $T_{stuck}$  without being released in between times. The *Button Free* event occurs immediately if the push button is released after the *Button Stuck* event.

The minimum set value for this time is 5 s; the default value is 20 s (see table above).

Note that the *Long Press* event does not temporally overlap the *Button Stuck* event, therefore the value should not be set too small.

4.1.2.11.2 FB\_DALI301QueryDoubleTimer



The function block reads the variable `tDouble` [▶ 531].

The value is read in steps of 20 ms. The maximum permissible value is 2000 ms or 100. The value 0 disables the timer. The minimum permissible value can be queried with `FB_DALI301QueryDoubleTimerMin` [▶ 516].

The output `nDouble` contains a valid value only if the function block was executed without errors (`bError = FALSE`).



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...31). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
```

```
bBusy          : BOOL;
nDouble       : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nDouble	BYTE	Value of the variable <i>tDouble</i> [20 ms]. The maximum output value is 100, as a multiple of the step size of 20 ms.

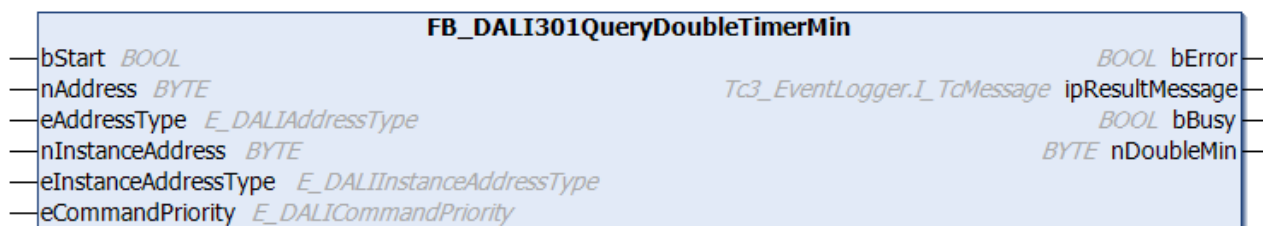
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.11.3 FB\_DALI301QueryDoubleTimerMin**



The function block reads the variable [tDoubleMin](#) [▶ 532].

The value is read in steps of 20 ms.

The output *nDoubleMin* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.



 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType <a href="#">▶ 861</a>	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nDoubleMin      : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nDoubleMin	BYTE	Value of the variable <i>tDoubleMin</i> [20 ms].

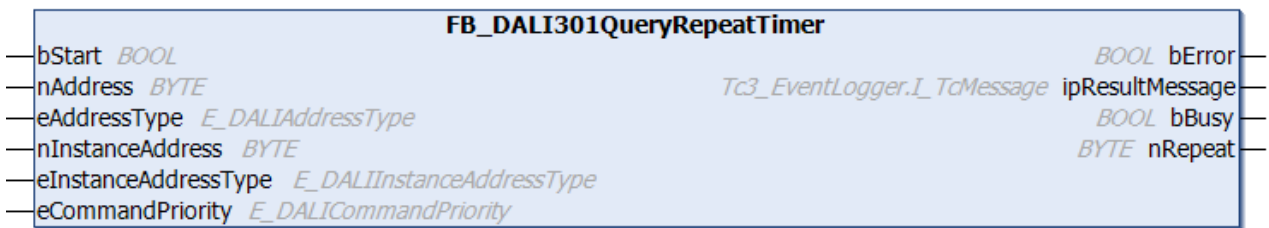
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.11.4 FB\_DALI301QueryRepeatTimer**



The function block reads the variable `tRepeat` [[▶ 532](#)].

The value is read in steps of 20 ms. The permissible value range is from 100 ms to 2000 ms. It is output in numbers from 5 to 100.

The output `nRepeat` only contains a valid value if the function block was executed without errors (`bError = FALSE`).



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nRepeat         : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nRepeat	BYTE	Value of the variable <i>tRepeat</i> [20 ms]. The output value is between 5 and 100, as a multiple of the step size of 20 ms.

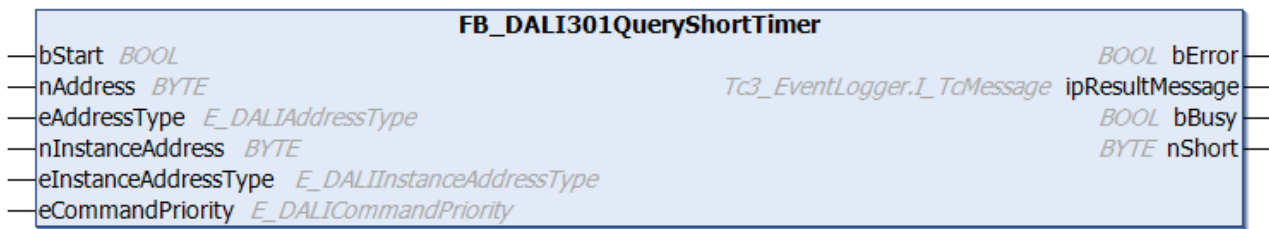
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.1.2.11.5 FB\_DALI301QueryShortTimer



The function block reads the variable `tShort` [► 531].

The value is read in steps of 20 ms. The maximum permissible value is 5100 ms, or 255. The minimum permissible value can be queried with `FB_DALI301QueryShortTimerMin` [► 521].

The output `nShort` contains a valid value only if the function block was executed without errors (`bError = FALSE`).



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...31). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [► 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nShort         : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nShort	BYTE	Value of the variable <i>tShort</i> [20 ms]. The maximum output value is 255, as a multiple of the step size of 20 ms.

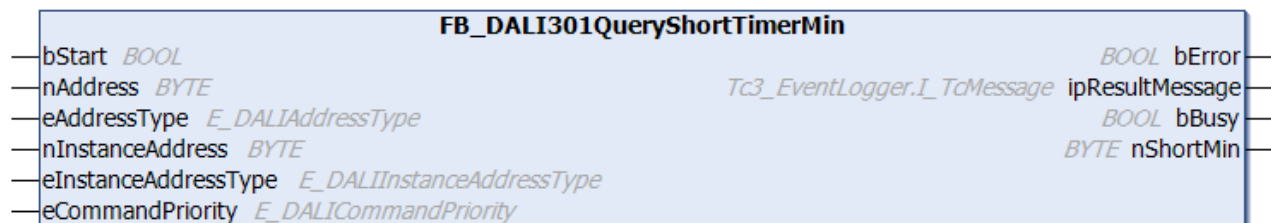
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.11.6 FB\_DALI301QueryShortTimerMin**



The function block reads the variable *tShortMin* [▶ 531].

The value is read in steps of 20 ms.

The output *nShortMin* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  
```

```
eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
eCommandPriority     : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	<a href="#">E_DALIInstanceAddressType</a> [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nShortMin       : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nShortMin	BYTE	Value of the variable <i>tShortMin</i> [20 ms].

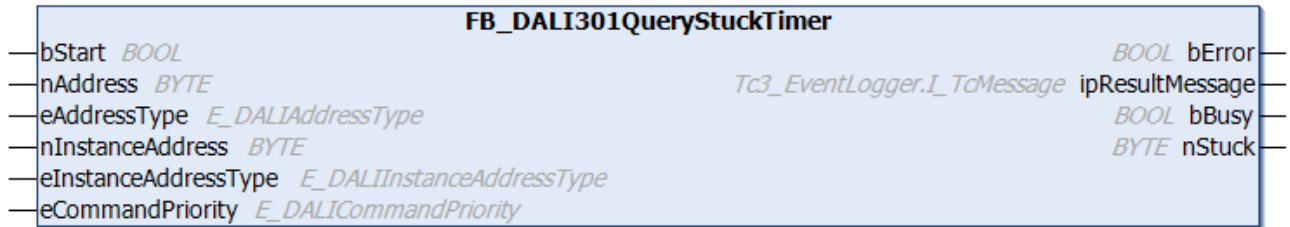
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.11.7 FB\_DALI301QueryStuckTimer



The function block reads the variable `tStuck` [► 532].

The value is read in steps of 1 s. The permissible value range is 5 s to 255 s.

The output `nStuck` contains a valid value only if the function block was executed without errors (`bError = FALSE`).



The function block cannot be used when using the KL6811.

Inputs

```

VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    nInstanceAddress : BYTE := 0;
    eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...31). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [► 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nStuck         : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nStuck	BYTE	Value of the variable <i>tStuck</i> [s].

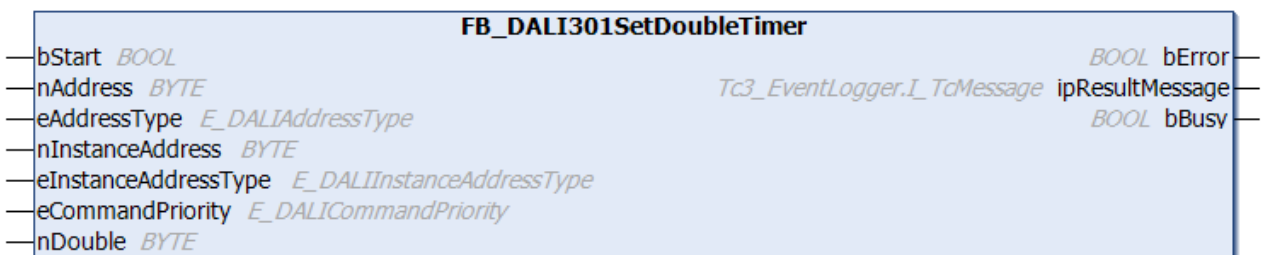
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.11.8 FB\_DALI301SetDoubleTimer**



The function block stores the value of *nDouble* in the variable *tDouble* [▶ 531] of the DALI control device.

The value is written in steps of 20 ms. The maximum permissible value is 2000 ms, or 100. The value 0 disables the timer. The smallest permitted value can be queried with the function block [FB\\_DALI301QueryDoubleTimerMin](#) [▶ 516].



The function block cannot be used when using the KL6811.



 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType   : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nDouble        : BYTE := 0;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDouble	BYTE	Value of the variable <i>tDouble</i> [20 ms]. The maximum input value is 100, as a multiple of the step size of 20 ms.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

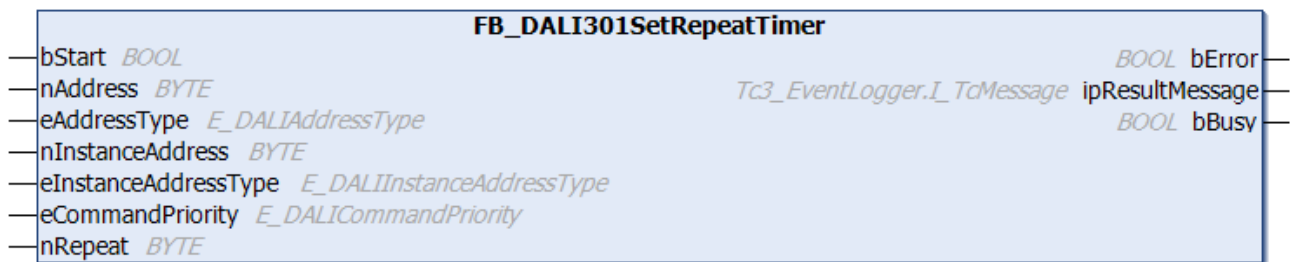
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.11.9 FB\_DALI301SetRepeatTimer**



The function block stores the value of *nRepeat* in the variable *tRepeat* [► 532] of the DALI control device.

The value is written in steps of 20 ms. The permissible value range is 100 ms to 2000 ms, i.e. from 5 to 100.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nRepeat         : BYTE := 8;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [► 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nRepeat	BYTE	Value of the variable <i>tRepeat</i> [20 ms]. The input value is between 2 and 100, as a multiple of the step size of 20 ms.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

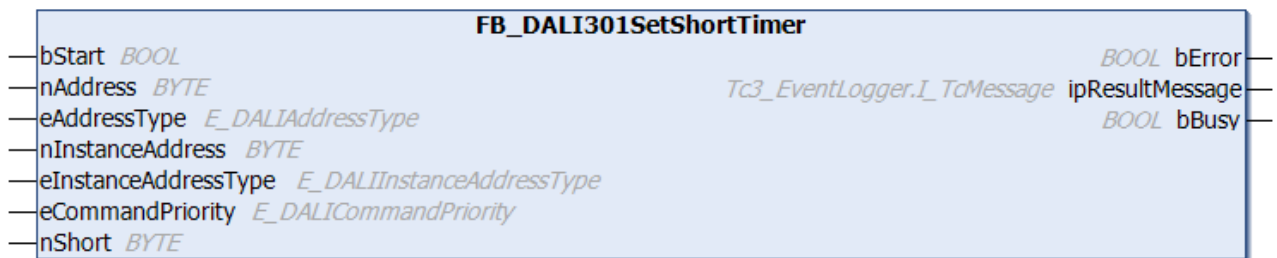
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.11.10 FB\_DALI301SetShortTimer**



The function block stores the value of *nShort* in the variable *tShort* [▶ 531] of the DALI control device.

The value is written in steps of 20 ms. The maximum permissible value is 5100 ms, or 255. The smallest permitted value can be queried with the function block [FB\\_DALI301QueryShortTimerMin](#) [▶ 521].



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nShort          : BYTE := 25;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	<a href="#">E_DALIInstanceAddressType</a> [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nShort	BYTE	Value of the variable <i>tShort</i> [20 ms]. The maximum input value is 255, as a multiple of the step size of 20 ms.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

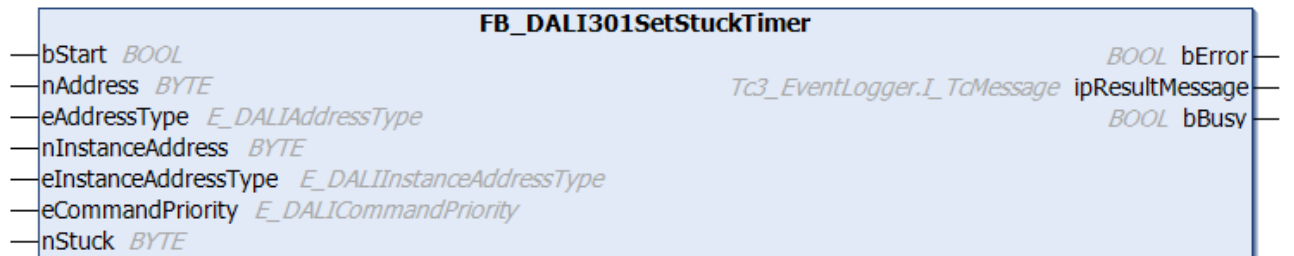
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.11.11 FB\_DALI301SetStuckTimer**



The function block stores the value of *nStuck* in the variable *tStuck* [[▶ 532](#)] of the DALI control device.

The value is read in steps of 1 s. The permissible value range is 5 s to 255 s.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    nInstanceAddress : BYTE := 0;
    eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
    nStuck          : BYTE := 20;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nStuck	BYTE	Value of the variable <i>tStuck</i> [s] (5...255 s)

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

#### 4.1.2.11.12 Variables

Each instance of a DALI control device has a certain number of variables (parameters). These variables are used to configure the instance and thus influence its behavior. The values of the variables are stored in the respective DALI control device.

Name	Reset value	Scope	Size	Comment
<a href="#">resolution</a> [► 531]	No change	1	1 byte	Read only
<a href="#">eventFilter</a> [► 531]	2#1111_0100	2#XXXX_XXXX	1 byte	
<a href="#">tShort</a> [► 531]	Max(25, <i>ShortMin</i> )	<i>tShortMin</i> ...255	1 byte	
<a href="#">tShortMin</a> [► 531]	No change	10...255	1 byte	Read only
<a href="#">tDouble</a> [► 531]	0	0, <i>tDoubleMin</i> ...100	1 byte	
<a href="#">tDoubleMin</a> [► 532]	No change	10...100	1 byte	Read only
<a href="#">tRepeat</a> [► 532]	8	5...100	1 byte	
<a href="#">tStuck</a> [► 532]	20	5...255	1 byte	

X: undetermined

##### resolution

Manufacturer-dependent value for the resolution of the input values.

The value of the variable can be read out with the function block [FB\\_DALI103QueryResolution](#) [► 208].

##### eventFilter

Specific events for each instance are enabled or disabled with the event filter (see [Event filter](#) [► 512]). If the respective bit is set, the corresponding event is enabled, otherwise it is disabled.

The function block [FB\\_DALI103QueryEventFilter](#) [► 188] can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter](#) [► 170] can be used to write to it.

##### tShort

The variable defines the time value for a short keystroke. After the time has elapsed, a long keystroke is detected.

The value must be  $\geq$  the value *tShortMin*. If *tShortMin* is  $< 25$ , then 25 is automatically adopted. According to the DALI standard, 25 (500 ms) is the physical minimum.

The value is processed in steps of 20 ms.

The function block [FB\\_DALI301QueryShortTimer](#) [► 520] can be used to query the value, the function block [FB\\_DALI301SetShortTimer](#) [► 527] can be used to specify it.

##### tShortMin

The variable *tShortMin* is a minimum value specified by the manufacturer for detecting a short keystroke.

The value can be queried with the function block [FB\\_DALI301QueryShortTimerMin](#) [► 521].

##### tDouble

The variable *tDouble* represents the time interval in which a repeated short keystroke must occur to be recognized as a double keystroke.

If the value is 0, the double keystroke event is disabled.

The value must be  $\geq$  the value *tDoubleMin*.

100 is the maximum value of the variable. It corresponds to a time of 2000 ms for a step size of 20 ms.

The function block [FB\\_DALI301QueryDoubleTimer](#) [► 515] can be used to query the value, the function block [FB\\_DALI301SetDoubleTimer](#) [► 524] can be used to specify it.

### tDoubleMin

The variable *tDoubleMin* is a minimum value specified by the manufacturer for detecting a double keystroke.

The value can be queried with the function block [FB\\_DALI301QueryDoubleTimerMin](#) [► 516].

### tRepeat

The variable *tRepeat* specifies the interval for repeating an event when a long keystroke is executed.

The value is given in steps of 20 ms. The permissible value range is 5 (100 ms) to 100 (2000 ms).

The function block [FB\\_DALI301QueryRepeatTimer](#) [► 518] can be used to query the value, the function block [FB\\_DALI301SetRepeatTimer](#) [► 526] can be used to specify it.

### tStuck

The variable *tStuck* indicates the time after a keystroke from which the key is considered to be stuck or defective.

The value is read in steps of 1 s. The permissible value range is 5 s to 255 s.

The function block [FB\\_DALI301QueryStuckTimer](#) [► 523] can be used to query the value, the function block [FB\\_DALI301SetStuckTimer](#) [► 529] can be used to specify it.

## 4.1.2.12 Part 302 (absolute input device)

### 4.1.2.12.1 Introduction

IEC 62386-302 describes DALI control devices that are defined as absolute input device.

DALI control devices according to IEC 62386-302 correspond to instance type 2. The type of an instance can be queried with the function block [FB\\_DALI103QueryInstanceType](#) [► 201].

#### Input value

The current state (*inputValue*) of the absolute encoder can be queried via the function block [FB\\_DALI103QueryInputValue](#) [► 194].

The variable can assume different values:

<i>inputValue</i>	Slide resistor	Analog input	Switch or binary input
0	Minimum position	Lower limit value	Opened
$1, 2^{\text{resolution}-2}$	Linear position display between the minimum and maximum position (with a <i>resolution</i> > 2)	Linear display of the value within the upper and lower limits.	Closed
$2^{\text{resolution}-1}$	Maximum position	Upper limit value	Closed

Depending on the connected DALI control device, the *inputValue* can assume values that depend on the *resolution* (see [Samples](#) [► 834]).

#### Timer

IEC 62386-302 defines specific timers with which the sending of the different events can be influenced.

The respective time is calculated from



$$Time = T_{incr} \times Multiplier$$

Time	Multiplier	Preset value of the multiplier	T <sub>incr</sub> Incremental value	T <sub>default</sub> Default value	T <sub>min</sub> lowest possible set value	T <sub>max</sub> highest possible set value
T <sub>deadtime</sub>	tDeadtime	2	50 ms	100 ms	0 s	12.75 s
T <sub>report</sub>	tReport	0	1 s	Vendor-specific	1 s	4 min 15 s



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.

### Event filter

DALI control devices are capable of sending different events. The event filter consists of an 8-bit value, where each individual bit defines whether an event is blocked or released. The event is released if the corresponding bit is set, otherwise it is blocked.

The function block [FB\\_DALI103QueryEventFilter \[► 188\]](#) can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter \[► 170\]](#) can be used to write to it.

The event filter (see [eventFilter \[► 542\]](#)) must be configured accordingly in order to trigger time-controlled events.

Bit	Event	Description
0	<i>Position Report</i>	<i>inputValue</i> has changed or the Report Timer has expired.
1	-	Reserved
2	-	Reserved
3	-	Reserved
4	-	Reserved
5	-	Reserved
6	-	Reserved
7	-	Reserved

### Input Notification event

With instance type 2 there is the possibility to release the event with the help of the event filter. The event is sent from the DALI control device by a *Input Notification* and can be received with [FB\\_DALIGetInputNotification \[► 841\]](#).

The following table lists the event.

Event Name	Event Information	Description
<i>Position Report</i>	position	If the resolution is less than or equal to 10, this event contains the current position (10 bits). If the resolution is greater than 10, the 10 MSB are transmitted with this event.

### Deadtime timer

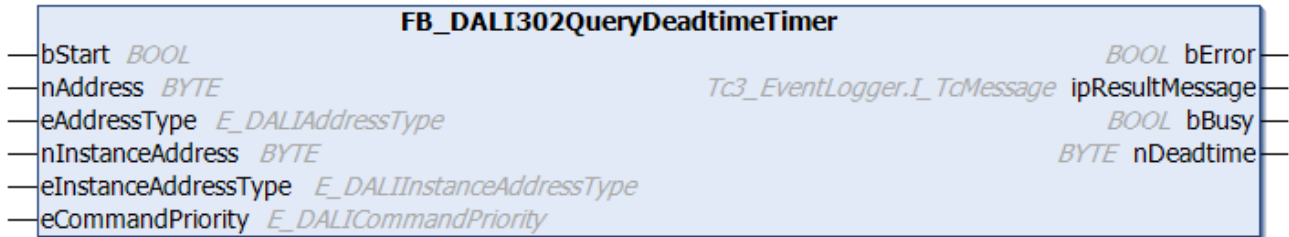
Several events may not be sent from an instance at the same time. To avoid this a dead time (T<sub>deadtime</sub>) can be activated (see [tDeadtime \[► 542\]](#)).

If the Deadtime Timer is active, an instance should not send any further events until the timer T<sub>deadtime</sub> has expired. The timer is restarted after the sending of an event.

If a new event occurs before the current event is sent, then the new event replaces the existing one. This can happen if the bus is not available or if the timer  $T_{\text{deadtime}}$  is in use.

If  $T_{\text{report}} < T_{\text{deadtime}}$ , then  $T_{\text{report}} = T_{\text{deadtime}}$ , regardless of which value  $T_{\text{report}}$  had before.

### 4.1.2.12.2 FB\_DALI302QueryDeadtimeTimer



The function block reads the variable `tDeadtime` [► 542].

The value is read in steps of 50 ms. The permissible value range is from 0 s to 12.75 s. It is output in numbers from 0 to 255.

The output *nDeadtime* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    nInstanceAddress : BYTE := 0;
    eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [► 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nDeadtime      : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bDeadtime	BYTE	Value of the variable <i>tDeadtime</i> [50 ms]. The output value is between 0 and 255, as a multiple of the step size of 50 ms.

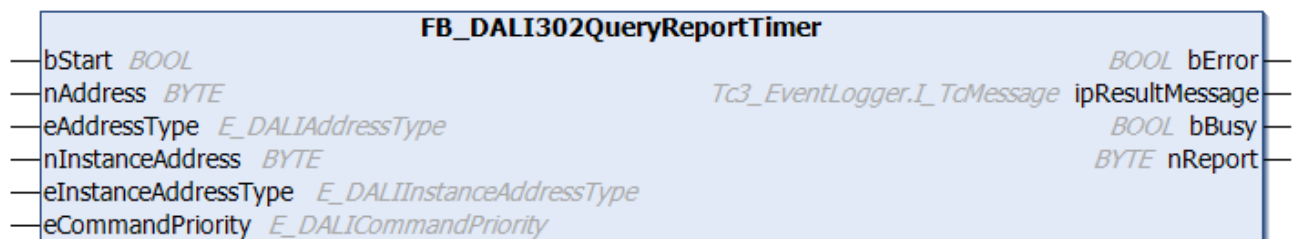
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.12.3 FB\_DALI302QueryReportTimer**



The function block reads the variable *tReport* [▶ 542].

The value is read in steps of 1 s. The permissible value range is from 0 s to 4 min 15 s. It is output in numbers from 0 to 255.

The output *nReport* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nReport         : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nReport	BYTE	Value of the variable <i>tReport</i> [1 s]. The output value is between 0 and 255, as a multiple of the step size of 1 s.

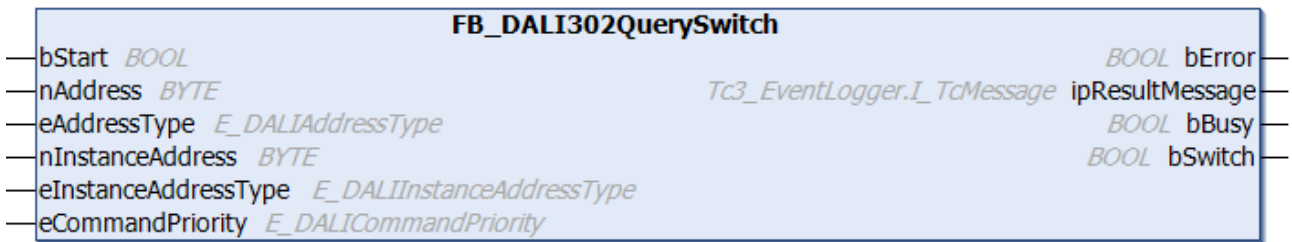
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

4.1.2.12.4 FB\_DALI302QuerySwitch



The function block queries whether the instance represents a switch.

The output *bSwitch* only contains a valid value if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 Inputs

```

VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    nInstanceAddress : BYTE := 0;
    eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType <a href="#">▶ 861</a>	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  bSwitch        : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bSwitch	BOOL	The output is set if the instance represents a switch.

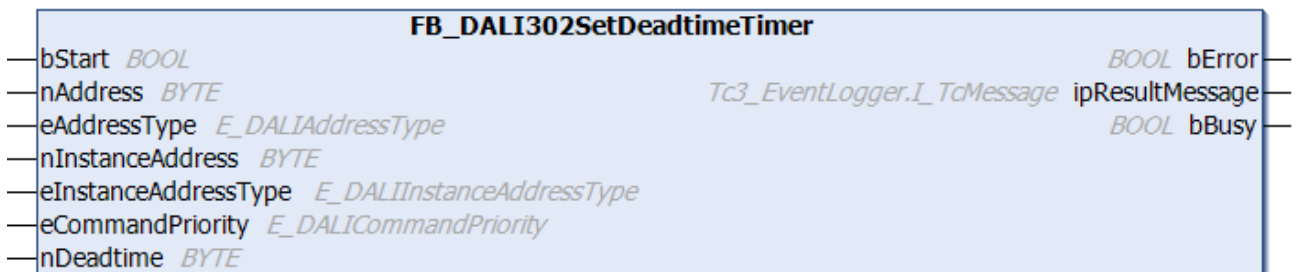
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.12.5 FB\_DALI302SetDeadtimeTimer**



The function block stores the value of *nDeadtime* in the variable *tDeadtime* [[▶ 542](#)] of the DALI control device.

The value is written in steps of 50 ms. The permissible value range is 0 s to 12.75 s. It is entered in numbers from 0 to 255.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nDeadtime      : BYTE := 2;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	<a href="#">E_DALIInstanceAddressType</a> [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDeadime	BYTE	Value of the variable <i>tDeadtime</i> [50 ms]. The input value is between 0 and 255, as a multiple of the step size of 50 ms.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

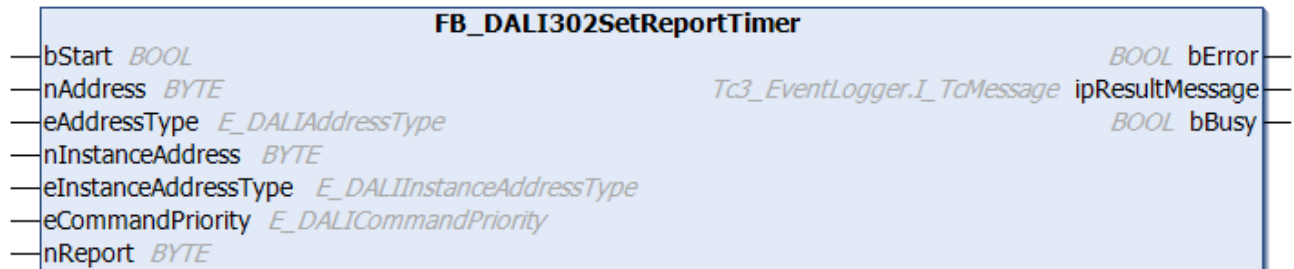
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.1.2.12.6 FB\_DALI302SetReportTimer**



The function block stores the value of *nReport* in the variable *tReport* [[► 542](#)] of the DALI control device.

The value is written in steps of 1 s. The permissible value range is 0 s to 4 min 15 s. It is entered in numbers from 0 to 255.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nReport     : BYTE := 0;
END_VAR
    
```



Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nReport	BYTE	Value of the variable <i>tReport</i> [1 s]. The input value is between 0 and 255, as a multiple of the step size of 1 s.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

Name	Type	Description

### 4.1.2.12.7 Variables

Each instance of a DALI control device has a certain number of variables (parameters). These variables are used to configure the instance and thus influence its behavior. The values of the variables are stored in the respective DALI control device.

Name	Reset value	Scope	Size	Comment
<a href="#">evenFilter</a> [ <a href="#">▶ 542</a> ]	2#0000_0001	2#0000_000X	1 byte	
<a href="#">tDeadtime</a> [ <a href="#">▶ 542</a> ]	2	0...255	1 byte	
<a href="#">tReport</a> [ <a href="#">▶ 542</a> ]	0	0...255	1 byte	

X: undetermined

#### eventFilter

Specific events for each instance are enabled or disabled with the event filter (see [Event filter](#) [[▶ 533](#)]). If the respective bit is set, the corresponding event is enabled, otherwise it is disabled.

The function block [FB\\_DALI103QueryEventFilter](#) [[▶ 188](#)] can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter](#) [[▶ 170](#)] can be used to write to it.

#### tDeadtime

The variable *tDeadtime* defines the time value for a dead time in which no new event is sent until the dead time has expired. If this timer is active, it is restarted after each transmitted event.

If the value is 0, the dead time event is disabled.

The value is given in steps of 50 ms. The permissible value range is 0 to 255 (12.75 s).

The function block [FB\\_DALI302QueryDeadtimeTimer](#) [[▶ 534](#)] can be used to query the value, the function block [FB\\_DALI302SetDeadtimeTimer](#) [[▶ 538](#)] can be used to specify it.

#### tReport

The variable *tReport* specifies the report time of an absolute input device after the occurrence of an event.

On expiry of the time specified in the variable *tReport*, an event is triggered and the time is restarted. If an event was triggered before the expiry of the time *tReport*, the time starts over.

The value is given in steps of 1 s. The permissible value range is 0 (0 s) to 255 (4 min 15 s).

The function block [FB\\_DALI302QueryReportTimer](#) [[▶ 535](#)] can be used to query the value, the function block [FB\\_DALI302SetReportTimer](#) [[▶ 540](#)] can be used to specify it.

## 4.1.2.13 Part 303 (occupancy sensor)

### 4.1.2.13.1 Introduction

IEC 62386-303 describes DALI control devices that are defined as occupancy sensors.

DALI control devices according to IEC 62386-303 correspond to instance type 3. The type of an instance can be queried with the function block [FB\\_DALI103QueryInstanceType](#) [► 201].

**Input value**

The current state (*inputValue*) of the light sensor can be queried via the function block [FB\\_DALI103QueryInputValue](#) [► 194].

The variable can assume four values:

<i>inputValue</i>	Occupancy	Movement
16#00	No	No
16#55	No	Yes
16#AA	Yes	No
16#FF	Yes	Yes

For simplification the graphics below only show whether an area is *occupied* or *vacant*.

**Timer**

IEC 62386-303 defines specific timers with which the sending of the different events can be influenced.

The respective time is calculated from

$$Time = T_{incr} \times Multiplier$$

Time	Multiplier	Preset value of the multiplier	T <sub>incr</sub> Incremental value	T <sub>default</sub> Default value	T <sub>min</sub> lowest possible set value	T <sub>max</sub> highest possible set value
T <sub>deadtime</sub>	<i>tDeadtime</i>	2	50 ms	100 ms	0 s	12.75 s
T <sub>hold</sub>	<i>tHold</i>	90	10 s	15 min	1 s	42 min 20 s
T <sub>report</sub>	<i>tReport</i>	20	1 s	20 s	1 s	4 min 15 s



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.

**Event filter**

DALI control devices are capable of sending different events. The event filter consists of an 8-bit value, where each individual bit defines whether an event is blocked or released. The event is released if the corresponding bit is set, otherwise it is blocked.

The function block [FB\\_DALI103QueryEventFilter](#) [► 188] can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter](#) [► 170] can be used to write to it.

The event filter (see [eventFilter](#) [► 560]) must be configured accordingly in order to trigger time-controlled events.

Bit	Event	Description
0	<i>Occupied</i>	Occupancy detection
1	<i>Vacant</i>	Area vacant detection
2	<i>Repeat</i>	Event repeat
3	<i>Movement</i>	Movement detection
4	<i>No Movement</i>	No Movement detection
5	-	reserved
6	-	reserved
7	-	reserved



Release only those events that you need for the application. If too many events are sent, this can have a negative effect on the behavior of the application.

### ***Input Notification event***

With instance type 3 it is possible to release different events with the help of the event filter. All events are sent from the DALI control device by a *Input Notification* and can be received with `FB_DALIGetInputNotification` [► 841].

The possible events (*Event Name* column) and their binary representation (*Event Information* column) are listed in the following table.

<b>Event Name</b>	<b>Event Information</b>	<b>Description</b>
<i>No Movement</i>	2#00_0000_XXX0	No movement detected.
<i>Movement</i>	2#00_0000_XXX1	Movement detected.
<i>Vacant</i>	2#00_0000_X00X	The area is not occupied.
<i>Still Vacant</i>	2#00_0000_X10X	The area is vacant over a lengthy period. The event occurs at regular intervals as long as the vacant condition is satisfied.
<i>Occupied</i>	2#00_0000_X01X	The area is occupied.
<i>Still Occupied</i>	2#00_0000_X11X	The area is occupied over a lengthy period. The event occurs at regular intervals as long as the occupied condition is satisfied.
<i>Presence Sensor</i>	2#00_0000_0XXX	The current event was triggered by an occupancy sensor.
<i>Movement Sensor</i>	2#00_0000_XXX1	The current event was triggered by a movement sensor.

The points marked with X can assume a value of either 0 or 1 and thus combine several events with one another.

### **Motion sensor**

The instance detects exclusively whether there is movement in the room or not.

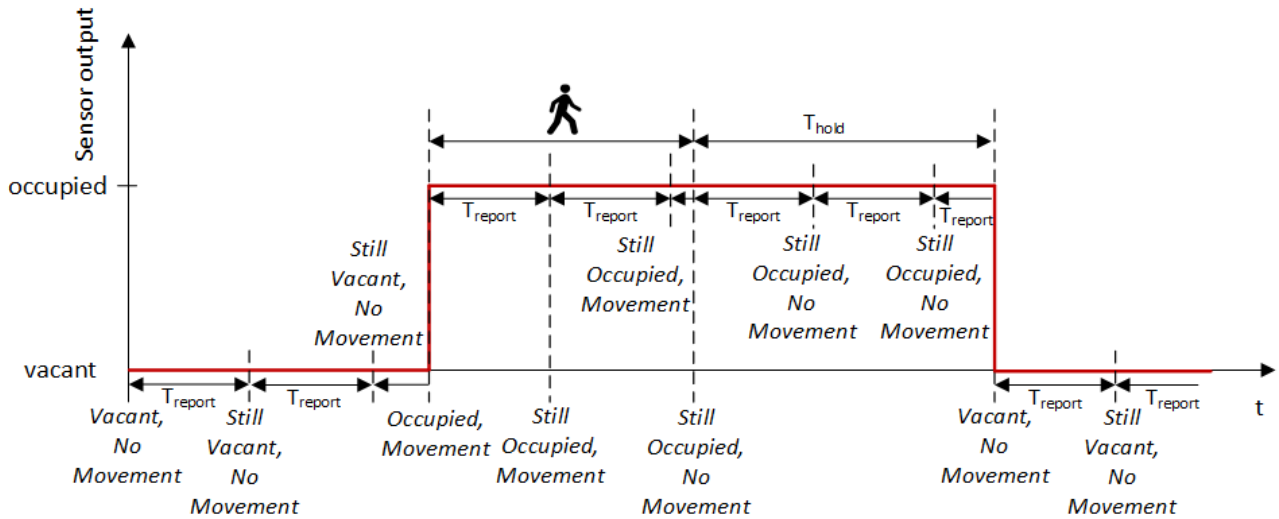
If no movement was detected within a certain time, the area is regarded as vacant.

### **Occupancy sensor**

The instance detects occupancy of the room. Some devices detect occupancy and movement.

Further information can be found in IEC 62386-303.

Events with HOLD TIMER

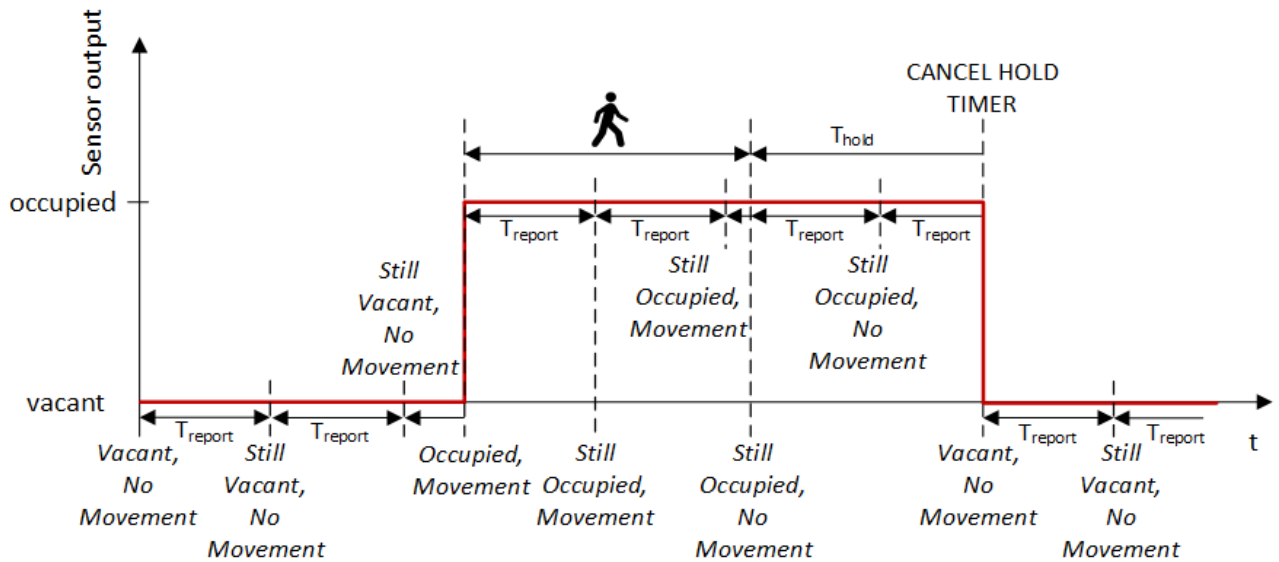


On expiry, the time  $T_{report}$  triggers the "Repeat" event (bit 2 of the event filter) (see [tReport](#) [► 561]). The time then starts over. The previous event remains active during the time  $T_{report} \cdot T_{report}$  is restarted by a newly detected event.

$T_{hold}$  provides for a hold time following the detection of movement (see [tHold](#) [► 560]). Not all devices support this variable.

Events with CANCEL HOLD TIMER

If the DALI device supports the hold timer, it can be deactivated via the command [FB DALI303CancelHoldTimer](#) [► 546].



The hold timer was deactivated before the expiry of its set time. The *Vacant* und *No Movement* events are triggered immediately at this moment.

Deadtime timer

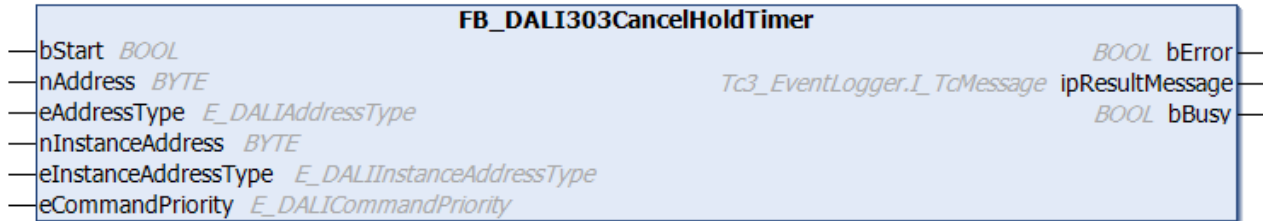
Several events may not be sent from an instance at the same time. To avoid this a dead time ( $T_{deadtime}$ ) can be activated (see [tDeadtime](#) [► 560]).

If the Deadtime Timer is active, an instance should not send any further events until the timer  $T_{deadtime}$  has expired. The timer is restarted after the sending of an event.

If a new event occurs before the current event is sent, then the new event replaces the existing one. This can happen if the bus is not available or if the timer  $T_{\text{Deadtime}}$  is in use.

If  $T_{\text{report}} < T_{\text{deadtime}}$ , then  $T_{\text{report}}$  should be equal to  $T_{\text{deadtime}}$  regardless of the value that  $T_{\text{report}}$  previously had.

### 4.1.2.13.2 FB\_DALI303CancelHoldTimer



The function block ends the Hold Timer prematurely if the latter was implemented in the DALI control device.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    nInstanceAddress : BYTE := 0;
    eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

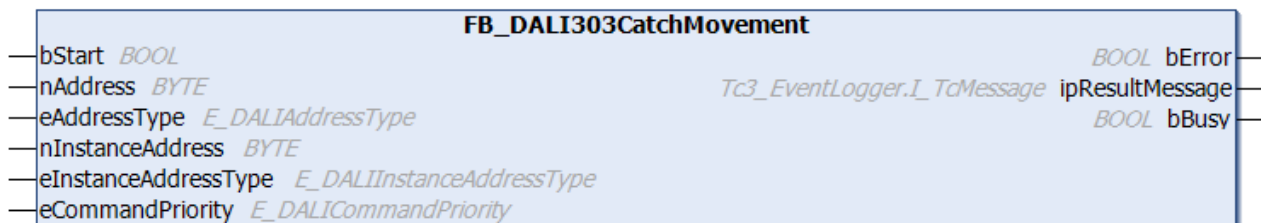
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.13.3 FB\_DALI303CatchMovement**



*Catching* is activated for the DALI control devices by this function block. If the DALI control device detects motion, an event is sent once and the *Catching* is reset.

The [Event Filter \[▶ 560\]](#) should be set so that the *Movement* event is disabled. This ensures that the event is only sent once as soon as *Catching* is active and movement is detected.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0



### 4.1.2.13.4 FB\_DALI303QueryCatching



The function block queries whether *Catching* is active.

The *Catching* can be enabled by [FB\\_DALI303CatchMovement](#) [► 547].

The output *bCatching* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [► 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

#### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  bCatching      : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bCatching	BOOL	<i>Catching</i> is active.

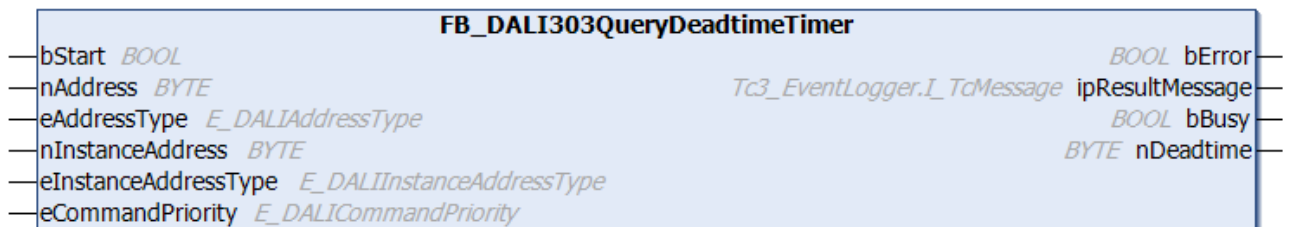
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.13.5 FB\_DALI303QueryDeadtimeTimer**



The function block reads the variable [tDeadtime](#) [▶ 560].

The value is read in steps of 50 ms. The permissible value range is from 0 s to 12.75 s. It is output in numbers from 0 to 255.

The output *nDeadtime* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nDeadtime       : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bDeadtime	BYTE	Value of the variable <i>tDeadtime</i> [50 ms]. The output value is between 0 and 255, as a multiple of the step size of 50 ms.

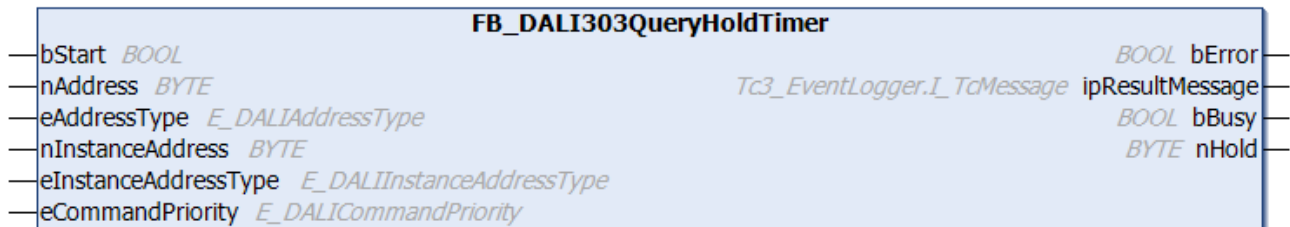
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.13.6 FB\_DALI303QueryHoldTimer**



The function block reads the variable `tHold` [► 560].

The value is read in steps of 10 s. The permissible value range is from 1 s to 42.5 min. It is output in numbers from 0 to 254.

The output `nHold` only contains a valid value if the function block was executed without errors (`bError = FALSE`).



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...31). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [► 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nHold           : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nHold	BYTE	Value of the variable <i>tHold</i> [10 s]. The output value is between 1 s and 42.5 min, as a multiple of the step size of 10 s.

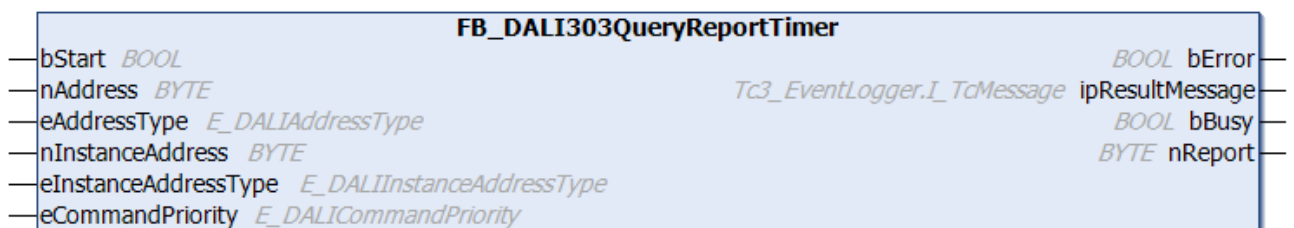
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.13.7 FB\_DALI303QueryReportTimer**



The function block reads the variable *tReport* [[▶ 561](#)].

The value is read in steps of 1 s. The permissible value range is from 0 s to 4 min 15 s. It is output in numbers from 0 to 255.

The output *nReport* contains a valid value only if the function block was executed without errors (*bError* = FALSE).



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType <a href="#">[▶ 861]</a>	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nReport         : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nReport	BYTE	Value of the variable <i>tReport</i> [1 s]. The output value is between 0 and 255, as a multiple of the step size of 1 s.

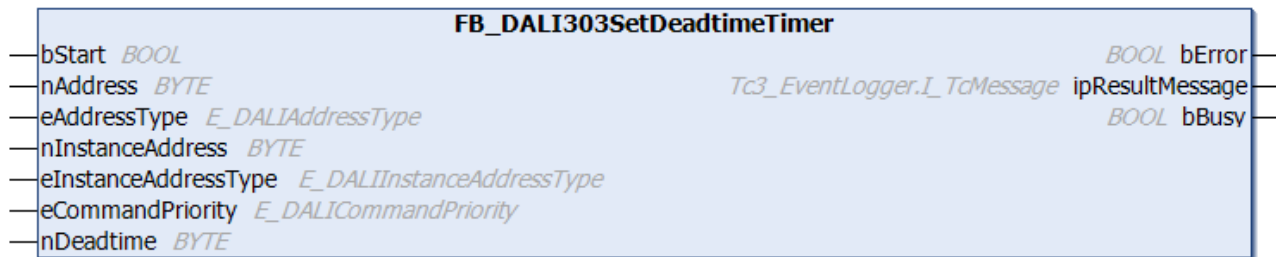
 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.13.8 FB\_DALI303SetDeadtimeTimer



The function block stores the value of *nDeadtime* in the variable *tDeadtime* [[▶ 560](#)] of the DALI control device.

The value is written in steps of 50 ms. The permissible value range is 0 s to 12.75 s. It is entered in numbers from 0 to 255.



The function block cannot be used when using the KL6811.

 Inputs

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nDeadtime       : BYTE := 2;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDeadime	BYTE	Value of the variable <i>tDeadtime</i> [50 ms]. The input value is between 0 and 255, as a multiple of the step size of 50 ms.

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

### Properties

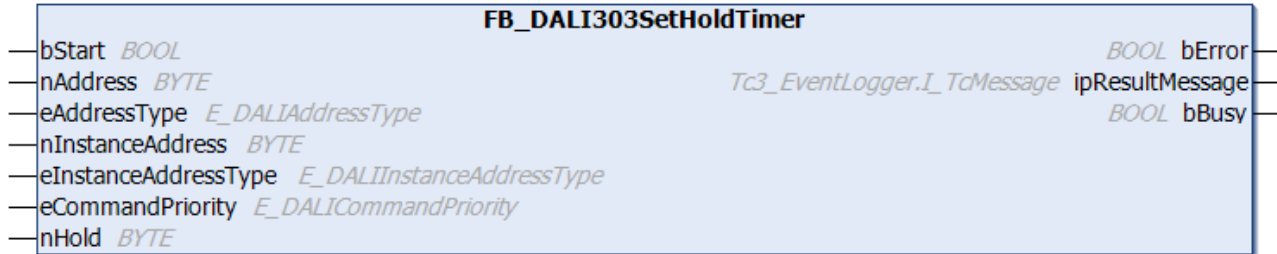
Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).



Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.13.9 FB\_DALI303SetHoldTimer



The function block stores the value of *nHold* in the variable *tHold* [▶ 560] of the DALI control device.

The value is written in steps of 10 s. The permissible value range is 1 s to 42.3 min. It is entered in numbers from 0 to 254. If the value 0 is passed, the variable *tHold* is set to 1 s.



The function block cannot be used when using the KL6811.

Inputs

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nHold      : BYTE := 90;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nHold	BYTE	Value of the variable <i>tHold</i> [10 s]. The input value is between 0 and 254, as a multiple of the step size of 10 s.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

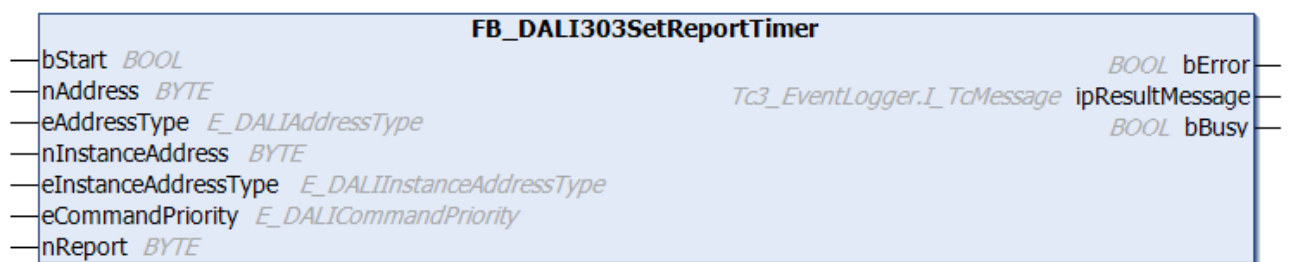
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICo mmunicati on	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.13.10 FB\_DALI303SetReportTimer**



The function block stores the value of *nReport* in the variable *tReport* [▶ 561] of the DALI control device.

The value is written in steps of 1 s. The permissible value range is 0 s to 4 min 15 s. It is entered in numbers from 0 to 255.



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart : BOOL;
  nAddress : BYTE;
```

```
eAddressType      : E_DALIAddressType := E_DALIAddressType.Short;
nInstanceAddress  : BYTE := 0;
eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
eCommandPriority  : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
nReport          : BYTE := 20;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nReport	BYTE	Value of the variable <i>tReport</i> [1 s]. The input value is between 0 and 255, as a multiple of the step size of 1 s.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.13.11 Variables**

Each instance of a DALI control device has a certain number of variables (parameters). These variables are used to configure the instance and thus influence its behavior. The values of the variables are stored in the respective DALI control device.

Name	Reset value	Scope	Size	Comment
<a href="#">resolution</a> [ <a href="#">▶ 560</a> ]	No change	2	1 byte	Read only
<a href="#">eventFilter</a> [ <a href="#">▶ 560</a> ]	2#0000_0011	2#000X_XXXX	1 byte	
<a href="#">tDeadtime</a> [ <a href="#">▶ 560</a> ]	2	0...255	1 byte	
<a href="#">tHold</a> [ <a href="#">▶ 560</a> ]	90	0...254	1 byte	
<a href="#">tReport</a> [ <a href="#">▶ 561</a> ]	20	0...255	1 byte	

X: not specified

**resolution**

Manufacturer-dependent value for the resolution of the input values.

The value of the variable can be read out with the function block [FB\\_DALI103QueryResolution](#) [[▶ 208](#)].

**eventFilter**

Specific events for each instance are enabled or disabled with the event filter (see [Event filter](#) [[▶ 543](#)]). If the respective bit is set, the corresponding event is enabled, otherwise it is disabled.

The function block [FB\\_DALI103QueryEventFilter](#) [[▶ 188](#)] can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter](#) [[▶ 170](#)] can be used to write to it.

**tDeadtime**

The variable defines the time value for a dead time in which no new event is sent until the dead time has elapsed. If this timer is active, it is restarted after each transmitted event.

If the value is 0, the dead time event is disabled.

The value is given in steps of 50 ms. The permissible value range is 0 to 255 (12.75 s).

The function block [FB\\_DALI303QueryDeadtimeTimer](#) [[▶ 550](#)] can be used to query the value, the function block [FB\\_DALI303SetDeadtimeTimer](#) [[▶ 555](#)] can be used to specify it.

**tHold**

The variable *tHold* specifies the hold time of an occupancy sensor after presence detection.

Not all devices support this variable.

The value is given in steps of 10 s. The permissible value range is 0 (1 s) to 254 (42.5 min).

The function block [FB\\_DALI303QueryHoldTimer \[▶ 552\]](#) can be used to query the value, the function block [FB\\_DALI303SetHoldTimer \[▶ 557\]](#) can be used to specify it.

The hold time *tHold* can be terminated prematurely with the function block [FB\\_DALI303CancelHoldTimer \[▶ 546\]](#).

**tReport**

The variable *tReport* specifies the report time of an occupancy sensor after the occurrence of an event.

After the time specified in the variable *tReport* has elapsed, the repeat event from the event filter is enabled. This keeps the occupied event active.

The value is given in steps of 1 s. The permissible value range is 0 (0 s) to 255 (4 min 15 s).

The function block [FB\\_DALI303QueryReportTimer \[▶ 553\]](#) can be used to query the value, the function block [FB\\_DALI303SetReportTimer \[▶ 558\]](#) can be used to specify it.

**4.1.2.14 Part 304 (light sensors)**

**4.1.2.14.1 Introduction**

IEC 62386-304 describes DALI control devices that are defined as brightness sensors.

DALI control devices according to IEC 62386-304 correspond to instance type 4. The type of an instance can be queried with the function block [FB\\_DALI103QueryInstanceType \[▶ 201\]](#).

**Input value**

The current state (*inputValue*) of the light sensor can be queried via the function block [FB\\_DALI103QueryInputValue \[▶ 194\]](#).

**Timer**

IEC 62386-304 defines various timers for triggering device-specific events.

The event for the instance type 4 in conjunction with the associated timers is explained below.

The respective time is calculated from

$$Time = T_{incr} \times Multiplier$$

Time	Multiplier	Preset value of the multiplier	T <sub>incr</sub> Incremental value	T <sub>default</sub> Default value	T <sub>min</sub> lowest possible set value	T <sub>max</sub> highest possible set value
T <sub>deadtime</sub>	<i>tDeadtime</i>	30	50 ms	1.5 s	0 s	12.75 s
T <sub>report</sub>	<i>tReport</i>	30	1 s	30 s	1 s	4 min 15 s



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.

**Event filter**

DALI control devices are capable of sending different events. The event filter consists of an 8-bit value, where each individual bit defines whether an event is blocked or released. The event is released if the corresponding bit is set, otherwise it is blocked.

The function block [FB\\_DALI103QueryEventFilter \[► 188\]](#) can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter \[► 170\]](#) can be used to write to it.

The event filter (see [eventFilter \[► 577\]](#)) must be configured accordingly in order to trigger time-controlled events.

Bit	Event	Description
0	<i>Illuminance level report</i>	Illuminance event
1	-	reserved
2	-	reserved
3	-	reserved
4	-	reserved
5	-	reserved
6	-	reserved
7	-	reserved

### **Input Notification event**

With instance type 4 there is the possibility to release the event with the help of the event filter. The event is sent from the DALI control device by a *Input Notification* and can be received with [FB\\_DALIGetInputNotification \[► 841\]](#).

The following table lists the event.

Event Name	Event Information	Description
<i>Illuminance Level Event</i>	<i>position</i>	If the resolution is less than or equal to 10, this event contains the current input value (10 bits). If the resolution is greater than 10, only a change of the input value is communicated with this event. This can then be read with <a href="#">FB_DALI103QueryInputValue [► 194]</a> .

### **Hysteresis**

To avoid too frequent, disruptive changes of the illuminance, it is possible with DALI devices with the instance type 4 to define a hysteresis.

A percentage value can be specified via the [hysteresis \[► 578\]](#) variable for the calculation of a hysteresis band.

The value can be queried with the function block [FB\\_DALI304QueryHysteresis \[► 565\]](#) and specified with [FB\\_DALI304SetHysteresis \[► 572\]](#).

Since the percentage hysteresis can also lead to wide fluctuations at low illuminance, a minimum hysteresis can be specified via the variable [hysteresisMin \[► 577\]](#). The minimum hysteresis is an absolute value with a range from 0 to 255.

The reset value depends on the resolution of the input values specified by the vendor.

resolution	Vendor-dependent reset value
1...6	0
7	1
8	2
9	5
10	10
11	20
12	40
13	81
14	163
> 15	255

The value can be queried with the function block [FB\\_DALI304QueryHysteresisMin](#) [▶ 567] and specified manually with [FB\\_DALI304SetHysteresisMin](#) [▶ 573].

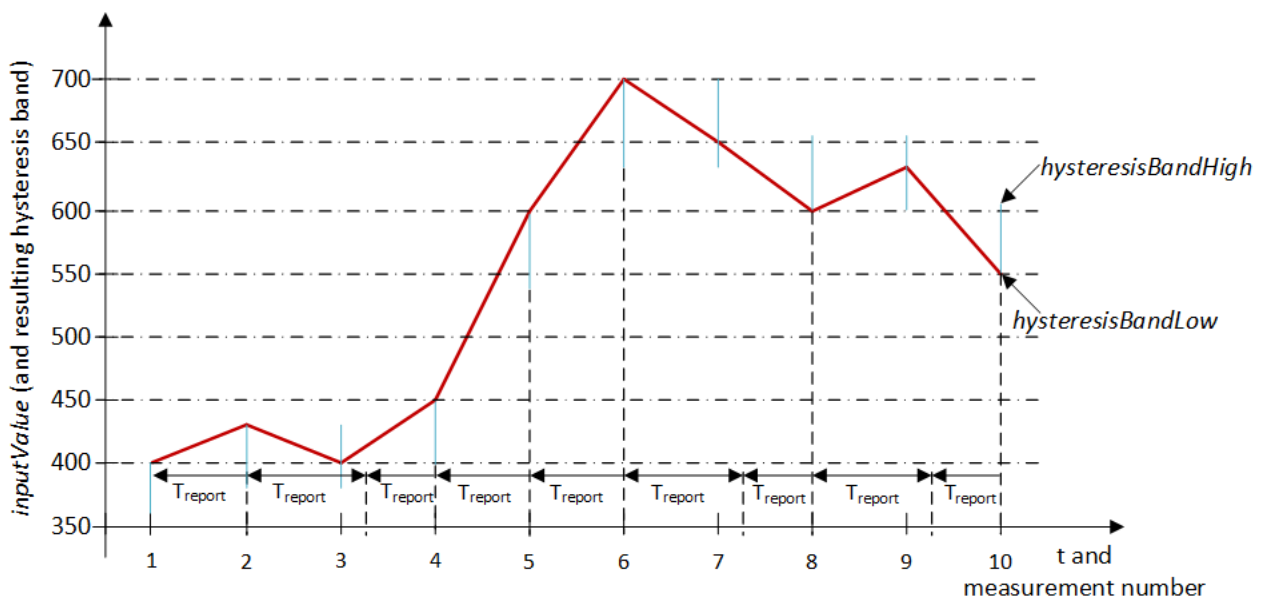
### Event Illuminance Level Report

The illuminance event is triggered each time an *inputValue* lies outside of the hysteresis band.

As described above, the hysteresis band is calculated from the percentage hysteresis of the input value and the value of the minimum hysteresis. The upper and lower points of the hysteresis band are called *hysteresisBandHigh* and *hysteresisBandLow* respectively.

The illuminance event is generated when the *inputValue* is greater than the calculated value *hysteresisBandHigh* or smaller than *hysteresisBandLow*.

It is also generated when the time  $T_{report}$  has expired, irrespective of the current input value.



The example shows the change in the input value with the resulting hysteresis band and the report timer  $T_{report}$ . The value for *hysteresis* is 10% and the value for *hysteresisMin* is 50.

The *Illuminance Level Report* event is generated at the measuring points 1, 2, 4, 5, 6, 8 and 10 through the change in the input value beyond the limits of the defined hysteresis band. No illuminance event is generated at the measuring points 3, 7 and 9 because the measured illuminance is within the hysteresis band and the timer  $T_{report}$  (see [tReport](#) [▶ 561]) has not yet expired. On expiry of the timer  $T_{report}$  the event is generated at a later time (between the measuring points 3 and 4, 7 and 8, 9 and 10).

**Deadtime timer**

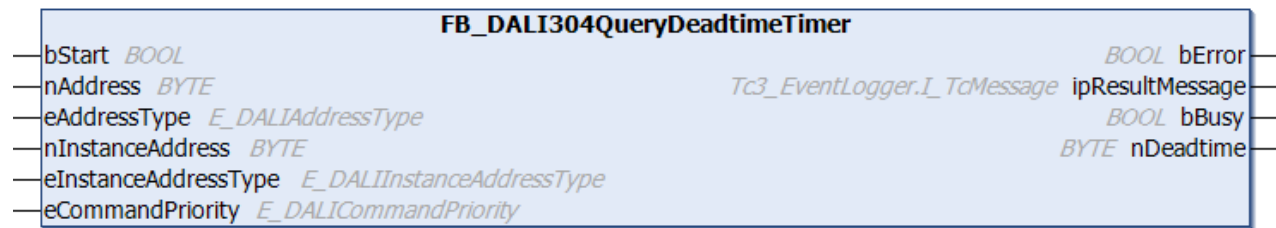
Several events may not be sent from an instance at the same time. To avoid this a dead time ( $T_{\text{deadtime}}$ ) can be activated (see [tDeadtime](#) [► 560]).

If the Deadtime Timer is active, an instance should not send any further events until the timer  $T_{\text{deadtime}}$  has expired. The timer is restarted after the sending of an event.

If a new event occurs before the current event is sent, then the new event replaces the existing one. This can happen if the bus is not available or if the timer  $T_{\text{Deadtime}}$  is in use.

If  $T_{\text{report}} < T_{\text{deadtime}}$ , then  $T_{\text{report}}$  should be equal to  $T_{\text{deadtime}}$  regardless of the value that  $T_{\text{report}}$  previously had.

**4.1.2.14.2 FB\_DALI304QueryDeadtimeTimer**



The function block reads the variable [tDeadtime](#) [► 577].

The value is read in steps of 50 ms. The permissible value range is 0 s to 12.75 s. It is output in numbers from 0 to 255.



The function block cannot be used when using the KL6811.

The output *nDeadtime* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [► 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	<a href="#">E_DALIInstanceAddressType</a> [► 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.





The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nDeadtime       : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
bDeadtime	BYTE	Value of the variable <i>tDeadtime</i> [50 ms]. The output value is between 0 and 255, as a multiple of the step size of 50 ms.



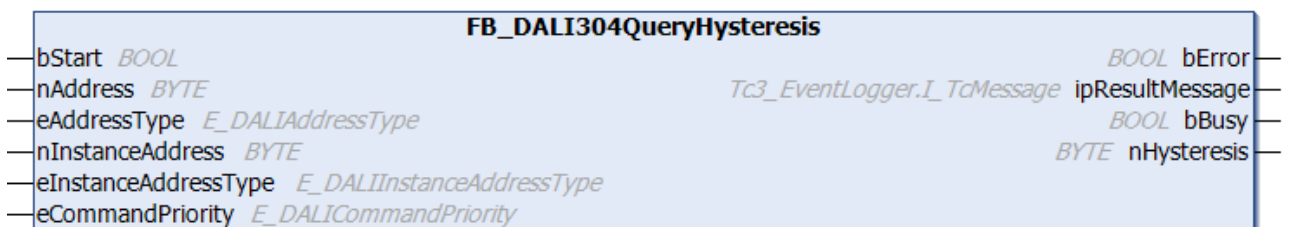
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.14.3 FB\_DALI304QueryHysteresis**



The function block reads the variable [hysteresis \[▶ 578\]](#).

The value is stated in %. The permissible value range is between 0% and 25%.



The function block cannot be used when using the KL6811.

The output *nHysteresis* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

**Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
  nHysteresis     : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nHysteresis	BYTE	Value of the variable <i>hysteresis</i> in %.

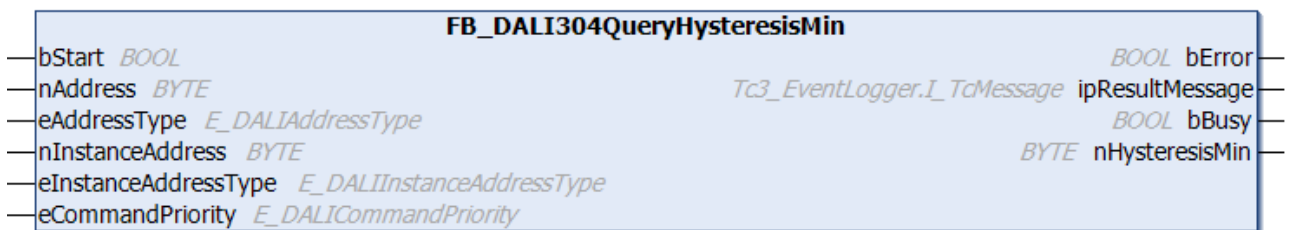
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.14.4 FB\_DALI304QueryHysteresisMin**



The function block reads the variable [hysteresisMin](#) [▶ 577].

The permissible value range is between 0 and 255.

Since the hysteresis is given as a percentage, the absolute value of the hysteresis depends on the actual measured value. For very small measured values, the hysteresis is therefore also very small. This causes unnecessary events to be sent. For this reason, a minimum possible hysteresis can be set.



The function block cannot be used when using the KL6811.

The output *nHysteresisMin* contains a valid value only if the function block was executed without errors (*bError* = FALSE).

 **Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
```

```
nInstanceAddress      : BYTE := 0;
eInstanceAddressType  : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
eCommandPriority      : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

### Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nHysteresisMin  : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nHysteresisMin	BYTE	Value of the smallest possible hysteresis.

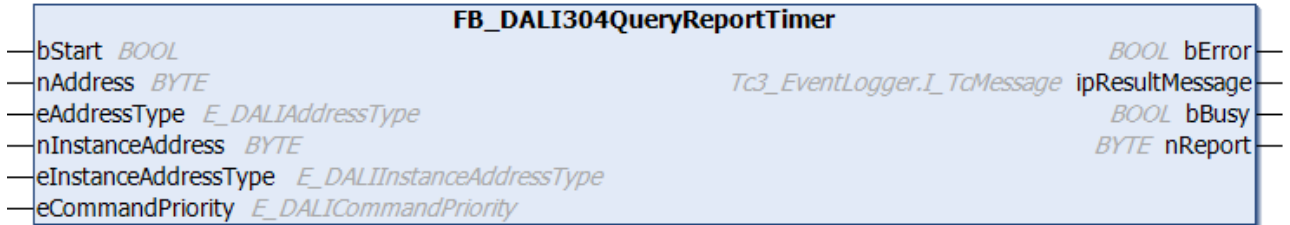
### Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.14.5 FB\_DALI304QueryReportTimer



The function block reads the variable `tReport` [► 577].

The value is read in steps of 1 s. The permissible value range is 0 s to 4 min 15 s. It is output in numbers from 0 to 255.



The function block cannot be used when using the KL6811.

The output `nReport` contains a valid value only if the function block was executed without errors (`bError = FALSE`).

Inputs

```

VAR_INPUT
    bStart          : BOOL;
    nAddress        : BYTE;
    eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
    nInstanceAddress : BYTE := 0;
    eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
    eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [► 854]	Defines whether the input <code>nAddress</code> contains a short address (0...63) or a group address (0...31). The input <code>nAddress</code> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [► 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [► 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy          : BOOL;
  nReport        : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nReport	BYTE	Value of the variable <i>tReport</i> [1 s]. The output value is between 0 and 255, as a multiple of the step size of 1 s.

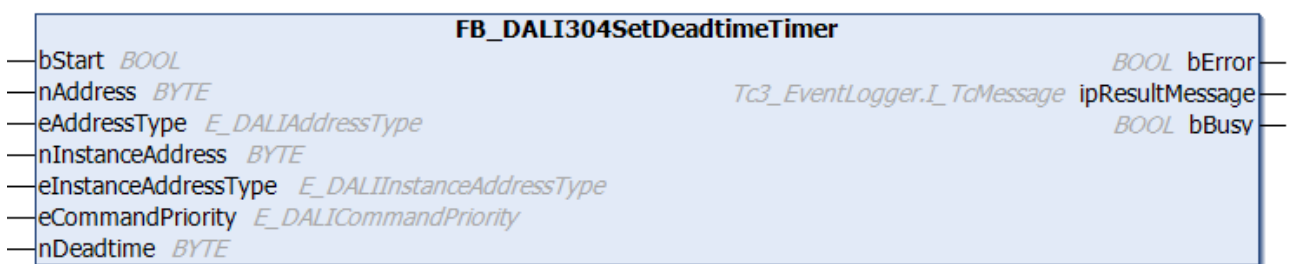
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.14.6 FB\_DALI304SetDeadtimeTimer**



The function block stores the value of *nDeadtime* in the variable *tDeadtime* [▶ 577] of the DALI control device.

The value is written in steps of 50 ms. The permissible value range is 0 s to 12.75 s. It is entered in numbers from 0 to 255.



The function block cannot be used when using the KL6811.

 **Inputs**

```
VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nDeadtime      : BYTE := 30;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">▶ 854</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType <a href="#">▶ 861</a>	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority <a href="#">▶ 856</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nDeadime	BYTE	Value of the variable <i>tDeadtime</i> [50 ms]. The input value is between 0 and 255, as a multiple of the step size of 50 ms.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation ▶ 887</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages ▶ 873</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

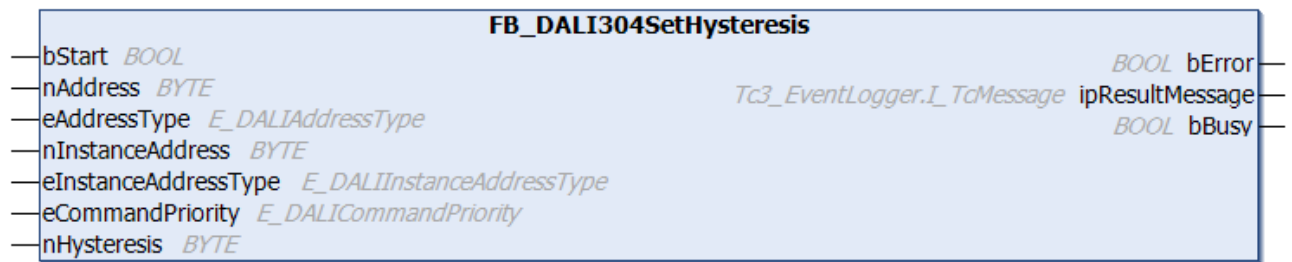
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.14.7 FB\_DALI304SetHysteresis**



The function block stores the value *nHysteresis* in the variable [hysteresis](#) [[▶ 578](#)] of the DALI control device.

The value is stated in %. The permissible value range is between 0% and 25%.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart          : BOOL;
  nAddress        : BYTE;
  eAddressType    : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nHysteresis     : BYTE := 5;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	<a href="#">E_DALIAddressType</a> [ <a href="#">▶ 854</a> ]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	<a href="#">E_DALIInstanceAddressType</a> [ <a href="#">▶ 861</a> ]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	<a href="#">E_DALICommandPriority</a> [ <a href="#">▶ 856</a> ]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.





The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nHysteresis	BYTE	Input value of the variable <i>hysteresis</i> in %.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

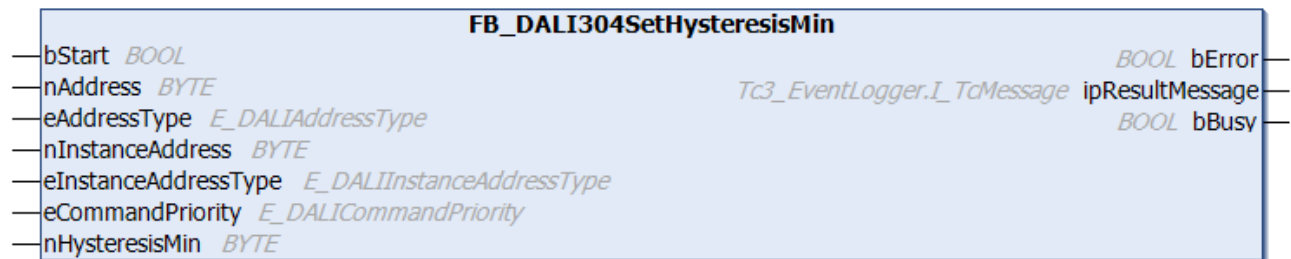
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.14.8 FB\_DALI304SetHysteresisMin**



The function block stores the value *nHysteresisMin* in the variable [hysteresisMin \[► 577\]](#) of the DALI control device.

Since the hysteresis is given as a percentage, the absolute value of the hysteresis depends on the actual measured value. For very small measured values, the hysteresis is therefore also very small. This causes unnecessary events to be sent. For this reason, a minimum possible hysteresis can be set.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nHysteresisMin : BYTE := 10;
END_VAR
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType <a href="#">[▶ 854]</a>	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType <a href="#">[▶ 861]</a>	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority <a href="#">[▶ 856]</a>	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nHysteresisMin	BYTE	Value of the smallest possible hysteresis <i>hysteresisMin</i> .

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy       : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

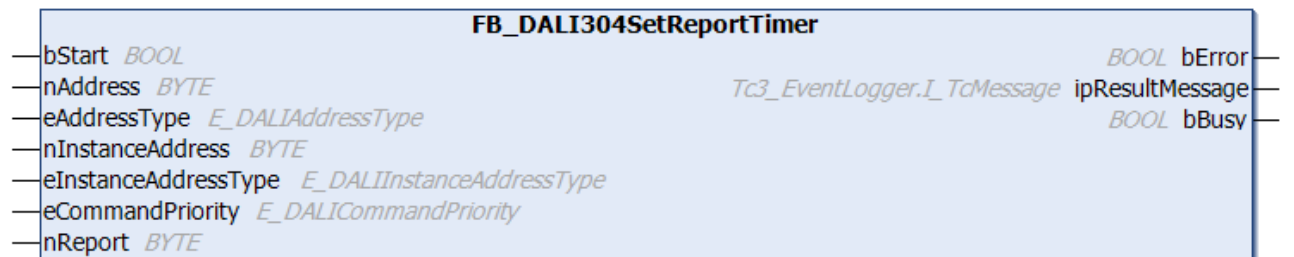
 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.2.14.9 FB\_DALI304SetReportTimer**



The function block stores the value of *nReport* in the variable *tReport* [[▶ 577](#)] of the DALI control device.

The value is written in steps of 1 s. The permissible value range is 0 s to 4 min 15 s. It is entered in numbers from 0 to 255.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bStart      : BOOL;
  nAddress    : BYTE;
  eAddressType : E_DALIAddressType := E_DALIAddressType.Short;
  nInstanceAddress : BYTE := 0;
  eInstanceAddressType : E_DALIInstanceAddressType := E_DALIInstanceAddressType.InstanceNumber;
  eCommandPriority : E_DALICommandPriority := E_DALICommandPriority.MiddleLow;
  nReport     : BYTE := 30;
END_VAR
    
```

Name	Type	Description
bStart	BOOL	Execution of the DALI commands is triggered via a positive edge at this input.
nAddress	BYTE	Address of a DALI control gear or a DALI group.
eAddressType	E_DALIAddressType [▶ 854]	Defines whether the input <i>nAddress</i> contains a short address (0...63) or a group address (0...31). The input <i>nAddress</i> has no meaning if a broadcast or a broadcast to unaddressed devices (BroadcastUnaddr) has been selected.
nInstanceAddress	BYTE	Instance number within a DALI control device. A DALI control device can support multiple instance numbers.
eInstanceAddressType	E_DALIInstanceAddressType [▶ 861]	Defines the access mode to the desired instance within the DALI control device.
eCommandPriority	E_DALICommandPriority [▶ 856]	Priority (low, middle low, middle, middle high, high) with which the DALI commands are sent.



The command priorities are supported by the KL6821 from the firmware version BD.

Name	Type	Description
nReport	BYTE	Value of the variable <i>tReport</i> [1 s]. The input value is between 0 and 255, as a multiple of the step size of 1 s.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bBusy           : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [▶ 892]).

Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.2.14.10 Variables

Each instance of a DALI control device has a certain number of variables (parameters). These variables are used to configure the instance and thus influence its behavior. The values of the variables are stored in the respective DALI control device.

Name	Reset value	Scope	Size	Comment
<a href="#">eventFilter [▶ 577]</a>	2#0000_0001	2#0000_000X	1 byte	
<a href="#">tReport [▶ 577]</a>	30	0...255	1 byte	
<a href="#">tDeadtime [▶ 577]</a>	30	0...255	1 byte	
<a href="#">hysteresisMin [▶ 577]</a>	See table	0...255	1 byte	
<a href="#">hysteresis [▶ 578]</a>	5	0...25	1 byte	

X: not specified

**eventFilter**

Specific events for each instance are enabled or disabled with the event filter (see [Event filter \[▶ 561\]](#)). If the respective bit is set, the corresponding event is enabled, otherwise it is disabled.

The function block [FB\\_DALI103QueryEventFilter \[▶ 188\]](#) can be used to read the value of the event filter; the function block [FB\\_DALI103SetEventFilter \[▶ 170\]](#) can be used to write to it.

**tReport**

The variable *tReport* specifies the report time of a brightness sensor after the occurrence of an event.

The report time is restarted when a new event is sent.

The value is given in steps of 1 s. The permissible value range is 0 (0 s) to 255 (4 min 15 s).

The function block [FB\\_DALI304QueryReportTimer \[▶ 569\]](#) can be used to query the value, the function block [FB\\_DALI304SetReportTimer \[▶ 575\]](#) can be used to specify it.

**tDeadtime**

The variable defines the time value for a dead time in which no new event is sent until the dead time has elapsed. If this timer is active, it is restarted after each transmitted event.

If the value is 0, the dead time event is disabled.

The value is given in steps of 50 ms. The permissible value range is 0 to 255 (12.75 s).

The function block [FB\\_DALI304QueryDeadtimeTimer \[▶ 564\]](#) can be used to query the value, the function block [FB\\_DALI304SetDeadtimeTimer \[▶ 570\]](#) can be used to specify it.

**hysteresisMin**

The variable *hysteresisMin* prevents large fluctuations in the illumination at low illuminance and thus low-percentage hysteresis.

The variable is an absolute value with a value range between 0 and 255.

The reset value depends on the resolution of the input values specified by the vendor.

resolution	Vendor-dependent reset value
1...6	0
7	1
8	2
9	5
10	10
11	20
12	40
13	81
14	163
>15	255

The value can be queried with the function block [FB\\_DALI304QueryHysteresisMin](#) [► 567] and specified with [FB\\_DALI304SetHysteresisMin](#) [► 573].

### hysteresis

The variable *hysteresis* indicates a percentage value of the hysteresis around a measured illuminance.

A value of 0 disables the hysteresis.

The function block [FB\\_DALI304QueryHysteresis](#) [► 565] can be used to query the value, the function block [FB\\_DALI304SetHysteresis](#) [► 572] can be used to specify it.

## 4.1.3 Communication

### 4.1.3.1 FB\_EL6821Communication

FB_EL6821Communication	
— bResetMaximumDemandCounter <i>BOOL</i>	<i>BOOL</i> bError
— bResetOverflowCounter <i>BOOL</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
— nOptions <i>DWORD</i>	<i>BOOL</i> bBusy
— bResetInactiveProcessImage <i>BOOL</i>	<i>BYTE</i> nBufferDemandMeter
— bInitialise <i>BOOL</i>	<i>BYTE</i> nBufferMaximumDemandMeter
— eCommandEBusWatchdog <i>E_DALIConfigurationCommand</i>	<i>UINT</i> nBufferOverflowCounter
— eCommandDI1RisingEdge <i>E_DALIConfigurationCommand</i>	<i>BOOL</i> bDigitalInput1Active
— eCommandDI1FallingEdge <i>E_DALIConfigurationCommand</i>	<i>BOOL</i> bDigitalInput2Active
— eCommandDI2RisingEdge <i>E_DALIConfigurationCommand</i>	<i>BOOL</i> bProcessImageInactive
— eCommandDI2FallingEdge <i>E_DALIConfigurationCommand</i>	<i>BOOL</i> bPowerSupplyError
— bDoNotLockProcessImage <i>BOOL</i>	<i>BOOL</i> bShortCircuit
— bDisableInternalPowerSupply <i>BOOL</i>	<i>BOOL</i> bInitialising
— stInData <i>ST_EL6821InData</i>	<i>BOOL</i> bQuiescentModeActive
— stOutData <i>ST_EL6821OutData</i>	<i>STRING(10)</i> sTerminalDescription
	<i>STRING(10)</i> sFirmwareVersion

The function blocks for the DALI commands do not directly access the process image of the EL6821, but place the individual DALI commands in a command buffer. The function block [FB\\_EL6821Communication](#) sequentially reads the DALI commands from the command buffer and forwards them to the EL6821. This prevents multiple function blocks accessing the EL6821 process image at the same time.

The function block also reads the events of the DALI control devices from the EL6821 and stores them in a special table. The function blocks [FB\\_DALIGetInputNotification](#) [► 841] and [FB\\_DALIGetPowerCycleNotification](#) [► 843] are used to filter out the desired events from this table.

One instance of the function block [FB\\_EL6821Communication](#) must be created for each EL6821. This instance must be called in a separate, faster task. This faster communication task must have a higher priority than the task in which the function blocks for the individual DALI commands are called.

The utilization rate of the command buffer can be determined using the outputs of the function block. If you find that the command buffer is overflowing regularly, you should take the following steps:

- How heavily are the individual PLC tasks utilized? TwinCAT provides suitable analysis tools.
- Try to reduce the cycle time of the task in which the function block FB\_EL6821Communication is called. The value should not be greater than 6 ms. The optimum value is 2 ms or less.
- If possible avoid polling (regular reading) of values. Only read values when they are actually required.
- Distribute the individual DALI devices evenly over several DALI lines. Since several DALI lines are processed simultaneously in each PLC cycle, this increases the data throughput.

For more information, see the chapter [Bus Timing](#) [▶ 12].

## Inputs

```
VAR_INPUT
  bResetMaximumDemandCounter : BOOL;
  bResetOverflowCounter      : BOOL;
  nOptions                   : DWORD := 0;
  bResetInactiveProcessImage : BOOL;
  bInitialise                 : BOOL := FALSE;
  eCommandEBusWatchdog       : E_DALIConfigurationCommand := E_DALIConfigurationCommand.DoNothing;
  eCommandDI1RisingEdge      : E_DALIConfigurationCommand := E_DALIConfigurationCommand.Off;
  eCommandDI1FallingEdge     : E_DALIConfigurationCommand := E_DALIConfigurationCommand.DoNothing;
  eCommandDI2RisingEdge      : E_DALIConfigurationCommand := E_DALIConfigurationCommand.RecallMaxLev
el;
  eCommandDI2FallingEdge     : E_DALIConfigurationCommand := E_DALIConfigurationCommand.DoNothing;
  bDoNotLockProcessImage     : BOOL := FALSE;
  bDisableInternalPowerSupply: BOOL := FALSE;
END_VAR
```

Name	Type	Description
bResetMaximumDemandCounter	BOOL	A positive edge resets the stored value for the maximum utilization of the command buffer, <i>nBufferMaximumDemandMeter</i> (0...100 %).
bResetOverflowCounter	BOOL	A positive edge resets the stored value for the number of overflows of the command buffer, <i>nBufferOverflowCounter</i> .
nOptions	DWORD	Reserved for future extensions.
bResetInactiveProcessImage	BOOL	A positive edge cancels the blocking of the process image of the terminal. The output <i>bProcessImageInactive</i> is reset to FALSE. The blocking is activated as soon as one of the two digital inputs on the terminal has been actuated and the input parameter <i>bDoNotLockProcessImage</i> is FALSE.
bInitialise	BOOL	Configuration of the bus terminal is started by a positive edge at this input. Initialization is also carried out automatically when the controller is started. During this time no DALI commands are processed.
eCommandEBusWatchdog	E_DALIConfigurationCommand	Defines the DALI command that is sent as soon as the bus terminal is no longer addressed via the E-bus. The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.
eCommandDI1RisingEdge, eCommandDI2RisingEdge	E_DALIConfigurationCommand	Defines the DALI command that is sent as soon as a rising edge is detected at the respective input of the bus terminal. The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.
eCommandDI1FallingEdge, eCommandDI2FallingEdge	E_DALIConfigurationCommand	Defines the DALI command that is sent as soon as a falling edge is detected at the respective input of the bus terminal. The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.
bDoNotLockProcessImage	BOOL	Defines whether the process image for the PLC is not blocked by actuating the digital inputs (see also <i>bInactiveProcessImage</i> ). The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.
bDisableInternalPowerSupply	BOOL	Defines the operation mode of the internal DALI power supply unit. The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.

 **Inputs/outputs**

```
VAR_IN_OUT
  stInData          : ST_EL6821InData;
  stOutData         : ST_KL6821OutData;
END_VAR
```

Name	Type	Description
stInData	ST_EL6821InData [ <a href="#">▶ 853</a> ]	Structure in the input process image of the EL6821. It is used for communication from the EL6821 to the PLC.
stOutData	ST_EL6821OutData [ <a href="#">▶ 853</a> ]	Structure in the output process image of the EL6821. It is used for communication from the PLC to the EL6821.

 **Outputs**

```
VAR_OUTPUT
  bError              : BOOL;
  ipResultMessage    : I_TcMessage;
  bBusy               : BOOL;
  nBufferDemandMeter : BYTE;
  nBufferMaximumDemandMeter : BYTE;
  nBufferOverflowCounter : UINT;
  bDigitalInputnActive : BOOL;
  bProcessImageInactive : BOOL;
```



```
bCollisionError      : BOOL;
bPowerSupplyError    : BOOL;
bShortCircuit        : BOOL;
bInitialising        : BOOL;
bQuiescentModeActive : BOOL;
sTerminalDescription : STRING(10);
sFirmwareVersion     : STRING(10);
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nBufferDemandMeter	BYTE	Utilization rate of the command buffer (0...100 %).
nBufferMaximumDemandMeter	BYTE	Maximum utilization rate of the command buffer reached so far (0...100 %). The counter can be reset via the input <i>bResetMaximumDemandCounter</i> .
nBufferOverflowCounter	UINT	Number of command buffer overflows to date. The counter can be reset via the input <i>bResetOverflowCounter</i> .
bDigitalInput1Active, bDigitalInput2Active	BOOL	One of the digital inputs at the terminal was actuated or is actuated (see also terminal documentation). If the input <i>bDoNotLockProcessImage</i> is not set, the output <i>bProcessImageInactive</i> is set and no further DALI commands can be sent via the bus terminal.
bProcessImageInactive	BOOL	One of the two digital inputs on the bus terminal was actuated and <i>bDoNotLockProcessImage</i> is initialized with FALSE. No further DALI commands can be sent from the PLC via the bus terminal. The blockage can be released again via the input <i>bResetInactiveProcessImage</i> .
bCollisionError	BOOL	A data collision was detected on the DALI bus while a DALI command was sent.
bPowerSupplyError	BOOL	When using the internal DALI power supply unit: Power supply unit fault detected.
bShortCircuit	BOOL	The 24 V DC supply voltage at connections 1 and 5 of the EL6821 is missing, or a short circuit has been detected on the DALI bus.
bInitialising	BOOL	During initialization of the bus terminal, the output is set and remains active until initialization has been completed. Initialization is also carried out automatically when the controller is started. During this time no DALI commands are processed.
bQuiescentModeActive	BOOL	If the EL6821 receives the DALI command START QUIESCENT MODE, this output is set to TRUE. This tells the PLC program that the Quiescent mode is active. This remains active for approx. 15 min or until the DALI command STOP QUIESCENT MODE has been received.
sTerminalDescription	STRING	Contains the Terminal Name (e.g. EL6821). This information is contained in CoE object 16#1008 of the bus terminal.
sFirmwareVersion	STRING	Contains the firmware version. This information is contained in CoE object 16#100A of the bus terminal.

 Properties

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.47	Tc3_DALI from v3.16.1.0

4.1.3.2 FB\_KL6811Communication



The function blocks for the DALI commands do not directly access the process image of the KL6811, but place the individual DALI commands in a command buffer. The function block FB\_KL6811Communication sequentially reads the DALI commands from the command buffer and forwards them to the KL6811. This prevents multiple function blocks accessing the KL6811 process image at the same time.



Most function blocks have the input *eCommandPriority*, which has no function when using the KL6811. Priorities for the DALI commands are only supported from DALI-2 onwards.

One instance of the function block FB\_KL6811Communication is required per KL6811. This instance must be called in a separate, faster task. This faster communication task must have a higher priority than the task in which the function blocks for the individual DALI commands are called.

The utilization rate of the command buffer can be determined using the outputs of the function block. If you find that the command buffer is overflowing regularly, you should take the following steps:

- How heavily are the individual PLC tasks utilized? TwinCAT provides suitable analysis tools.
- Try to reduce the cycle time of the task in which the function block FB\_KL6811Communication is called. The value should not be greater than 6 ms. The optimum value is 2 ms or less.
- If possible avoid polling (regular reading) of values. Only read values when they are actually required.
- Distribute the individual DALI control gears evenly over several DALI lines. Since several DALI lines are processed simultaneously in each PLC cycle, this increases the data throughput.

For more information, see the chapter [Bus Timing](#) [[▶ 12](#)].

 **Inputs**

```
VAR_INPUT
  bResetMaximumDemandCounter      : BOOL;
  bResetOverflowCounter            : BOOL;
  nOptions                         : DWORD;
  bInitialise                      : BOOL := FALSE;
  bDisableInternalPowerSupply      : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bResetMaximumDemandCounter	BOOL	A positive edge resets the stored value for the maximum utilization of the command buffer, <i>nBufferMaximumDemandMeter</i> (0...100%).
bResetOverflowCounter	BOOL	A positive edge resets the stored value for the number of overflows of the command buffer, <i>nBufferOverflowCounter</i> .
nOptions	DWORD	Reserved for future extensions.
bInitialise	BOOL	Configuration of the bus terminal is started by a positive edge at this input. Initialization is also carried out automatically when the controller is started. During this time no DALI commands are processed.
bDisableInternalPowerSupply	BOOL	Defines the operation mode of the internal DALI power supply. The value is written to the terminal by a positive edge at input bInitialise and stored there persistently.

 **Inputs/outputs**

```
VAR_IN_OUT
  stInData          : ST_KL6811InData;
  stOutData         : ST_KL6811OutData;
END_VAR
```

Name	Type	Description
stInData	<a href="#">ST_KL6811InData [► 853]</a>	Structure in the input process image of the KL6811. It is used for communication from the KL6811 to the PLC.
stOutData	<a href="#">ST_KL6811OutData [► 854]</a>	Structure in the output process image of the KL6811. It is used for communication from the PLC to the KL6811.

 **Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nBufferDemandMeter    : BYTE;
  nBufferMaximumDemandMeter : BYTE;
  nBufferOverflowCounter : UINT;
  bCollisionErrorFC     : BOOL;
  bCollisionErrorBC     : BOOL;
  bPowerSupplyError     : BOOL;
  bInitialising         : BOOL;
  nTerminalDescription  : WORD;
  nFirmwareVersion      : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nBufferDemandMeter	BYTE	Utilization rate of the command buffer (0...100 %).
nBufferMaximumDemandMeter	BYTE	Maximum utilization rate of the command buffer reached so far (0...100 %). The counter can be reset via the input <i>bResetMaximumDemandCounter</i> .
nBufferOverflowCounter	UINT	Number of command buffer overflows to date. The counter can be reset via the input <i>bResetOverflowCounter</i> .
bCollisionErrorFC	BOOL	If the output is TRUE, a DALI collision was detected on the forward channel. This can mean that a collision with the send data of another DALI control device was detected during the transmission of a DALI frame. The output is automatically set to FALSE again as soon as no collision has occurred on the forward channel during transmission.
bCollisionErrorBC	BOOL	If the output is TRUE, a DALI collision was detected on the backward channel. This can mean that a collision with the send data of another DALI device was detected during the transmission of a DALI frame. The output is automatically set to FALSE again as soon as no collision has occurred on the backward channel during transmission.
bPowerSupplyError	BOOL	If the output is TRUE, an overload was detected when using the internal DALI power supply unit of the bus terminal (bus undervoltage).
bInitialising	BOOL	During initialization of the bus terminal, the output is set and remains active until initialization has been completed. Initialization is also carried out automatically when the controller is started. During this time no DALI commands are processed.
nTerminalDescription	WORD	Contains the Terminal Name (e.g. 6811). This information is contained in register 8 of the bus terminal.
nFirmwareVersion	WORD	Contains the firmware version. This information is contained in register 9 of the bus terminal.

 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

### 4.1.3.3 FB\_KL6821Communication

FB_KL6821Communication	
— bResetMaximumDemandCounter <i>BOOL</i>	<i>BOOL</i> bError
— bResetOverflowCounter <i>BOOL</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
— nOptions <i>DWORD</i>	<i>BOOL</i> bBusy
— bResetInactiveProcessImage <i>BOOL</i>	<i>BYTE</i> nBufferDemandMeter
— bInitialise <i>BOOL</i>	<i>BYTE</i> nBufferMaximumDemandMeter
— eCommandKBusWatchdog <i>E_DALIConfigurationCommand</i>	<i>UINT</i> nBufferOverflowCounter
— eCommandDI1RisingEdge <i>E_DALIConfigurationCommand</i>	<i>BOOL</i> bDigitalInput1Active
— eCommandDI1FallingEdge <i>E_DALIConfigurationCommand</i>	<i>BOOL</i> bDigitalInput2Active
— eCommandDI2RisingEdge <i>E_DALIConfigurationCommand</i>	<i>BOOL</i> bProcessImageInactive
— eCommandDI2FallingEdge <i>E_DALIConfigurationCommand</i>	<i>BOOL</i> bCollisionError
— bDoNotLockProcessImage <i>BOOL</i>	<i>BOOL</i> bPowerSupplyError
— bDisableInternalPowerSupply <i>BOOL</i>	<i>BOOL</i> bShortCircuit
— stInData <i>ST_KL6821InData</i>	<i>BOOL</i> bInitialising
— stOutData <i>ST_KL6821OutData</i>	<i>WORD</i> nTerminalDescription
	<i>WORD</i> nFirmwareVersion

The function blocks for the DALI commands do not directly access the process image of the KL6821, but place the individual DALI commands in a command buffer. The function block FB\_KL6821Communication sequentially reads the DALI commands from the command buffer and forwards them to the KL6821. This prevents multiple function blocks accessing the KL6821 process image at the same time.

The function block also reads the events of the DALI control devices from the KL6821 and stores them in a special table. The function blocks [FB\\_DALIGetInputNotification \[► 841\]](#) and [FB\\_DALIGetPowerCycleNotification \[► 843\]](#) are used to filter out the desired events from this table.

One instance of the function block FB\_KL6821Communication must be created for each KL6821. This instance must be called in a separate, faster task. This faster communication task must have a higher priority than the task in which the function blocks for the individual DALI commands are called.

The utilization rate of the command buffer can be determined using the outputs of the function block. If you find that the command buffer is overflowing regularly, you should take the following steps:

- How heavily are the individual PLC tasks utilized? TwinCAT provides suitable analysis tools.
- Try to reduce the cycle time of the task in which the function block FB\_KL6821Communication is called. The value should not be greater than 6 ms. The optimum value is 2 ms or less.
- If possible avoid polling (regular reading) of values. Only read values when they are actually required.
- Distribute the individual DALI devices evenly over several DALI lines. Since several DALI lines are processed simultaneously in each PLC cycle, this increases the data throughput.

For more information, see the chapter [Bus Timing \[► 12\]](#).

#### Inputs

```

VAR_INPUT
  bResetMaximumDemandCounter : BOOL;
  bResetOverflowCounter      : BOOL;
  nOptions                   : DWORD := 0;
  bResetInactiveProcessImage : BOOL;
  bInitialise                : BOOL := FALSE;
  eCommandKBusWatchdog       : E_DALIConfigurationCommand := E_DALIConfigurationCommand.DoNothing;
  eCommandDI1RisingEdge      : E_DALIConfigurationCommand := E_DALIConfigurationCommand.Off;
  eCommandDI1FallingEdge     : E_DALIConfigurationCommand := E_DALIConfigurationCommand.DoNothing;
  eCommandDI2RisingEdge      : E_DALIConfigurationCommand := E_DALIConfigurationCommand.RecallMaxLev
el;
  eCommandDI2FallingEdge     : E_DALIConfigurationCommand := E_DALIConfigurationCommand.DoNothing;
  bDoNotLockProcessImage     : BOOL := FALSE;
  bDisableInternalPowerSupply: BOOL := FALSE;
END_VAR

```

Name	Type	Description
bResetMaximumDemandCounter	BOOL	A positive edge resets the stored value for the maximum utilization of the command buffer, <i>nBufferMaximumDemandMeter</i> (0...100 %).
bResetOverflowCounter	BOOL	A positive edge resets the stored value for the number of overflows of the command buffer, <i>nBufferOverflowCounter</i> .
nOptions	DWORD	Reserved for future extensions.
bResetInactiveProcessImage	BOOL	A positive edge cancels the blocking of the process image of the terminal. The output <i>bProcessImageInactive</i> is reset to FALSE. The blocking is activated as soon as one of the two digital inputs on the terminal has been actuated and the input parameter <i>bDoNotLockProcessImage</i> is FALSE.
bInitialise	BOOL	Configuration of the bus terminal is started by a positive edge at this input. Initialization is also carried out automatically when the controller is started. During this time no DALI commands are processed.
eCommandKBusWatchdog	E_DALIConfigurationCommand	Defines the DALI command that is sent as soon as the bus terminal is no longer addressed via the K-bus. The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.
eCommandDI1RisingEdge, eCommandDI2RisingEdge	E_DALIConfigurationCommand	Defines the DALI command that is sent as soon as a rising edge is detected at the respective input of the bus terminal. The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.
eCommandDI1FallingEdge, eCommandDI2FallingEdge	E_DALIConfigurationCommand	Defines the DALI command that is sent as soon as a falling edge is detected at the respective input of the bus terminal. The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.
bDoNotLockProcessImage	BOOL	Defines whether the process image for the PLC is not blocked by actuating the digital inputs (see also <i>bInactiveProcessImage</i> ). The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.
bDisableInternalPowerSupply	BOOL	Defines the operation mode of the internal DALI power supply unit. The value is written to the terminal by a positive edge at input <i>bInitialise</i> and stored there persistently.

 **Inputs/outputs**

```
VAR_IN_OUT
  stInData          : ST_KL6821InData;
  stOutData         : ST_KL6821OutData;
END_VAR
```

Name	Type	Description
stInData	ST_KL6821InData [ <a href="#">▶ 854</a> ]	Structure in the input process image of the KL6821. It is used for communication from the KL6821 to the PLC.
stOutData	ST_KL6821OutData [ <a href="#">▶ 854</a> ]	Structure in the output process image of the KL6821. It is used for communication from the PLC to the KL6821.

 **Outputs**

```
VAR_OUTPUT
  bError              : BOOL;
  ipResultMessage    : I_TcMessage;
  bBusy               : BOOL;
  nBufferDemandMeter : BYTE;
  nBufferMaximumDemandMeter : BYTE;
  nBufferOverflowCounter : UINT;
  bDigitalInputnActive : BOOL;
  bProcessImageInactive : BOOL;
```



```

bCollisionError      : BOOL;
bPowerSupplyError   : BOOL;
bShortCircuit       : BOOL;
bInitialising       : BOOL;
nTerminalDescription : WORD;
nFirmwareVersion    : WORD;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nBufferDemandMeter	BYTE	Utilization rate of the command buffer (0...100 %).
nBufferMaximumDemandMeter	BYTE	Maximum utilization rate of the command buffer reached so far (0...100 %). The counter can be reset via the input <i>bResetMaximumDemandCounter</i> .
nBufferOverflowCounter	UINT	Number of command buffer overflows to date. The counter can be reset via the input <i>bResetOverflowCounter</i> .
bDigitalInput1Active, bDigitalInput2Active	BOOL	One of the digital inputs at the terminal was actuated or is actuated (see also terminal documentation). If the input <i>bDoNotLockProcessImage</i> is not set, the output <i>bProcessImageInactive</i> is set and no further DALI commands can be sent via the bus terminal.
bProcessImageInactive	BOOL	One of the two digital inputs on the bus terminal was actuated and <i>bDoNotLockProcessImage</i> is initialized with FALSE. No further DALI commands can be sent from the PLC via the bus terminal. The blockage can be released again via the input <i>bResetInactiveProcessImage</i> .
bCollisionError	BOOL	A data collision was detected on the DALI bus while a DALI command was sent.
bPowerSupplyError	BOOL	When using the internal DALI power supply unit: Power supply unit fault detected.
bShortCircuit	BOOL	The 24 V DC supply voltage at connections 1 and 5 of the KL6821 is missing, or a short circuit has been detected on the DALI bus.
bInitialising	BOOL	During initialization of the bus terminal, the output is set and remains active until initialization has been completed. Initialization is also carried out automatically when the controller is started. During this time no DALI commands are processed.
nTerminalDescription	WORD	Contains the Terminal Name (e.g. 6821). This information is contained in register 8 of the bus terminal.
nFirmwareVersion	WORD	Contains the firmware version. This information is contained in register 9 of the bus terminal.



 **Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4 Devices**

All certified DALI-2 devices are listed in a product database on the homepage of the DALI Alliance (<https://www.dali-alliance.org/products>).

Each certified DALI-2 device has a unique GTIN (Global Trade Item Number). This GTIN can be read via memory bank 0 within the addresses 16#03 to 16#08 with the help of the function blocks [FB\\_DALI102ReadMemoryBank](#) [[▶ 47](#)] and [FB\\_DALI103ReadMemoryBank](#) [[▶ 52](#)]. Furthermore, the memory bank can also be read out with the DALI PLC Commissioning Tool (see [DALI PLC Commissioning Tool](#) [[▶ 882](#)]) and the KS2000 (see [Commissioning KL6821](#) [[▶ 879](#)]).

The devices that are supported by a function block in the library are listed below:

**B.E.G.**

Designation	Function block name	GTIN
Indoor 180-BMS DALI-2	<a href="#">FB_DALI_BEG_Indoor_180_BMS_V8</a> [ <a href="#">▶ 596</a> ]	4007529935406
LC-Mini 120-BMS DALI-2	<a href="#">FB_DALI_BEG_LC_Mini_120_BMS_V8</a> [ <a href="#">▶ 599</a> ]	4007529935413
PD11-BMS-FLAT DALI-2	<a href="#">FB_DALI_BEG_PD11_BMS_V8_Flat</a> [ <a href="#">▶ 602</a> ]	4007529935420
PD2N-BMS-FC DALI-2	<a href="#">FB_DALI_BEG_PD2N_BMS_V8_FC</a> [ <a href="#">▶ 605</a> ]	4007529935437
PD2N-BMS-FM DALI-2	<a href="#">FB_DALI_BEG_PD2N_BMS_V8_FM</a> [ <a href="#">▶ 608</a> ]	4007529935444
PD4N-BMS DALI-2	<a href="#">FB_DALI_BEG_PD4N_BMS_V8</a> [ <a href="#">▶ 611</a> ]	4007529935468
PD4-BMS-GH-SM DALI-2	<a href="#">FB_DALI_BEG_PD4_BMS_V8_GH_SM</a> [ <a href="#">▶ 614</a> ]	4007529935451
PICO-BMS DALI-2	<a href="#">FB_DALI_BEG_PICO_BMS_V8</a> [ <a href="#">▶ 617</a> ]	4007529935475

**Esylux**

Designation	Function block name	GTIN
PD-C 360/8 BMS DALI-2	<a href="#">FB_DALI_Esylux_PD_C_360_BMS</a> [ <a href="#">▶ 620</a> ]	4015120428203
PD-C 360/24 BMS DALI-2	<a href="#">FB_DALI_Esylux_PD_C_360_BMS</a> [ <a href="#">▶ 620</a> ]	4015120428210
PD-C 360/32 BMS DALI-2	<a href="#">FB_DALI_Esylux_PD_C_360_BMS</a> [ <a href="#">▶ 620</a> ]	4015120424885
PD-C 360/8 mini BMS DALI-2	<a href="#">FB_DALI_Esylux_PD_C_360_Mini_BMS</a> [ <a href="#">▶ 624</a> ]	4015120423048
PD-FLAT 360/8 RW BMS DALI-2	<a href="#">FB_DALI_Esylux_PD_FLAT_360_BMS</a> [ <a href="#">▶ 628</a> ]	4015120423031
PD-FLAT 360/8 SW BMS DALI-2	<a href="#">FB_DALI_Esylux_PD_FLAT_360_BMS</a> [ <a href="#">▶ 628</a> ]	4015120423024
PD-FLAT-L 360/8 RW BMS DALI-2	<a href="#">FB_DALI_Esylux_PD_FLAT_360_BMS</a> [ <a href="#">▶ 628</a> ]	4015120423055

**Glamox**

Designation	Function block name	GTIN
LMS DALI-2 PB8 SWITCH PANEL	FB_DALI_Glamox_LMS_PB8_SwitchPanel [ <a href="#">▶ 632</a> ]	7020940423762

**Helvar**

Designation	Function block name	GTIN
142WD2 2-Button Module white	FB_DALI_Helvar_2ButtonModule [ <a href="#">▶ 635</a> ]	5030797801424
137WD2 4-Button Module white	FB_DALI_Helvar_4ButtonModule [ <a href="#">▶ 638</a> ]	5030797821378
137BD2 4-Button Module black	FB_DALI_Helvar_4ButtonModule [ <a href="#">▶ 638</a> ]	5030797811379
144WD2 4-Button Module White	FB_DALI_Helvar_4ButtonModule [ <a href="#">▶ 638</a> ]	5030797801448
134WD2 5-Button Module white	FB_DALI_Helvar_5ButtonModule [ <a href="#">▶ 641</a> ]	5030797821347
134BD2 5-Button Module black	FB_DALI_Helvar_5ButtonModule [ <a href="#">▶ 641</a> ]	5030797811348
146WD2 6-Button Module white	FB_DALI_Helvar_6ButtonModule [ <a href="#">▶ 644</a> ]	5030797801462
135WD2 7-Button Module white	FB_DALI_Helvar_7ButtonModule [ <a href="#">▶ 647</a> ]	5030797821354
135BD2 7-Button Module black	FB_DALI_Helvar_7ButtonModule [ <a href="#">▶ 647</a> ]	5030797811355
136WD2 8-Button Module white	FB_DALI_Helvar_8ButtonModule [ <a href="#">▶ 650</a> ]	5030797821361
136BD2 8-Button Module black	FB_DALI_Helvar_8ButtonModule [ <a href="#">▶ 650</a> ]	5030797811362
148WD2 8-Button Module white	FB_DALI_Helvar_8ButtonModule [ <a href="#">▶ 650</a> ]	5030797801486

**Loytec**

Designation	Function block name	GTIN
LDALI-BM2	FB_DALI_Loytec_LDALI_BM2 [ <a href="#">▶ 653</a> ]	4710901730512
LDALI-MS4-BT	FB_DALI_Loytec_LDALI_MS4_BT [ <a href="#">▶ 663</a> ]	4710901730741
LDALI-MS2-BT	FB_DALI_Loytec_LDALI_MS2 [ <a href="#">▶ 657</a> ]	4710901730734
LDALI-MS2	FB_DALI_Loytec_LDALI_MS2 [ <a href="#">▶ 657</a> ]	4710901730505

**Lunatone**

Designation	Function block name	GTIN
DALI-2 CS Integration	FB_DALI_Lunatone_CS [ <a href="#">▶ 667</a> ]	9010342013546
DALI-2 CS Integration THP	FB_DALI_Lunatone_CS_THP [ <a href="#">▶ 671</a> ]	9010342013546
DALI-2 MC	FB_DALI_Lunatone_MC [ <a href="#">▶ 675</a> ]	9010342012730
DALI-2 MC4L	FB_DALI_Lunatone_MC4L [ <a href="#">▶ 677</a> ]	9010342012754
DALI-2 Cross Switch	FB_DALI_Lunatone_Cross_Switch [ <a href="#">▶ 680</a> ]	9010342010224

**Niko**

Designation	Function block name	GTIN
P46LR for surface mounting	FB_DALI_Niko_P46LR [ <a href="#">▶ 683</a> ]	5413736374195
P46LR for SnapFit flush mounting	FB_DALI_Niko_P46LR [ <a href="#">▶ 683</a> ]	5413736374164
P46LR for flush-mounting box	FB_DALI_Niko_P46LR [ <a href="#">▶ 683</a> ]	5413736374140
P46MR for surface mounting	FB_DALI_Niko_P46MR [ <a href="#">▶ 687</a> ]	5413736374393
P46MR for SnapFit flush mounting	FB_DALI_Niko_P46MR [ <a href="#">▶ 687</a> ]	5413736374362
P46MR for flush-mounting box	FB_DALI_Niko_P46MR [ <a href="#">▶ 687</a> ]	5413736374348

## Osram

Designation	Function block name	GTIN
DALI COUPLER E G2	FB_DALI_Osram_Coupler_E_G2 [ <a href="#">▶ 690</a> ]	4062172072212
DALI COUPLER HF G2	FB_DALI_Osram_Coupler_HF_G2 [ <a href="#">▶ 693</a> ]	4062172072199
DALI COUPLER LS HIGHBAY G2	FB_DALI_Osram_Coupler_LS_Highbay_G2 [ <a href="#">▶ 696</a> ]	4062172072137
DALI COUPLER MULTI3 G2	FB_DALI_Osram_Coupler_Multi3_G2 [ <a href="#">▶ 699</a> ]	4062172072113
DALI COUPLER PB G2	FB_DALI_Osram_Coupler_PushB_G2 [ <a href="#">▶ 702</a> ]	4062172087575
LS/PD CI G2	FB_DALI_Osram_LS_PD_CI_G2 [ <a href="#">▶ 704</a> ]	4062172072069
LS/PD LI G2	FB_DALI_Osram_LS_PD_LI_G2 [ <a href="#">▶ 708</a> ]	4062172072069
LS/PD LI UF G2	FB_DALI_Osram_LS_PD_LI_UF_G2 [ <a href="#">▶ 711</a> ]	4062172072045

**Steinel**

<b>Designation</b>	<b>Function block name</b>	<b>GTIN</b>
Dual HF	FB_DALI_Steinel_ControlProDualHF [ <a href="#">▶ 714</a> ]	4007841057459
Dual Tech US	FB_DALI_Steinel_ControlProDualTech [ <a href="#">▶ 717</a> ]	4007841057473
Dual US	FB_DALI_Steinel_ControlProDualUS [ <a href="#">▶ 720</a> ]	4007841057466
IR Quattro HD	FB_DALI_Steinel_ControlProIRQuattroHD [ <a href="#">▶ 723</a> ]	4007841057497
Single US	FB_DALI_Steinel_ControlProSingleUS [ <a href="#">▶ 726</a> ]	4007841057503
US 360	FB_DALI_Steinel_ControlProUS360 [ <a href="#">▶ 729</a> ]	4007841057534
Dual Light Sensor AP	FB_DALI_Steinel_DualLightSensor [ <a href="#">▶ 732</a> ]	4007841066291
Dual Light Sensor AP square shape	FB_DALI_Steinel_DualLightSensor [ <a href="#">▶ 732</a> ]	4007841057411
Dual Light Sensor AP round shape	FB_DALI_Steinel_DualLightSensor [ <a href="#">▶ 732</a> ]	4007841057428
Dual Light Sensor UP	FB_DALI_Steinel_DualLightSensor [ <a href="#">▶ 732</a> ]	4007841066369
Dual Light Sensor UP square shape	FB_DALI_Steinel_DualLightSensor [ <a href="#">▶ 732</a> ]	4007841057435
Dual Light Sensor UP round shape	FB_DALI_Steinel_DualLightSensor [ <a href="#">▶ 732</a> ]	4007841057442
Hallway IPD UP	FB_DALI_Steinel_Hallway_IPD [ <a href="#">▶ 734</a> ]	4007841064532
Hallway IPD AP	FB_DALI_Steinel_Hallway_IPD [ <a href="#">▶ 734</a> ]	4007841064549
Hallway IPD ECO UP	FB_DALI_Steinel_Hallway_IPD_ECO [ <a href="#">▶ 740</a> ]	4007841079048
Hallway IPD ECO AP	FB_DALI_Steinel_Hallway_IPD_ECO [ <a href="#">▶ 740</a> ]	4007841079031
HF 3360 AP	FB_DALI_Steinel_HF3360 [ <a href="#">▶ 745</a> ]	4007841066284
HF 3360 AP square shape	FB_DALI_Steinel_HF3360 [ <a href="#">▶ 745</a> ]	4007841057312
HF 3360 AP round shape	FB_DALI_Steinel_HF3360 [ <a href="#">▶ 745</a> ]	4007841057329
HF 3360 UP	FB_DALI_Steinel_HF3360 [ <a href="#">▶ 745</a> ]	4007841066352
HF 3360 UP square shape	FB_DALI_Steinel_HF3360 [ <a href="#">▶ 745</a> ]	4007841057336
HF 3360 UP round shape	FB_DALI_Steinel_HF3360 [ <a href="#">▶ 745</a> ]	4007841057343
HF 360	FB_DALI_Steinel_HF360 [ <a href="#">▶ 748</a> ]	4007841057480
HF 360 II IPD UP	FB_DALI_Steinel_HF360_II_IPD [ <a href="#">▶ 751</a> ]	4007841064280
HF 360 II IPD AP	FB_DALI_Steinel_HF360_II_IPD [ <a href="#">▶ 751</a> ]	4007841064297
HF 360 II IPD ECO UP	FB_DALI_Steinel_HF360_II_IPD_ECO [ <a href="#">▶ 755</a> ]	4007841067465
HF 360 II IPD ECO AP	FB_DALI_Steinel_HF360_II_IPD_ECO [ <a href="#">▶ 755</a> ]	4007841067458
IR Micro	FB_DALI_Steinel_IRMicro [ <a href="#">▶ 758</a> ]	4007841057732
IR Quattro Micro	FB_DALI_Steinel_IRQuattroMicro [ <a href="#">▶ 761</a> ]	4007841053871
IR Quattro Slim XS	FB_DALI_Steinel_IRQuattroSlimXS [ <a href="#">▶ 764</a> ]	4007841065034
IR Quattro Slim XS square shape	FB_DALI_Steinel_IRQuattroSlimXS [ <a href="#">▶ 764</a> ]	4007841057510
IR Quattro Slim XS round shape	FB_DALI_Steinel_IRQuattroSlimXS [ <a href="#">▶ 764</a> ]	4007841057527
IS 345 AP	FB_DALI_Steinel_IS345 [ <a href="#">▶ 767</a> ]	4007841066253
IS 345 AP square shape	FB_DALI_Steinel_IS345 [ <a href="#">▶ 767</a> ]	4007841057275
IS 345 AP round shape	FB_DALI_Steinel_IS345 [ <a href="#">▶ 767</a> ]	4007841057299
IS 345 UP	FB_DALI_Steinel_IS345 [ <a href="#">▶ 767</a> ]	4007841066321
IS 345 UP square shape	FB_DALI_Steinel_IS345 [ <a href="#">▶ 767</a> ]	4007841057282
IS 345 UP round shape	FB_DALI_Steinel_IS345 [ <a href="#">▶ 767</a> ]	4007841057305
IS 345 MX AP	FB_DALI_Steinel_IS345MX [ <a href="#">▶ 770</a> ]	4007841066260
IS 345 MX AP square shape	FB_DALI_Steinel_IS345MX [ <a href="#">▶ 770</a> ]	4007841057398

Designation	Function block name	GTIN
IS 345 MX UP	FB_DALI_Steinel_IS345MX [ <a href="#">▶ 770</a> ]	4007841066338
IS 3180 AP	FB_DALI_Steinel_IS3180 [ <a href="#">▶ 773</a> ]	4007841066277
IS 3180 AP square shape	FB_DALI_Steinel_IS3180 [ <a href="#">▶ 773</a> ]	4007841057350
IS 3180 UP	FB_DALI_Steinel_IS3180 [ <a href="#">▶ 773</a> ]	4007841066345
IS 3180 UP square shape	FB_DALI_Steinel_IS3180 [ <a href="#">▶ 773</a> ]	4007841057367
IS 3360 AP	FB_DALI_Steinel_IS3360 [ <a href="#">▶ 776</a> ]	4007841066239
IS 3360 AP square shape	FB_DALI_Steinel_IS3360 [ <a href="#">▶ 776</a> ]	4007841057237
IS 3360 AP round shape	FB_DALI_Steinel_IS3360 [ <a href="#">▶ 776</a> ]	4007841057251
IS 3360 UP	FB_DALI_Steinel_IS3360 [ <a href="#">▶ 776</a> ]	4007841066307
IS 3360 UP square shape	FB_DALI_Steinel_IS3360 [ <a href="#">▶ 776</a> ]	4007841057244
IS 3360 UP round shape	FB_DALI_Steinel_IS3360 [ <a href="#">▶ 776</a> ]	4007841057268
IS 3360 MX AP	FB_DALI_Steinel_IS3360MX [ <a href="#">▶ 779</a> ]	4007841066246
IS 3360 MX AP square shape	FB_DALI_Steinel_IS3360MX [ <a href="#">▶ 779</a> ]	4007841057374
IS 3360 MX UP	FB_DALI_Steinel_IS3360MX [ <a href="#">▶ 779</a> ]	4007841066314
IS 3360 MX UP square shape	FB_DALI_Steinel_IS3360MX [ <a href="#">▶ 779</a> ]	4007841057381

### Sunricher

Designation	Function block name	GTIN
Push Button Coupler	FB_DALI_Sunricher_PushButtonCoupler [ <a href="#">▶ 782</a> ]	6971542121819

### Theben

Designation	Function block name	GTIN
PlanoSpot 360	FB_DALI_Theben_PlanoSpot_360 [ <a href="#">▶ 785</a> ]	4003468200360
thePassa P360	FB_DALI_Theben_thePassa_P360 [ <a href="#">▶ 789</a> ]	4003468200353
theRonda P360	FB_DALI_Theben_theRonda_P360 [ <a href="#">▶ 794</a> ]	4003468200353
theRonda S360	FB_DALI_Theben_theRonda_S360 [ <a href="#">▶ 799</a> ]	4003468200384
TA 4 S DALI-2	FB_DALI_Theben_TA_4_S [ <a href="#">▶ 803</a> ]	4003468491584

### Tridonic

Designation	Function block name	GTIN
MSensor G3 SRC 30 PIR 5DPI	FB_DALI_Triconic_MSensorG3_SRC_30_PIR [ <a href="#">▶ 812</a> ]	9006210666693
MSensor G3 SFI 30 PIR 5DPI BK	FB_DALI_Triconic_MSensorG3_SFI_30_PIR [ <a href="#">▶ 808</a> ]	9006210672540
MSensor G3 SFI 30 PIR 5DPI WH	FB_DALI_Triconic_MSensorG3_SFI_30_PIR [ <a href="#">▶ 808</a> ]	9006210672502
MSensor G3 SFI 30 PIR 16DPI WH	FB_DALI_Triconic_MSensorG3_SFI_30_PIR [ <a href="#">▶ 808</a> ]	9006210656205
MSensor G3 SFI 30 PIR 10DPI WH	FB_DALI_Triconic_MSensorG3_SFI_30_PIR [ <a href="#">▶ 808</a> ]	9006210648149
DALI XC G3 CWM 30 DA2	FB_DALI_Triconic_XC_G3_CWM30 [ <a href="#">▶ 816</a> ]	9006210708881

**Zencontrol**

Designation	Function block name	GTIN
PIR	<a href="#">FB_DALI_Zencontrol_PIR [▶ 819]</a>	9336462199091
Scenepanel Switch	<a href="#">FB_DALI_Zencontrol_Scenepanel_Switch [▶ 822]</a>	6971103530319
Switch 1 Button	<a href="#">FB_DALI_Zencontrol_Switch_1Button [▶ 824]</a>	9336462191897

As an alternative to the device function blocks, individual instances of DALI control devices can be addressed via the general function blocks. Each of the general function blocks refers to a specific part of IEC 62386.

IEC 62386-XXX	Function	Function block name
103	General properties of the control devices	<a href="#">FB_DALI103ControlDevice [▶ 826]</a>
301	Push button	<a href="#">FB_DALI301PushButton [▶ 830]</a>
302	Absolute encoder	<a href="#">FB_DALI302AbsoluteInputDevice [▶ 832]</a>
303	Occupancy sensor	<a href="#">FB_DALI303OccupancySensor [▶ 835]</a>
304	Light sensor	<a href="#">FB_DALI304LightSensor [▶ 837]</a>

Further function blocks to simplify the application of device function blocks

Function block name	Function
<a href="#">FB_DALI_ToPushButtonState [▶ 840]</a>	Determines the static state from the events of a push button.

**4.1.4.1 Introduction**

**Device function blocks**

Various manufacturers offer devices that combine several DALI parts in one device.

For simple integration into projects, the Tc3\_DALI library contains a constantly growing number of function blocks for certified DALI-2 devices. Additional information on the certification of DALI-2 devices can be found on the homepage of the DALI Alliance ([www.dali-alliance.org](http://www.dali-alliance.org)).

Individual manufacturers frequently offer further information on the devices that describe each parameter in more detail.

All parameters of a DALI control unit are mapped on the device function block via properties and can be written to the DALI control unit.

The properties contain internally specified default values. Therefore, it is not absolutely necessary to change the properties of a device function block.

For additional information on properties, see the chapter [Use of properties \[▶ 890\]](#).

## 4.1.4.2 B.E.G.

### 4.1.4.2.1 FB\_DALI\_BEG\_Indoor\_180\_BMS\_V8

FB_DALI_BEG_Indoor_180_BMS_V8	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryOccupancy <i>BOOL</i>	<i>BOOL</i> bReadingOccupancy
bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
bCancelHoldTimerOccupancy <i>BOOL</i>	<i>BOOL</i> bOccupied
	<i>UINT</i> nBrightnessLevel

The function block represents the Indoor 180-BMS DALI-2 DALI sensor from B.E.G..

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```



Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

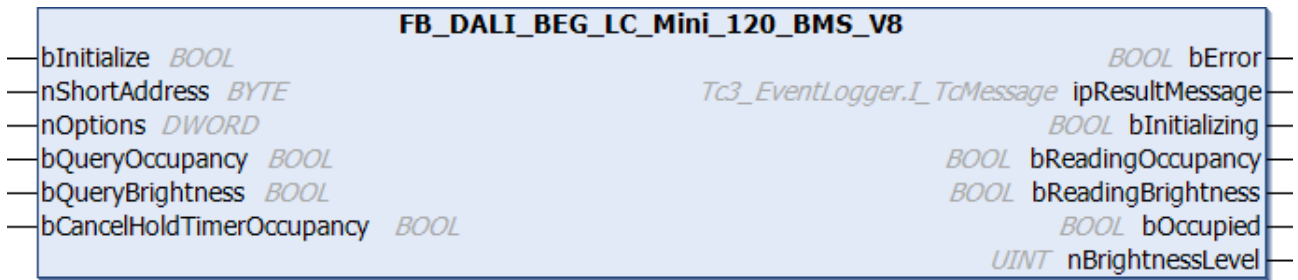
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

4.1.4.2.2 FB\_DALI\_BEG\_LC\_Mini\_120\_BMS\_V8



The function block represents the LC-Mini 120-BMS DALI-2 DALI sensor from B.E.G..

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

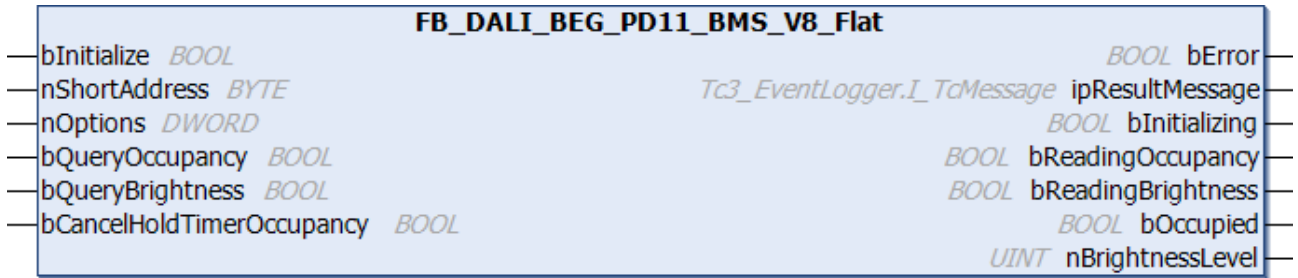
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

### 4.1.4.2.3 FB\_DALI\_BEG\_PD11\_BMS\_V8\_Flat



The function block represents the PD11-BMS-FLAT DALI-2 DALI sensor from B.E.G.

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL := FALSE;
  bQueryBrightness    : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).



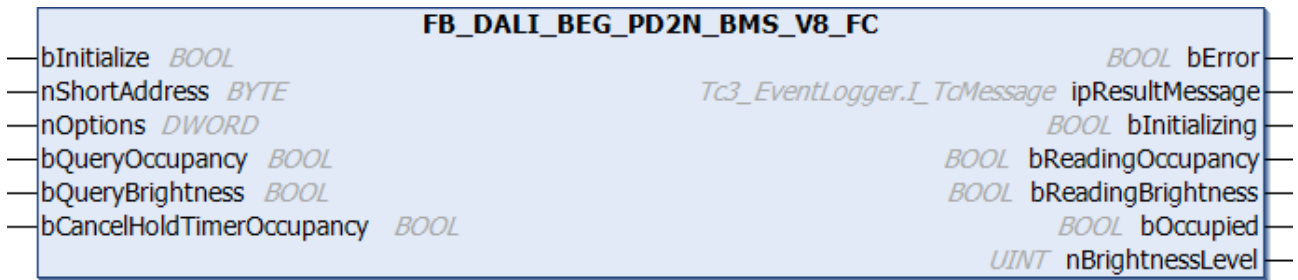
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0



4.1.4.2.4 FB\_DALI\_BEG\_PD2N\_BMS\_V8\_FC



The function block represents the PD2N-BMS-FC DALI-2 DALI sensor from B.E.G..

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[▶ 542\]](#)

Part 304 (light sensors) – [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy       : BOOL  := FALSE;
  bQueryBrightness      : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

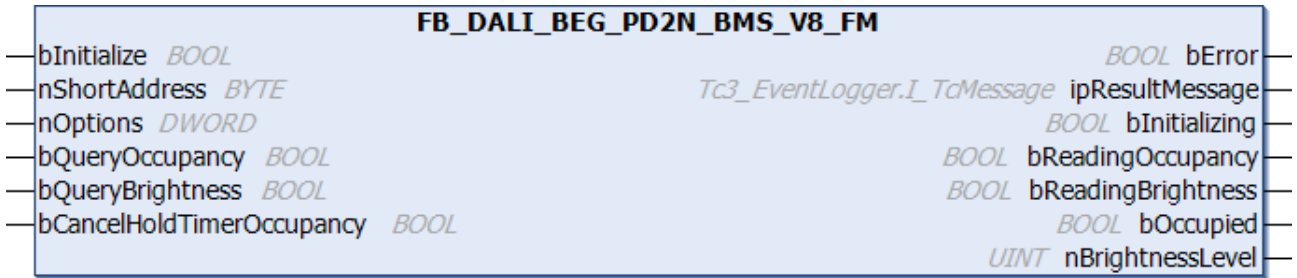
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

### 4.1.4.2.5 FB\_DALI\_BEG\_PD2N\_BMS\_V8\_FM



The function block represents the PD2N-BMS-V8-FM DALI-2 DALI sensor from B.E.G..

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```

VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
    
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

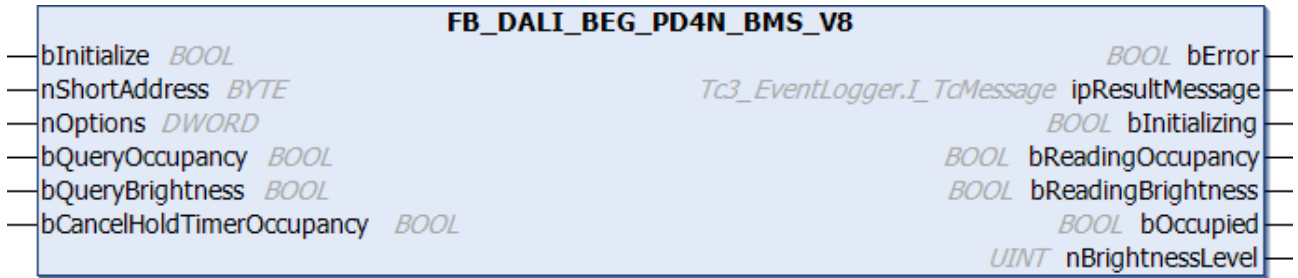
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

4.1.4.2.6 FB\_DALI\_BEG\_PD4N\_BMS\_V8



The function block represents the PD4N-BMS DALI-2 DALI sensor from B.E.G..

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

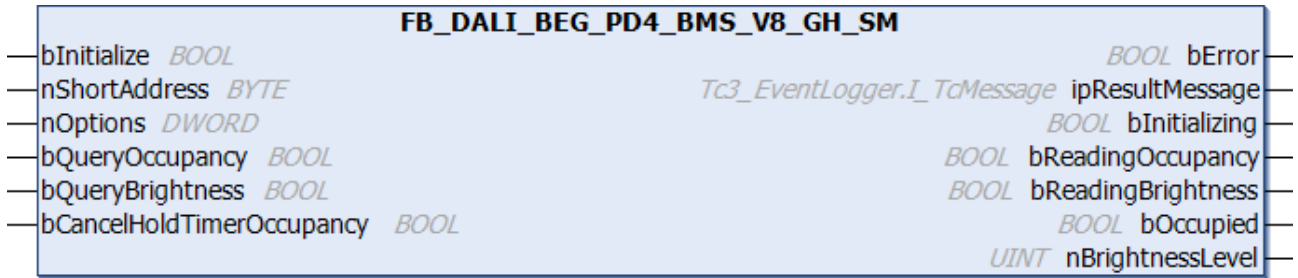


Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

### 4.1.4.2.7 FB\_DALI\_BEG\_PD4\_BMS\_V8\_GH\_SM



The function block represents the PD4-BMS-GH-SM DALI-2 DALI sensor from B.E.G..

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

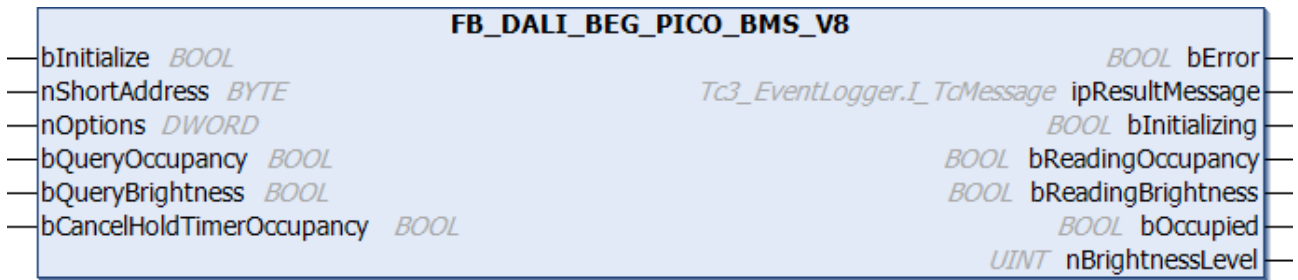
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

4.1.4.2.8 FB\_DALI\_BEG\_PICO\_BMS\_V8



The function block represents the PICO-BMS DALI-2 DALI sensor from B.E.G..

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at `bInitialize` and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (`bError = FALSE`). If the output `bError = TRUE`, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables `bQueryOccupancy` and `bQueryBrightness`, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress       : BYTE := 0;
  nOptions            : DWORD := 0;
  bQueryOccupancy    : BOOL := FALSE;
  bQueryBrightness   : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0



### 4.1.4.3 Esylux

#### 4.1.4.3.1 FB\_DALI\_Esylux\_PD\_C\_360\_BMS

FB_DALI_Esylux_PD_C_360_BMS	
– bInitialize <i>BOOL</i>	<i>BOOL</i> bError
– nShortAddress <i>USINT</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
– nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
– bQueryPushButton1 <i>BOOL</i>	<i>BOOL</i> bReadingPushButton1
– bQueryPushButton2 <i>BOOL</i>	<i>BOOL</i> bReadingPushButton2
– bQueryOccupancy <i>BOOL</i>	<i>BOOL</i> bReadingOccupancy
– bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
– bCancelHoldTimerOccupancy <i>BOOL</i>	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event
	<i>BOOL</i> bOccupied
	<i>UINT</i> nBrightnessLevel

The function block FB\_DALI\_Esylux\_PD\_C\_360\_BMS represents the DALI sensors PD-C 360/8 BMS DALI-2, PD-C 360/24 BMS DALI-2 and PD-C 360/32 BMS DALI-2 from Esylux.

This outputs two push button signals (instances 0 to 1), an occupancy signal (instance 2) and a measured brightness value (instance 3).

Further information on the supported instance types can be found here:

Part 301 (push buttons) – [Introduction \[► 511\]](#)

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryPushButtonN*, *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### ALS correction factor

During commissioning, it may be necessary to calibrate the brightness measurement. The ALS (ambient light sensor) parameters required for this can be changed via the properties. The function block transmits the individual ALS parameters (0.1...5.0) to the light sensor integrated in the DALI control device, where they are also stored. The correction factor of a light sensor is determined at 100 and 500 lux on the floor. For details on calibration, please refer to the vendor's documentation.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress       : USINT := 0;
```



```
nOptions          : DWORD := 0;
bQueryPushButton1 : BOOL  := FALSE;
bQueryPushButton2 : BOOL  := FALSE;
bQueryOccupancy   : BOOL  := FALSE;
bQueryBrightness  : BOOL  := FALSE;
bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	USINT	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryPushButtonN	BOOL	The state of the push button is queried immediately by a positive edge at this input.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
bError          : BOOL;
ipResultMessage : I_TcMessage;
bInitializing    : BOOL;
bReadingPushButton1 : BOOL;
bReadingPushButton2 : BOOL;
bReadingOccupancy : BOOL;
bReadingBrightness : BOOL;
nPushButton1Event : WORD;
nPushButton2Event : WORD;
bOccupied       : BOOL;
nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingPushButtonN	BOOL	The output is TRUE as long as values of the instance of the push button are being read by the DALI control device.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the motion sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
nPushButtonNEvent	WORD	The output outputs the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block `FB_DALI_ToPushButtonState` [► 840] can be used to convert the state of a push button into a BOOL variable.

Name	Type	Description
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.



### Properties

All parameters that are written to the DALI control device via `bInitialize` are available as properties [► 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnablePushButtonN	BOOL	Get, Set	FALSE	Push button input 1 or 2, which can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
fALS0CorrectionFactor100	LREAL	Get, Set	1.0	Correction factor (0.1...5.0) of light sensor 0 at a measured value of 100 lux on the surface to be measured.
fALS0CorrectionFactor500	LREAL	Get, Set	1.0	Correction factor (0.1...5.0) of light sensor 0 at a measured value of 500 lux on the surface to be measured.
fALS1CorrectionFactor100	LREAL	Get, Set	1.0	Correction factor (0.1...5.0) of light sensor 1 at a measured value of 100 lux on the surface to be measured.
fALS1CorrectionFactor500	LREAL	Get, Set	1.0	Correction factor (0.1...5.0) of light sensor 1 at a measured value of 500 lux on the surface to be measured.
nReportTimerBrightness	USINT	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	USINT	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	USINT	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.

Name	Type	Access	Initial value	Description
nReportTimerOccupancy	USINT	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	USINT	Get, Set	95	Adjustment of the sensitivity of the occupancy sensor instance from 0 (low sensitivity) to 100 (high sensitivity).
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <a href="#">event filter</a> [► 232].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <a href="#">event priority</a> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	USINT	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
bModeLED	BOOL	Get, Set	TRUE	LED mode: FALSE: LED disabled TRUE: LED enabled
nBrightnessLED	USINT	Get, Set	50	LED brightness (5...100 %).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.50	Tc3_DALI from v3.17.1.0

**4.1.4.3.2 FB\_DALI\_Esylux\_PD\_C\_360\_Mini\_BMS**

FB_DALI_Esylux_PD_C_360_Mini_BMS	
– bInitialize <i>BOOL</i>	<i>BOOL</i> bError
– nShortAddress <i>USINT</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
– nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
– bQueryOccupancy <i>BOOL</i>	<i>BOOL</i> bReadingOccupancy
– bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
– bCancelHoldTimerOccupancy <i>BOOL</i>	<i>BOOL</i> bOccupied
	<i>UINT</i> nBrightnessLevel

The function block FB\_DALI\_Esylux\_PD\_C\_360\_Mini\_BMS represents the DALI sensor PD-C 360/8 mini BMS DALI-2 from Esylux.

This outputs an occupancy signal (instance 0) and the measured brightness value (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction](#) [► 542]

Part 304 (light sensors) – [Introduction](#) [► 561]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**ALS correction factor**

During commissioning, it may be necessary to calibrate the brightness measurement. The ALS (ambient light sensor) parameters required for this can be changed via the properties. The function block transmits the individual ALS parameters (0.1...5.0) to the light sensor integrated in the DALI control device, where they are also stored. The correction factor of a light sensor is determined at 100 and 500 lux on the floor. For details on calibration, please refer to the vendor's documentation.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : USINT := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	USINT	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingOccupancy     : BOOL;
  bReadingBrightness    : BOOL;
  bOccupied             : BOOL;
  nBrightnessLevel      : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the motion sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
fALSCorrectionFactor100	LREAL	Get, Set	1.0	Correction factor (0.1...5.0) of light sensor at a measured value of 100 lux on the surface to be measured.
fALSCorrectionFactor500	LREAL	Get, Set	1.0	Correction factor (0.1...5.0) of light sensor at a measured value of 500 lux on the surface to be measured.
nReportTimerBrightness	USINT	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	USINT	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	USINT	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	USINT	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	USINT	Get, Set	95	Adjustment of the sensitivity of the occupancy sensor instance from 0 (low sensitivity) to 100 (high sensitivity).
bModeLED	BOOL	Get, Set	TRUE	LED mode: FALSE: LED disabled TRUE: LED enabled
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).



## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.50	Tc3_DALI from v3.17.1.0

## 4.1.4.3.3 FB\_DALI\_Esylux\_PD\_FLAT\_360\_BMS

FB_DALI_Esylux_PD_FLAT_360_BMS	
– bInitialize <i>BOOL</i>	<i>BOOL</i> bError
– nShortAddress <i>USINT</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
– nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
– bQueryPushButton <i>BOOL</i>	<i>BOOL</i> bReadingPushButton
– bQueryOccupancy <i>BOOL</i>	<i>BOOL</i> bReadingOccupancy
– bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
– bCancelHoldTimerOccupancy <i>BOOL</i>	<i>WORD</i> nPushButtonEvent
	<i>BOOL</i> bOccupied
	<i>UINT</i> nBrightnessLevel

The function block FB\_DALI\_Esylux\_PD\_FLAT\_360\_BMS represents the DALI sensors PD-FLAT 360/8 RW BMS DALI-2, PD-FLAT 360/8 SW BMS DALI-2 and PD-FLAT-L 360/8 RW BMS DALI-2 from Esylux.

This outputs a push button signal (instance 0), an occupancy signal (instance 1) and the measured brightness value (instance 2).

Further information on the supported instance types can be found here:

Part 301 (push buttons) – [Introduction \[► 511\]](#)

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryPushButton*, *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

## ALS correction factor

During commissioning, it may be necessary to calibrate the brightness measurement. The ALS (ambient light sensor) parameters required for this can be changed via the properties. The function block transmits the individual ALS parameters (0.1...5.0) to the light sensor integrated in the DALI control device, where they are also stored. The correction factor of a light sensor is determined at 100 and 500 lux on the floor. For details on calibration, please refer to the vendor's documentation.



 **Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : USINT := 0;
  nOptions             : DWORD := 0;
  bQueryPushButton    : BOOL := FALSE;
  bQueryOccupancy     : BOOL := FALSE;
  bQueryBrightness    : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	USINT	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryPushButton	BOOL	The state of the push button is queried immediately by a positive edge at this input.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError               : BOOL;
  ipResultMessage     : I_TcMessage;
  bInitializing       : BOOL;
  bReadingPushButton  : BOOL;
  bReadingOccupancy  : BOOL;
  bReadingBrightness  : BOOL;
  nPushButtonEvent    : WORD;
  bOccupied           : BOOL;
  nBrightnessLevel    : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingPushButton	BOOL	The output is TRUE as long as values of the instance of the push button are being read by the DALI control device.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the motion sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
nPushButtonEvent	WORD	The output outputs the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block `FB_DALI_ToPushButtonState` [► 840] can be used to convert the state of a push button into a BOOL variable.

Name	Type	Description
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.



### Properties

All parameters that are written to the DALI control device via `bInitialize` are available as properties [► 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnablePushButton	BOOL	Get, Set	FALSE	Push button input 1 or 2, which can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
fALSCorrectionFactor100	LREAL	Get, Set	1.0	Correction factor (0.1...5.0) of light sensor at a measured value of 100 lux on the surface to be measured.
fALSCorrectionFactor500	LREAL	Get, Set	1.0	Correction factor (0.1...5.0) of light sensor at a measured value of 500 lux on the surface to be measured.
nReportTimerBrightness	USINT	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	USINT	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	USINT	Get, Set	40	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	USINT	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	USINT	Get, Set	95	Adjustment of the sensitivity of the occupancy sensor instance from 0 (low sensitivity) to 100 (high sensitivity).
nEventFilterPushButton	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 232].

Name	Type	Access	Initial value	Description
eEventPriorityPushButton	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButton	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButton	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButton	USINT	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
bModeLED	BOOL	Get, Set	TRUE	LED mode: FALSE: LED disabled TRUE: LED enabled
nBrightnessLED	USINT	Get, Set	50	LED brightness (5...100 %).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.50	Tc3_DALI from v3.17.1.0

**4.1.4.4 Glamox**

**4.1.4.4.1 FB\_DALI\_Glamox\_LMS\_PB8\_SwitchPanel**

FB_DALI_Glamox_LMS_PB8_SwitchPanel	
— bInitialize <i>BOOL</i>	<i>BOOL</i> bError
— nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
— nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
— bQueryInputValue1 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue1
— bQueryInputValue2 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue2
— bQueryInputValue3 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue3
— bQueryInputValue4 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue4
— bQueryInputValue5 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue5
— bQueryInputValue6 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue6
— bQueryInputValue7 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue7
— bQueryInputValue8 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue8
	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event
	<i>WORD</i> nPushButton3Event
	<i>WORD</i> nPushButton4Event
	<i>WORD</i> nPushButton5Event
	<i>WORD</i> nPushButton6Event
	<i>WORD</i> nPushButton7Event
	<i>WORD</i> nPushButton8Event

The function block represents the LMS DALI-2 PB8 Switch Panel DALI push button interface from Glamox. Up to eight push buttons (instances 0 to 7) can be connected directly via the device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize      : BOOL := FALSE;
  nShortAddress    : BYTE := 0;
  nOptions         : DWORD := 0;
  bQueryInputValue1 : BOOL := FALSE;
  bQueryInputValue2 : BOOL := FALSE;
  bQueryInputValue3 : BOOL := FALSE;
  bQueryInputValue4 : BOOL := FALSE;
  bQueryInputValue5 : BOOL := FALSE;
  bQueryInputValue6 : BOOL := FALSE;
  bQueryInputValue7 : BOOL := FALSE;
  bQueryInputValue8 : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

**Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  bReadingInputValue3 : BOOL;
  bReadingInputValue4 : BOOL;
  bReadingInputValue5 : BOOL;
  bReadingInputValue6 : BOOL;
  bReadingInputValue7 : BOOL;
  bReadingInputValue8 : BOOL;
  nPushButton1Event : WORD;
```

```

nPushButton2Event      : WORD;
nPushButton3Event      : WORD;
nPushButton4Event      : WORD;
nPushButton5Event      : WORD;
nPushButton6Event      : WORD;
nPushButton7Event      : WORD;
nPushButton8Event      : WORD;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState](#) [▶ 840] can be used to convert the state of a push button into a BOOL variable.

### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	A TRUE on this property releases the instance. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.4.5 Helvar**

**4.1.4.5.1 FB\_DALI\_Helvar\_2ButtonModule**

FB_DALI_Helvar_2ButtonModule	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryInputValue1 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue1
bQueryInputValue2 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue2
	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event

The function block represents the 2 Button DALI push button interface module from Helvar.

Up to eight push buttons (instances 0 to 1) can be connected directly via the DALI device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - Introduction [► 511]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors ( $bError = FALSE$ ). If the output  $bError = TRUE$ , the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable  $bQueryInputValueN$ , even if the corresponding instance is not enabled.

The properties  $bEnablePushButtonN$  make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress       : BYTE  := 0;
  nOptions            : DWORD := 0;
  bQueryInputValue1  : BOOL := FALSE;
  bQueryInputValue2  : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

### Outputs

```
VAR_OUTPUT
  bError              : BOOL;
  ipResultMessage    : I_TcMessage;
  bInitializing      : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  nPushButton1Event  : WORD;
  nPushButton2Event  : WORD;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	BOOL	The outputs output the events for the corresponding push button via one bit for one cycle.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.32	Tc3_DALI from v3.12.0.0

**4.1.4.5.2 FB\_DALI\_Helvar\_4ButtonModule**

<b>FB_DALI_Helvar_4ButtonModule</b>	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryInputValue1 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue1
bQueryInputValue2 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue2
bQueryInputValue3 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue3
bQueryInputValue4 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue4
	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event
	<i>WORD</i> nPushButton3Event
	<i>WORD</i> nPushButton4Event

The function block represents the 4 Button DALI push button interface module from Helvar.

Up to eight push buttons (instances 0 to 3) can be connected directly via the DALI device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction](#) [[▶ 511](#)]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize      : BOOL := FALSE;
  nShortAddress    : BYTE  := 0;
  nOptions         : DWORD := 0;
  bQueryInputValue1 : BOOL := FALSE;
  bQueryInputValue2 : BOOL := FALSE;
  bQueryInputValue3 : BOOL := FALSE;
  bQueryInputValue4 : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bInitializing    : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  bReadingInputValue3 : BOOL;
  bReadingInputValue4 : BOOL;
  nPushButton1Event : WORD;
  nPushButton2Event : WORD;
  nPushButton3Event : WORD;
  nPushButton4Event : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	BOOL	The outputs output the events for the corresponding push button via one bit for one cycle.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState](#) [▶ 840] can be used to convert the state of a push button into a BOOL variable.



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.32	Tc3_DALI from v3.12.0.0

**4.1.4.5.3 FB\_DALI\_Helvar\_5ButtonModule**

FB_DALI_Helvar_5ButtonModule	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryInputValue1 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue1
bQueryInputValue2 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue2
bQueryInputValue3 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue3
bQueryInputValue4 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue4
bQueryInputValue5 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue5
	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event
	<i>WORD</i> nPushButton3Event
	<i>WORD</i> nPushButton4Event
	<i>WORD</i> nPushButton5Event

The function block represents the 5 Button DALI push button interface module from Helvar.

Up to eight push buttons (instances 0 to 4) can be connected directly via the DALI device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[▶ 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryInputValue1    : BOOL := FALSE;
  bQueryInputValue2    : BOOL := FALSE;
  bQueryInputValue3    : BOOL := FALSE;
  bQueryInputValue4    : BOOL := FALSE;
  bQueryInputValue5    : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingInputValue1   : BOOL;
  bReadingInputValue2   : BOOL;
  bReadingInputValue3   : BOOL;
  bReadingInputValue4   : BOOL;
  bReadingInputValue5   : BOOL;
  nPushButton1Event     : WORD;
  nPushButton2Event     : WORD;
  nPushButton3Event     : WORD;
  nPushButton4Event     : WORD;
  nPushButton5Event     : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	BOOL	The outputs output the events for the corresponding push button via one bit for one cycle.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.32	Tc3_DALI from v3.12.0.0

**4.1.4.5.4 FB\_DALI\_Helvar\_6ButtonModule**

<b>FB_DALI_Helvar_6ButtonModule</b>	
— bInitialize <i>BOOL</i>	<i>BOOL</i> bError
— nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
— nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
— bQueryInputValue1 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue1
— bQueryInputValue2 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue2
— bQueryInputValue3 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue3
— bQueryInputValue4 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue4
— bQueryInputValue5 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue5
— bQueryInputValue6 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue6
	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event
	<i>WORD</i> nPushButton3Event
	<i>WORD</i> nPushButton4Event
	<i>WORD</i> nPushButton5Event
	<i>WORD</i> nPushButton6Event

The function block represents the 6 Button DALI push button interface module from Helvar.

Up to eight push buttons (instances 0 to 5) can be connected directly via the DALI device.



For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[▶ 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize      : BOOL := FALSE;
  nShortAddress    : BYTE  := 0;
  nOptions         : DWORD := 0;
  bQueryInputValue1 : BOOL := FALSE;
  bQueryInputValue2 : BOOL := FALSE;
  bQueryInputValue3 : BOOL := FALSE;
  bQueryInputValue4 : BOOL := FALSE;
  bQueryInputValue5 : BOOL := FALSE;
  bQueryInputValue6 : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  bReadingInputValue3 : BOOL;
  bReadingInputValue4 : BOOL;
  bReadingInputValue5 : BOOL;
  bReadingInputValue6 : BOOL;
  nPushButton1Event : WORD;
  nPushButton2Event : WORD;
  nPushButton3Event : WORD;
  nPushButton4Event : WORD;
```

```
nPushButton5Event      : WORD;
nPushButton6Event      : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	BOOL	The outputs output the events for the corresponding push button via one bit for one cycle.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.32	Tc3_DALI from v3.12.0.0

**4.1.4.5.5 FB\_DALI\_Helvar\_7ButtonModule**

FB_DALI_Helvar_7ButtonModule	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryInputValue1 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue1
bQueryInputValue2 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue2
bQueryInputValue3 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue3
bQueryInputValue4 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue4
bQueryInputValue5 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue5
bQueryInputValue6 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue6
bQueryInputValue7 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue7
	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event
	<i>WORD</i> nPushButton3Event
	<i>WORD</i> nPushButton4Event
	<i>WORD</i> nPushButton5Event
	<i>WORD</i> nPushButton6Event
	<i>WORD</i> nPushButton7Event

The function block represents the 7 Button DALI push button interface module from Helvar.

Up to eight push buttons (instances 0 to 6) can be connected directly via the DALI device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[▶ 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryInputValue1    : BOOL := FALSE;
  bQueryInputValue2    : BOOL := FALSE;
  bQueryInputValue3    : BOOL := FALSE;
  bQueryInputValue4    : BOOL := FALSE;
  bQueryInputValue5    : BOOL := FALSE;
  bQueryInputValue6    : BOOL := FALSE;
  bQueryInputValue7    : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingInputValue1   : BOOL;
  bReadingInputValue2   : BOOL;
  bReadingInputValue3   : BOOL;
  bReadingInputValue4   : BOOL;
  bReadingInputValue5   : BOOL;
  bReadingInputValue6   : BOOL;
  bReadingInputValue7   : BOOL;
  nPushButton1Event     : WORD;
  nPushButton2Event     : WORD;
  nPushButton3Event     : WORD;
```

```
nPushButton4Event      : WORD;
nPushButton5Event      : WORD;
nPushButton6Event      : WORD;
nPushButton7Event      : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	BOOL	The outputs output the events for the corresponding push button via one bit for one cycle.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.32	Tc3_DALI from v3.12.0.0

**4.1.4.5.6 FB\_DALI\_Helvar\_8ButtonModule**

<b>FB_DALI_Helvar_8ButtonModule</b>	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryInputValue1 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue1
bQueryInputValue2 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue2
bQueryInputValue3 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue3
bQueryInputValue4 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue4
bQueryInputValue5 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue5
bQueryInputValue6 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue6
bQueryInputValue7 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue7
bQueryInputValue8 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue8
	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event
	<i>WORD</i> nPushButton3Event
	<i>WORD</i> nPushButton4Event
	<i>WORD</i> nPushButton5Event
	<i>WORD</i> nPushButton6Event
	<i>WORD</i> nPushButton7Event
	<i>WORD</i> nPushButton8Event

The function block represents the 8 Button DALI push button interface module from Helvar.

Up to eight push buttons (instances 0 to 7) can be connected directly via the DALI device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[▶ 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress       : BYTE  := 0;
  nOptions            : DWORD := 0;
  bQueryInputValue1  : BOOL := FALSE;
  bQueryInputValue2  : BOOL := FALSE;
  bQueryInputValue3  : BOOL := FALSE;
  bQueryInputValue4  : BOOL := FALSE;
  bQueryInputValue5  : BOOL := FALSE;
  bQueryInputValue6  : BOOL := FALSE;
  bQueryInputValue7  : BOOL := FALSE;
  bQueryInputValue8  : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

**Outputs**

```
VAR_OUTPUT
  bError              : BOOL;
  ipResultMessage    : I_TcMessage;
  bInitializing       : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  bReadingInputValue3 : BOOL;
  bReadingInputValue4 : BOOL;
  bReadingInputValue5 : BOOL;
  bReadingInputValue6 : BOOL;
  bReadingInputValue7 : BOOL;
```

```

bReadingInputValue8      : BOOL;
nPushButton1Event       : WORD;
nPushButton2Event       : WORD;
nPushButton3Event       : WORD;
nPushButton4Event       : WORD;
nPushButton5Event       : WORD;
nPushButton6Event       : WORD;
nPushButton7Event       : WORD;
nPushButton8Event       : WORD;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValueN	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	BOOL	The outputs output the events for the corresponding push button via one bit for one cycle.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.



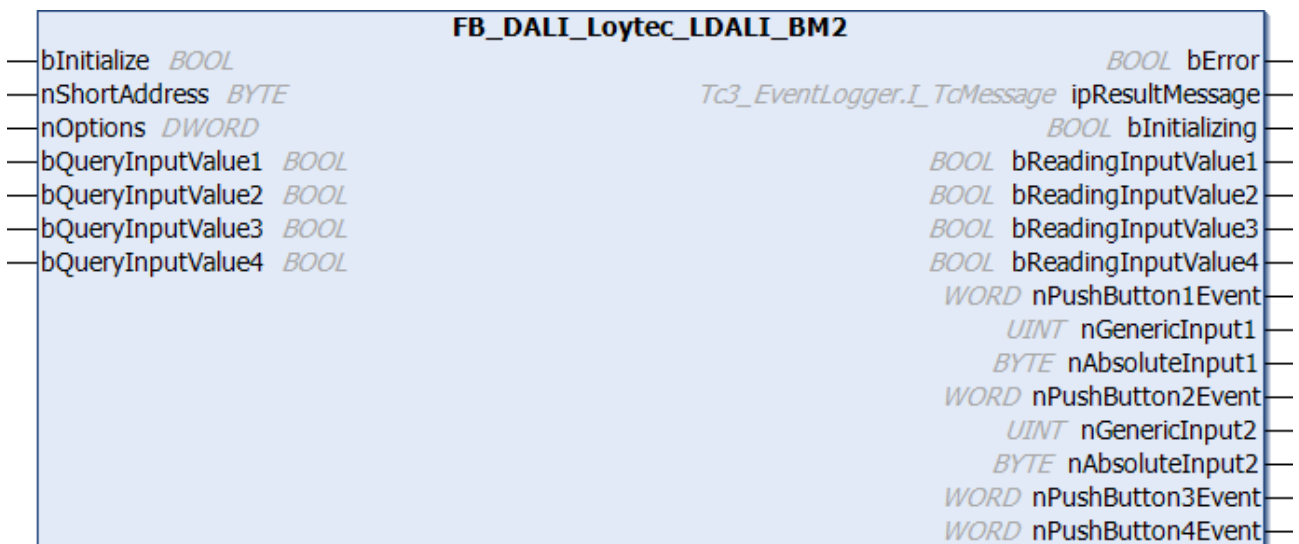
Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.32	Tc3_DALI from v3.12.0.0

**4.1.4.6 Loytec**

**4.1.4.6.1 FB\_DALI\_Loytec\_LDALI\_BM2**



The function block represents the LDALI-BM2 DALI push button interface from Loytec.

Up to four push buttons (instances 0, 3, 6 and 7) can be connected directly via the device.

The first two inputs are configurable and it is therefore also possible to use NTC/PTC temperature sensors (instances 1 and 4) or potentiometers/slide resistors (instances 2 and 5) instead of push buttons.

These functions can be configured via the properties *eInput1Configuration* and *eInput2Configuration* (see [E\\_DALILoytecInputConfiguration](#) [▶ 863]).

The resistance value is read directly on setting one of the two inputs as Generic Input (resistance thermometer). This reading takes place when a positive edge is detected at the respective input *bQueryInputValue1* or *bQueryInputValue2*.

The resistance value is read in % when setting one of the two inputs as Absolute Input (potentiometer). The corresponding channel is configured via the property *eResistanceValueAbsoluteInput1* or *eResistanceValueAbsoluteInput2* on the connected resistor (1 kΩ or 10 kΩ) (see [Use of properties](#) [▶ 890]). The output value lies between 0 (0 %) and 31 (100 %).

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

In addition, the time-dependent querying of these values is possible. The properties *nReportTimerAbsoluteInput1* and *nReportTimerAbsoluteInput2* are used for this. The value is read again on expiry of the set time [s]. To prevent events following one another too quickly, a dead time [ms] can be set via the properties *nDeadtimeTimerAbsoluteInput1* and *nDeadtimeTimerAbsoluteInput2*.

The push button inputs 3 and 4 can be disabled via the properties *bEnableInput3* and *bEnableInput4*.

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



Set the event priorities on the properties *eEventPriorityInput1* and *eEventPriorityInput2* to the value *MiddleLow* if the inputs are not used as push buttons.



The function block cannot be used when using the KL6811.

## Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress       : BYTE  := 0;
  nOptions            : DWORD := 0;
  bQueryInputValue1  : BOOL  := FALSE;
  bQueryInputValue2  : BOOL  := FALSE;
  bQueryInputValue3  : BOOL  := FALSE;
  bQueryInputValue4  : BOOL  := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input. Values of resistance sensors (setting of the input as Generic Input) can only be queried manually and not event-driven.

 **Outputs**

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  bReadingInputValue3 : BOOL;
  bReadingInputValue4 : BOOL;
  nPushButton1Event : WORD;
  nGenericInput1    : UINT;
  nAbsoluteInput1   : BYTE;
  nPushButton2Event : WORD;
  nGenericInput2    : UINT;
  nAbsoluteInput2   : BYTE;
  nPushButton3Event : WORD;
  nPushButton4Event : WORD;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[► 840\]](#) can be used to convert the state of a push button into a BOOL variable.

Name	Type	Description
nGenericInputN	UINT	Outputs containing the measured values of the connected resistance sensors (NTC/PTC).
nAbsoluteInputN	BYTE	Outputs containing the measured values of the connected potentiometers or slide resistors. The individual ranges are scaled from 0 to 100%, with the following output values: 0 = 0% and 31 = 100%.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [[▶ 890](#)].

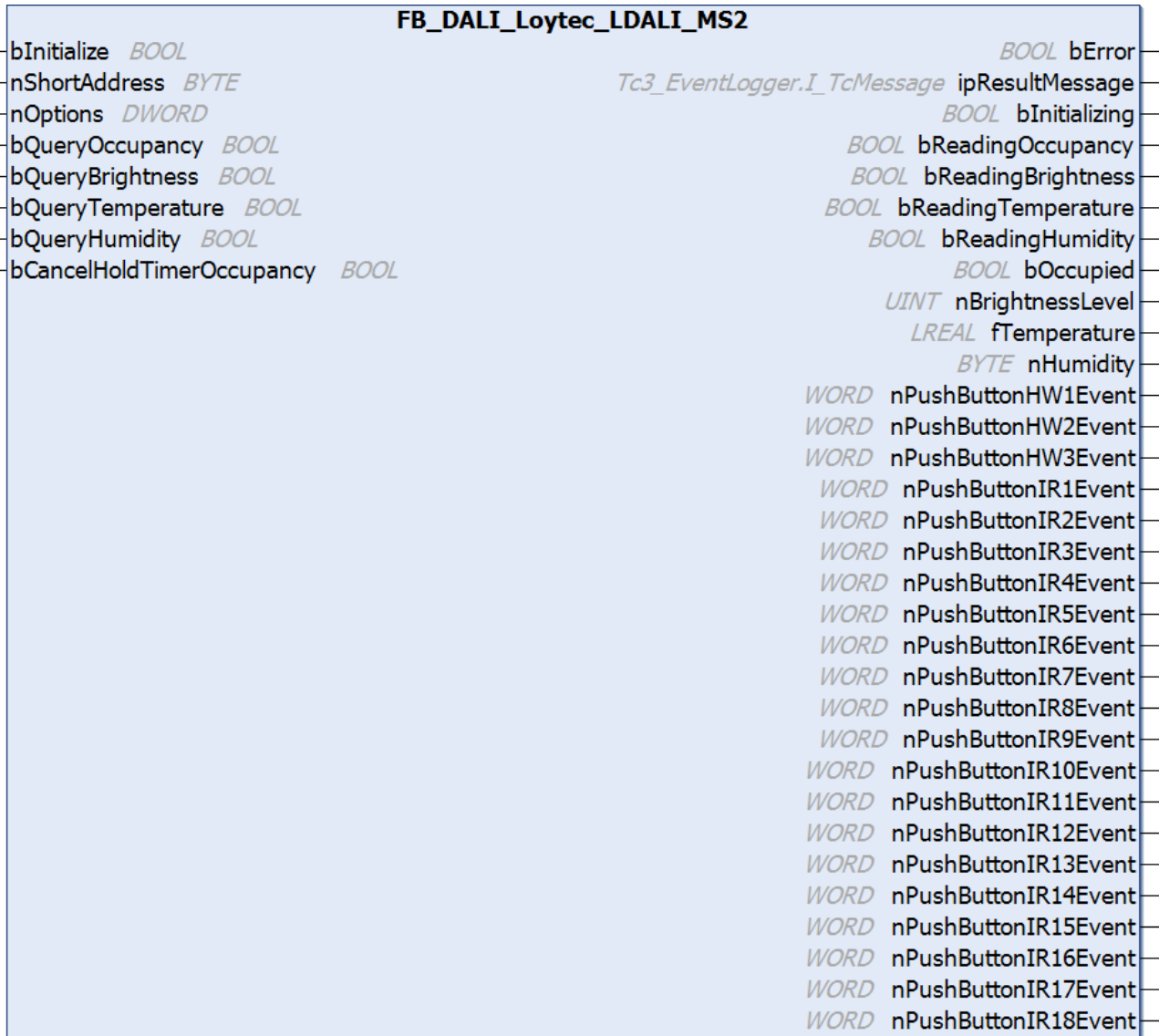
The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
eInputNConfiguration	E_DALILo ytecInput Configura tion [ <a href="#">▶ 863</a> ]	Get, Set	PushButt on	Configuration of input 1 or 2 as a push button or analog input for the connection of potentiometers or resistance sensors, e.g. NTC/PTC.
eResistanceValueAbsoluteInputN	E_DALILo ytecResist anceConf iguration [ <a href="#">▶ 863</a> ]	Get, Set	Resistanc e10kOhm	Configuration of the resistance value of the potentiometer or slide resistor at input 1 or 2.
bEnablePushButtonN	BOOL	Get, Set	TRUE	Push button input 3 or 4, which can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_ 0011	Setting of the <a href="#">event filter</a> [ <a href="#">▶ 512</a> ].
eEventPriorityInputN	E_DALIE ventPrior ity	Get, Set	Middle	Setting of the <a href="#">event priority</a> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerAbsoluteInputN	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nReportTimerAbsoluteInputN	BYTE	Get, Set	0 s	Time [0...255 s] after which the pending event of the absolute encoder is repeated if no other event has occurred in the meantime.
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICo mmunicat ion	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.4.6.2 FB\_DALI\_Loytec\_LDALI\_MS2



The function block FB\_DALI\_Loytec\_LDALI\_MS2 represents the LDALI-MS2-BT or LDALI-MS2 DALI sensor from Loytec.

This outputs a brightness value (instance 0), an occupancy signal (instance 1), a temperature value (instance 2), a relative humidity value (instance 3), three hardware push button signals (instances 4 to 6) and 18 infrared remote control push button signals (instances 7 to 24).

The infrared push button instances can be used via remote control from the vendor.

For information on the exact functions of the individual instances, please refer to the vendor's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) – [Introduction \[▶ 511\]](#)

Part 303 (occupancy sensors) - [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The temperature measurement and humidity measurement are executed as Generic Inputs. These values can only be queried via a positive edge at the inputs *bQueryTemperature* and *bQueryHumidity*.

For the other instances, the reading can also be triggered directly via the inputs *bQueryBrightness* and *bQueryOccupancy*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bQueryTemperature    : BOOL := FALSE;
  bQueryHumidity       : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bQueryTemperature	BOOL	The temperature is queried immediately by a positive edge at this input. The DALI device does not support transmission of the temperature per event.
bQueryHumidity	BOOL	The relative humidity is queried immediately by a positive edge at this input. The DALI device does not support transmission of the relative humidity per event.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

### Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingOccupancy     : BOOL;
  bReadingBrightness    : BOOL;
  bReadingTemperature   : BOOL;
  bReadingHumidity     : BOOL;
  bOccupied             : BOOL;
  nBrightnessLevel     : UINT;
  fTemperature          : LREAL;
  nHumidity             : BYTE;
```

```
nPushButtonHWNEvent : WORD;
nPushButtonIRNEvent : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
fTemperature	LREAL	Measured temperature value (-5.0...60.0°C) of the temperature sensor.
nHumidity	BYTE	Measured value of the relative humidity (0...100%).
nPushButtonHWNEvent	WORD	The outputs (1...3) output the events for the corresponding push button via one bit for one cycle.
nPushButtonIRNEvent	WORD	The outputs (1...18) output the events for the corresponding infrared button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[► 840\]](#) can be used to convert the state of a push button into a BOOL variable.



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.



Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnableTemperature	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for temperature measurement. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryTemperature</i> .
bEnableHumidity	BOOL	Get, Set	FALSE	A TRUE on this property enables the instance to measure relative humidity. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryHumidity</i> .
bEnablePushButtonHWN	BOOL	Get, Set	FALSE	Push button input 1 to 3, which can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
bEnablePushButtonIRN	BOOL	Get, Set	FALSE	Infrared buttons input 1 to 18, which can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	255	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.

Name	Type	Access	Initial value	Description
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
eEventPriorityPushButtonHWN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nEventFilterPushButtonHWN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
nDoubleTimerPushButtonHWN	UINT	Get, Set	0 s	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonHWN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonHWN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonHWN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
eEventPriorityPushButtonIRN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nEventFilterPushButtonIRN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
nDoubleTimerPushButtonIRN	UINT	Get, Set	0 s	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonIRN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonIRN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonIRN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

4.1.4.6.3 FB\_DALI\_Loytec\_LDALI\_MS4\_BT

FB_DALI_Loytec_LDALI_MS4_BT	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryOccupancy <i>BOOL</i>	<i>BOOL</i> bReadingOccupancy
bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
bQueryTemperature <i>BOOL</i>	<i>BOOL</i> bReadingTemperature
bQueryHumidity <i>BOOL</i>	<i>BOOL</i> bReadingHumidity
bCancelHoldTimerOccupancy <i>BOOL</i>	<i>BOOL</i> bOccupied
	<i>UINT</i> nBrightnessLevel
	<i>LREAL</i> fTemperature
	<i>BYTE</i> nHumidity
	<i>WORD</i> nPushButtonHW1Event
	<i>WORD</i> nPushButtonHW2Event
	<i>WORD</i> nPushButtonHW3Event
	<i>WORD</i> nPushButtonIR1Event
	<i>WORD</i> nPushButtonIR2Event
	<i>WORD</i> nPushButtonIR3Event
	<i>WORD</i> nPushButtonIR4Event
	<i>WORD</i> nPushButtonIR5Event
	<i>WORD</i> nPushButtonIR6Event
	<i>WORD</i> nPushButtonIR7Event
	<i>WORD</i> nPushButtonIR8Event
	<i>WORD</i> nPushButtonIR9Event
	<i>WORD</i> nPushButtonIR10Event
	<i>WORD</i> nPushButtonIR11Event
	<i>WORD</i> nPushButtonIR12Event
	<i>WORD</i> nPushButtonIR13Event
	<i>WORD</i> nPushButtonIR14Event
	<i>WORD</i> nPushButtonIR15Event
	<i>WORD</i> nPushButtonIR16Event
	<i>WORD</i> nPushButtonIR17Event
	<i>WORD</i> nPushButtonIR18Event

The function block FB\_DALI\_Loytec\_LDALI\_MS4 represents the LDALI-MS4-BT DALI sensor from Loytec.

This outputs a brightness value (instance 0), an occupancy signal (instance 1), a temperature value (instance 2), a relative humidity value (instance 3), three hardware push button signals (instances 4 to 6) and 18 infrared remote control push button signals (instances 7 to 24).

The infrared push button instances can be used via remote control from the vendor.

For information on the exact functions of the individual instances, please refer to the vendor's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) – [Introduction \[▶ 511\]](#)

Part 303 (occupancy sensors) - [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The temperature measurement and humidity measurement are executed as Generic Inputs. These values can only be queried via a positive edge at the inputs *bQueryTemperature* and *bQueryHumidity*.

For the other instances, the reading can also be triggered directly via the inputs *bQueryBrightness* and *bQueryOccupancy*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bQueryTemperature    : BOOL  := FALSE;
  bQueryHumidity       : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bQueryTemperature	BOOL	The temperature is queried immediately by a positive edge at this input. The DALI device does not support transmission of the temperature per event.
bQueryHumidity	BOOL	The relative humidity is queried immediately by a positive edge at this input. The DALI device does not support transmission of the relative humidity per event.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingOccupancy     : BOOL;
  bReadingBrightness    : BOOL;
  bReadingTemperature   : BOOL;
  bReadingHumidity      : BOOL;
  bOccupied             : BOOL;
  nBrightnessLevel      : UINT;
  fTemperature          : LREAL;
  nHumidity              : BYTE;
  nPushButtonHWNEvent   : WORD;
  nPushButtonIRNEvent   : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
fTemperature	LREAL	Measured temperature value (-5.0...60.0°C) of the temperature sensor.
nHumidity	BYTE	Measured value of the relative humidity (0...100%).
nPushButtonHWNEvent	WORD	The outputs (1...3) output the events for the corresponding push button via one bit for one cycle.
nPushButtonIRNEvent	WORD	The outputs (1...18) output the events for the corresponding infrared button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[► 840\]](#) can be used to convert the state of a push button into a BOOL variable.



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnableTemperature	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for temperature measurement. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryTemperature</i> .
bEnableHumidity	BOOL	Get, Set	FALSE	A TRUE on this property enables the instance to measure relative humidity. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryHumidity</i> .
bEnablePushButtonHWN	BOOL	Get, Set	FALSE	Push button input 1 to 3, which can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
bEnablePushButtonIRN	BOOL	Get, Set	FALSE	Infrared buttons input 1 to 18, which can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	255	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.



Name	Type	Access	Initial value	Description
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
eEventPriorityPushButtonHWN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nEventFilterPushButtonHWN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
nDoubleTimerPushButtonHWN	UINT	Get, Set	0 s	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonHWN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonHWN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonHWN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
eEventPriorityPushButtonIRN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nEventFilterPushButtonIRN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
nDoubleTimerPushButtonIRN	UINT	Get, Set	0 s	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonIRN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonIRN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonIRN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.15	Tc3_DALI from v3.6.2.0

**4.1.4.7 Lunatone**

**4.1.4.7.1 FB\_DALI\_Lunatone\_CS**

FB_DALI_Lunatone_CS	
— bInitialize <i>BOOL</i>	<i>BOOL</i> bError
— nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
— nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
— bQueryOccupancy <i>BOOL</i>	<i>BOOL</i> bReadingOccupancy
— bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
— bCancelHoldTimerOccupancy <i>BOOL</i>	<i>BOOL</i> bOccupied
	<i>UINT</i> nBrightnessLevel

The function block FB\_DALI\_Lunatone\_CS represents the DALI-2 sensor CS Integration from Lunatone.

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

The order number is stored in `memory bank 0` [▶ 16] from offset 143 (MSB) to offset 146 (LSB). The addition of the order number (e.g. "-HS") is in the ASCII code from offset 147 to 162.

For information on the exact functions of the individual instances, please refer to the vendor's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction](#) [▶ 542]

Part 304 (light sensors) – [Introduction](#) [▶ 561]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at `bInitialize` and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (`bError = FALSE`). If the output `bError = TRUE`, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables `bQueryOccupancy` and `bQueryBrightness`, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <code>bOccupied</code> .

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
```



```

bReadingOccupancy      : BOOL;
bReadingBrightness     : BOOL;
bOccupied              : BOOL;
nBrightnessLevel       : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the motion sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value (0...2046 lux) of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	20	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.22	Tc3_DALI from v3.9.0.0

4.1.4.7.2 FB\_DALI\_Lunatone\_CS\_THP

FB_DALI_Lunatone_CS_THP	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryOccupancy <i>BOOL</i>	<i>BOOL</i> bReadingOccupancy
bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
bQueryTemperature <i>BOOL</i>	<i>BOOL</i> bReadingTemperature
bQueryHumidity <i>BOOL</i>	<i>BOOL</i> bReadingHumidity
bQueryAirPreasure <i>BOOL</i>	<i>BOOL</i> bReadingAirPreasure
bCancelHoldTimerOccupancy <i>BOOL</i>	<i>BOOL</i> bOccupied
	<i>UINT</i> nBrightnessLevel
	<i>LREAL</i> fTemperature
	<i>LREAL</i> fHumidity
	<i>UINT</i> nAirPreasure

The function block FB\_DALI\_Lunatone\_CS\_THP represents the DALI-2 sensor CS Integration THP from Lunatone.

This outputs an occupancy signal (instance 0), a brightness value (instance 1), a temperature value (instance 2), a relative humidity value (instance 3) and an air pressure value (instance 4).

The order number is stored in memory bank 0 [▶ 16] from offset 143 (MSB) to offset 146 (LSB). The addition of the order number (e.g. "-HS") is in the ASCII code from offset 147 to 162.

For information on the exact functions of the individual instances, please refer to the vendor's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction](#) [▶ 542]

Part 304 (light sensors) – [Introduction](#) [▶ 561]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bQueryTemperature    : BOOL  := FALSE;
  bQueryHumidity       : BOOL  := FALSE;
```

```

bQueryAirPreasure      : BOOL := FALSE;
bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR

```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bQueryTemperature	BOOL	The temperature is queried immediately by a positive edge at this input. The DALI device does not support transmission of the temperature per event.
bQueryHumidity	BOOL	The relative humidity is queried immediately by a positive edge at this input. The DALI device does not support transmission of the relative humidity per event.
bQueryAirPreasure	BOOL	The air pressure is queried immediately by a positive edge at this input. The DALI device does not support transmission of the air pressure per event.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bReadingTemperature : BOOL;
  bReadingHumidity : BOOL;
  bReadingAirPreasure : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
  fTemperature     : LREAL;
  fHumidity        : LREAL;
  nAirPreasure     : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the motion sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bReadingTemperature	BOOL	The output is TRUE as long as values of the instance of the temperature sensor are being read by the DALI control device.
bReadingHumidity	BOOL	The output is TRUE as long as values of the instance of the humidity sensor are being read by the DALI control device.
bReadingAirPressure	BOOL	The output is TRUE as long as values of the instance of the air pressure sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value (0...2046 lux) of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
fTemperature	LREAL	Measured temperature value (-20.0...80.0 °C) of the temperature sensor.
fHumidity	LREAL	Measured value of the relative humidity (0.0...100.0 %).
nAirPressure	UINT	Measured value of air pressure (300...1100 hPa).



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnableTemperature	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for temperature measurement. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryTemperature</i> .
bEnableHumidity	BOOL	Get, Set	FALSE	A TRUE on this property enables the instance to measure relative humidity. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryHumidity</i> .
bEnableAirPressure	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for air pressure measurement. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryAirPressure</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	20	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.47	Tc3_DALI from v3.16.1.0

**4.1.4.7.3 FB\_DALI\_Lunatone\_MC**



The function block FB\_DALI\_Lunatone\_MC represents the MC DALI push button interface from Lunatone.

Up to four push buttons (instances 0 to 3) can be connected directly via the device.

For information on the exact functions of the individual instances, please refer to the vendor's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction](#) [[▶ 511](#)]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize      : BOOL := FALSE;
  nShortAddress    : BYTE := 0;
```

```
nOptions          : DWORD := 0;
bQueryInputValue1 : BOOL  := FALSE;
bQueryInputValue2 : BOOL  := FALSE;
bQueryInputValue3 : BOOL  := FALSE;
bQueryInputValue4 : BOOL  := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

 **Outputs**

```
VAR_OUTPUT
bError          : BOOL;
ipResultMessage : I_TcMessage;
bInitializing   : BOOL;
bReadingInputValue1 : BOOL;
bReadingInputValue2 : BOOL;
bReadingInputValue3 : BOOL;
bReadingInputValue4 : BOOL;
nPushButton1Event : WORD;
nPushButton2Event : WORD;
nPushButton3Event : WORD;
nPushButton4Event : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValueN	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.



The function block `FB_DALI_ToPushButtonState` [► 840] can be used to convert the state of a push button into a BOOL variable.

 **Properties**

All parameters that are written to the DALI control device via `bInitialize` are available as properties [► 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <code>bQueryPushButtonN</code> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <code>tDoubleMin</code> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <code>tShortMin</code> ...5100 ms) to distinguish between a long and a short button press. <code>tShortMin</code> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.7.4 FB\_DALI\_Lunatone\_MC4L**



The function block represents the MC4L DALI push button interface from Lunatone.

Up to four push buttons (instances 0 to 3) can be connected directly via the device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryInputValue1    : BOOL  := FALSE;
  bQueryInputValue2    : BOOL  := FALSE;
  bQueryInputValue3    : BOOL  := FALSE;
  bQueryInputValue4    : BOOL  := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingInputValue1   : BOOL;
  bReadingInputValue2   : BOOL;
  bReadingInputValue3   : BOOL;
  bReadingInputValue4   : BOOL;
  nPushButton1Event     : WORD;
  nPushButton2Event     : WORD;
  nPushButton3Event     : WORD;
  nPushButton4Event     : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

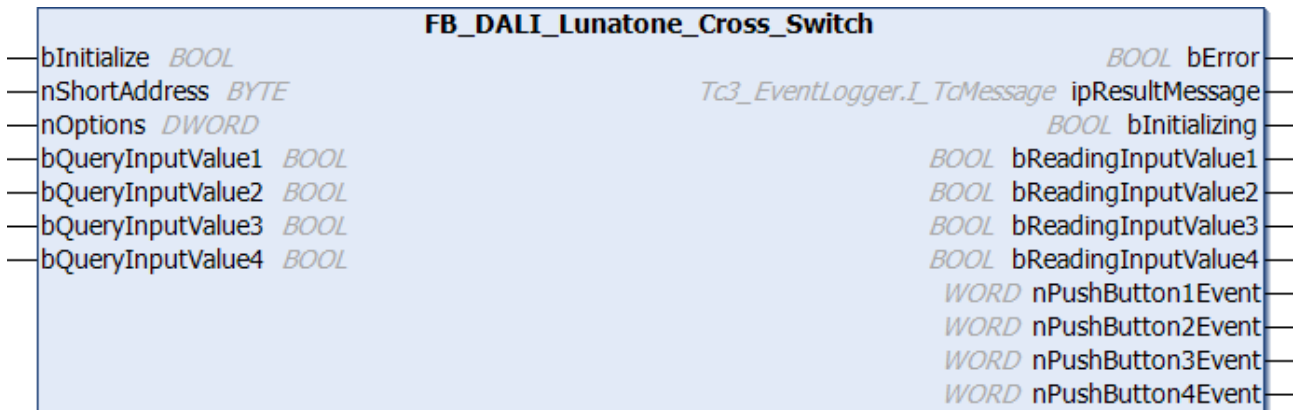
The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.22	Tc3_DALI from v3.9.0.0

**4.1.4.7.5 FB\_DALI\_Lunatone\_Cross\_Switch**



The function block represents the Cross Switch DALI push button interface from Lunatone.

Up to four push buttons (instances 0 to 3) can be connected directly via the device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction](#) [[▶ 511](#)]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryInputValue1    : BOOL  := FALSE;
  bQueryInputValue2    : BOOL  := FALSE;
  bQueryInputValue3    : BOOL  := FALSE;
  bQueryInputValue4    : BOOL  := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingInputValue1   : BOOL;
  bReadingInputValue2   : BOOL;
  bReadingInputValue3   : BOOL;
  bReadingInputValue4   : BOOL;
  nPushButton1Event     : WORD;
  nPushButton2Event     : WORD;
  nPushButton3Event     : WORD;
  nPushButton4Event     : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue N	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState](#) [► 840] can be used to convert the state of a push button into a BOOL variable.

### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.22	Tc3_DALI from v3.9.0.0

**4.1.4.8 Niko**

**4.1.4.8.1 FB\_DALI\_Niko\_P46LR**



The function block represents the DALI sensors P46LR for surface mounting, P46LR for SnapFit flush mounting and P46LR for flush-mounting box from Niko.

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction](#) [[▶ 542](#)]

Part 304 (light sensors) – [Introduction](#) [[▶ 561](#)]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage      : I_TcMessage;
  bInitializing         : BOOL;
  bReadingOccupancy    : BOOL;
  bReadingBrightness   : BOOL;
  bOccupied             : BOOL;
  nBrightnessLevel     : UINT;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.



### Properties

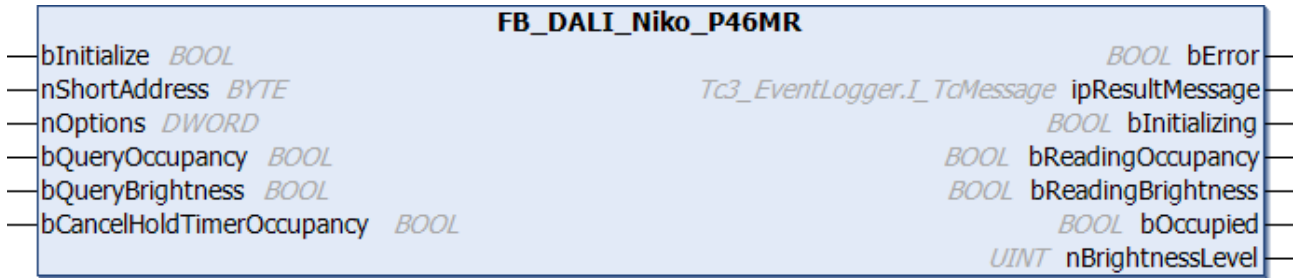
All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	1 (min)	Setting of the sensitivity of the occupancy sensor instance from 0 (off), 1 (min), 2 (low), 3 (high) to 4 (max).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

4.1.4.8.2 FB\_DALI\_Niko\_P46MR



The function block represents the DALI sensors P46MR for surface mounting, P46MR for SnapFit flush mounting and P46MR for flush-mounting box from Niko.

This outputs an occupancy signal (instance 0) and a brightness value (instance 1).

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightness    : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

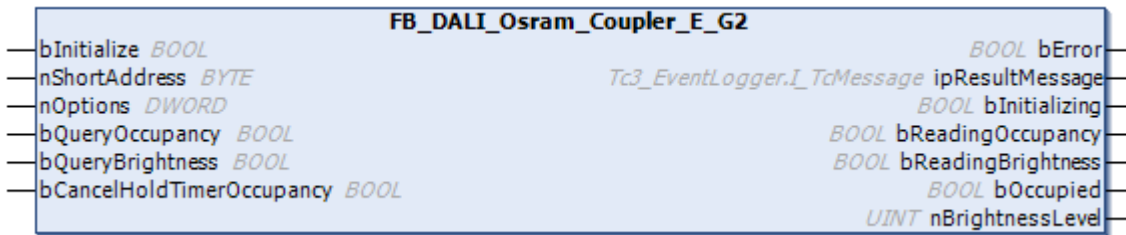
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	1 (min)	Setting of the sensitivity of the occupancy sensor instance from 0 (off), 1 (min), 2 (low), 3 (high) to 4 (max).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_DALI from v3.10.5.0

### 4.1.4.9 Osram

#### 4.1.4.9.1 FB\_DALI\_Osram\_Coupler\_E\_G2



The function block represents the DALI COUPLER E G2 DALI sensor from Osram.

This outputs the measured brightness value (instance 1) and an occupancy signal (instance 0).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

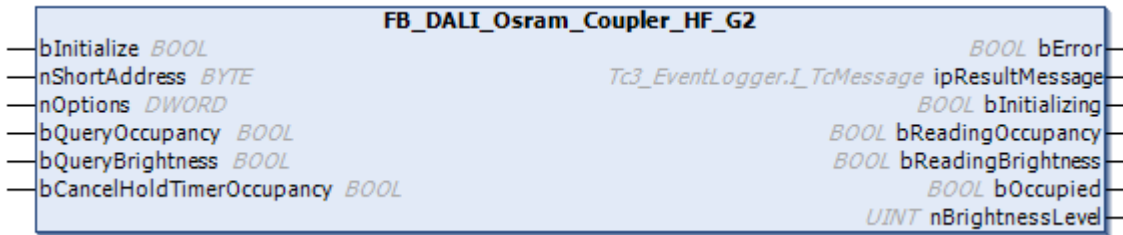
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEvent Priority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEvent Priority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0



4.1.4.9.2 FB\_DALI\_Osram\_Coupler\_HF\_G2



The function block represents the DALI COUPLER HF G2 DALI sensor from Osram.

This outputs the measured brightness value (instance 1) and an occupancy signal (instance 0).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy       : BOOL  := FALSE;
  bQueryBrightness      : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

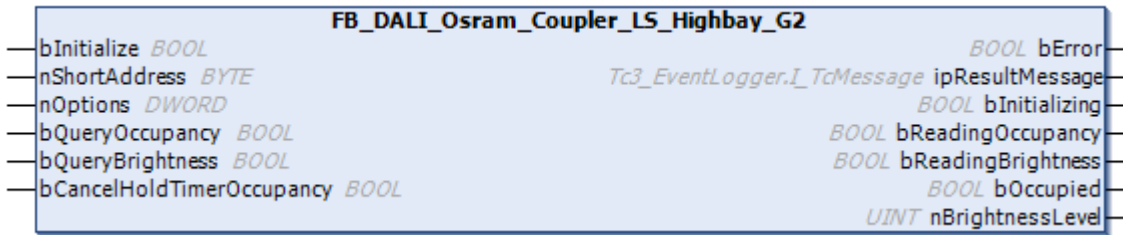
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEvent Priority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEvent Priority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

### 4.1.4.9.3 FB\_DALI\_Osram\_Coupler\_LS\_Highbay\_G2



The function block represents the DALI COUPLER LS HIGHBAY G2 DALI sensor from Osram.

This outputs the measured brightness value (instance 1) and an occupancy signal (instance 0).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightness    : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

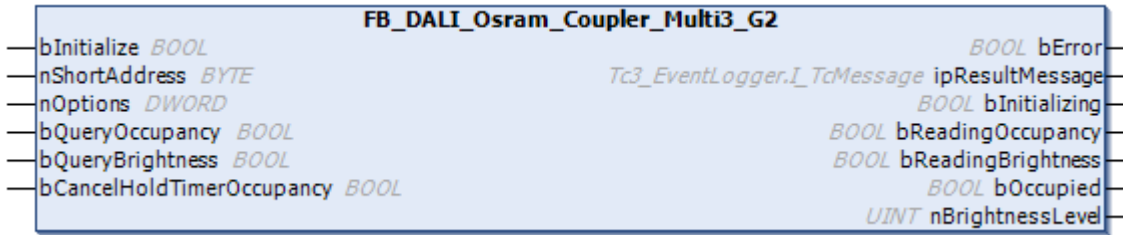
All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DAL IEvent Priority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DAL IEvent Priority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALI Communication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

4.1.4.9.4 FB\_DALI\_Osram\_Coupler\_Multi3\_G2



The function block represents the DALI COUPLER MULTI3 G2 DALI sensor from Osram.

This outputs the measured brightness value (instance 1) and an occupancy signal (instance 0).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

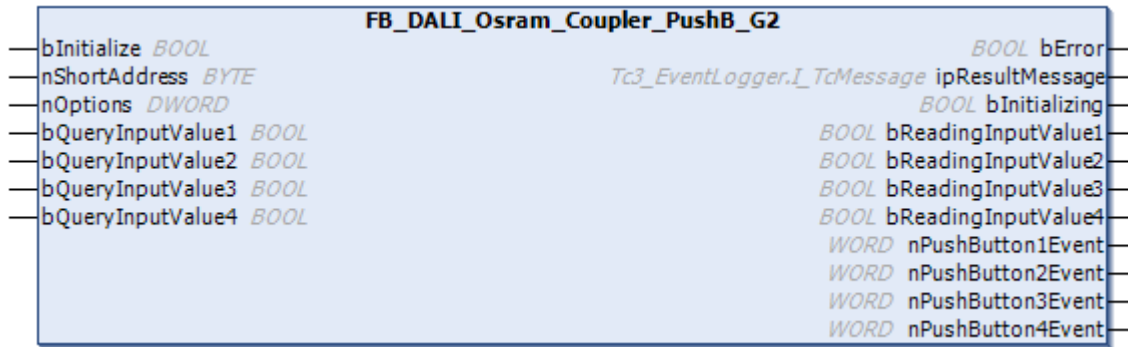


Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEvent Priority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEvent Priority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

#### 4.1.4.9.5 FB\_DALI\_Osram\_Coupler\_PushB\_G2



The function block represents the DALI COUPLER PushB G2 DALI push button interface from Osram.

Up to four push buttons (instances 0 to 3) can be connected directly via the device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryInputValue1   : BOOL := FALSE;
  bQueryInputValue2   : BOOL := FALSE;
  bQueryInputValue3   : BOOL := FALSE;
  bQueryInputValue4   : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  bReadingInputValue3 : BOOL;
  bReadingInputValue4 : BOOL;
  nPushButton1Event : WORD;
  nPushButton2Event : WORD;
  nPushButton3Event : WORD;
  nPushButton4Event : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValueN	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as properties [[▶ 890](#)].

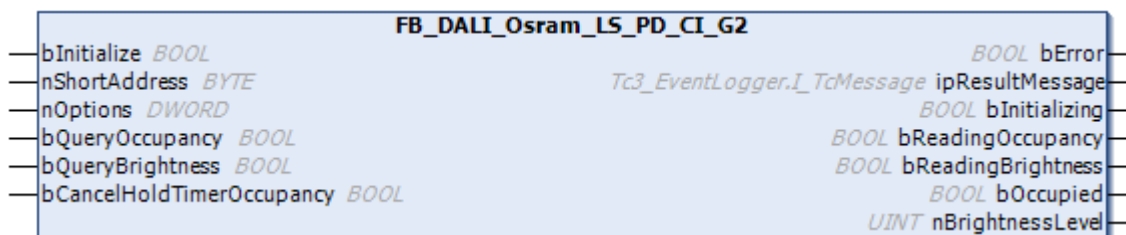
The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

**4.1.4.9.6 FB\_DALI\_Osram\_LS\_PD\_CI\_G2**



The function block represents the LS/PD CI G2 DALI sensor from Osram.

This outputs the measured brightness value (instance 1) and an occupancy signal (instance 0).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – Introduction [[▶ 542](#)]

Part 304 (light sensors) - Introduction [[▶ 561](#)]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightness    : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

**Outputs**

```
VAR_OUTPUT
  bError               : BOOL;
  ipResultMessage     : I_TcMessage;
  bInitializing       : BOOL;
  bReadingOccupancy   : BOOL;
  bReadingBrightness  : BOOL;
  bOccupied           : BOOL;
  nBrightnessLevel    : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.



### Properties

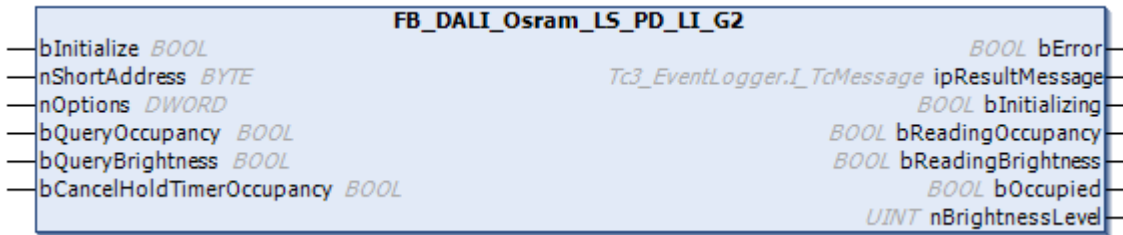
All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEvent Priority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEvent Priority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

### 4.1.4.9.7 FB\_DALI\_Osram\_LS\_PD\_LI\_G2



The function block represents the LS/PD LI G2 DALI sensor from Osram.

This outputs the measured brightness value (instance 1) and an occupancy signal (instance 0).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightness    : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .



 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

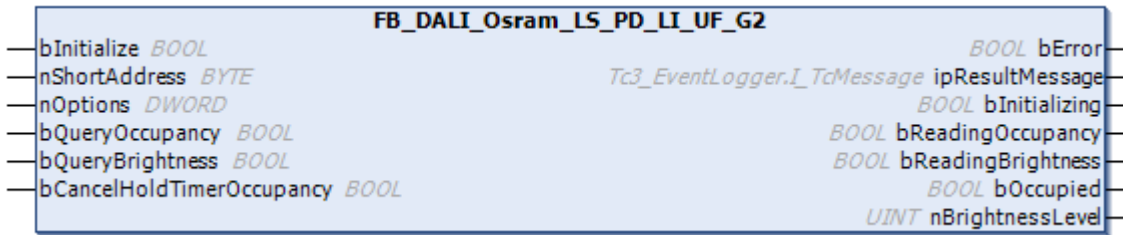
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DAL IEvent Priority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DAL IEvent Priority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALI Communication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.4.9.8 FB\_DALI\_Osram\_LS\_PD\_LI\_UF\_G2



The function block represents the LS/PD LI UF G2 DALI sensor from Osram.

This outputs the measured brightness value (instance 1) and an occupancy signal (instance 0).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightness    : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

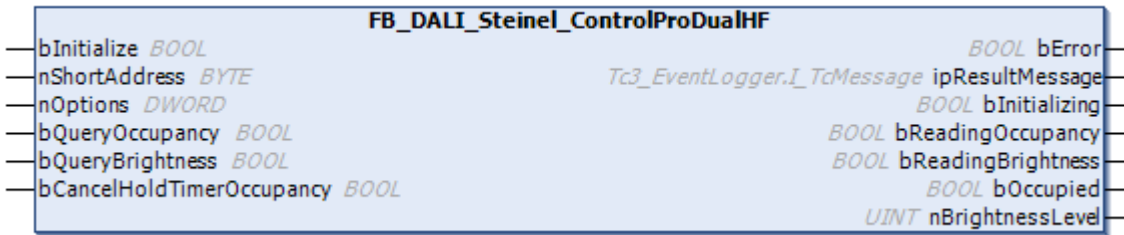
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEvent Priority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEvent Priority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

### 4.1.4.10 Steinel

#### 4.1.4.10.1 FB\_DALI\_Steinel\_ControlProDualHF



The function block represents the ControlPro Dual HF DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at `bInitialize` and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (`bError = FALSE`). If the output `bError = TRUE`, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables `bQueryOccupancy` and `bQueryBrightness`, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightness    : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

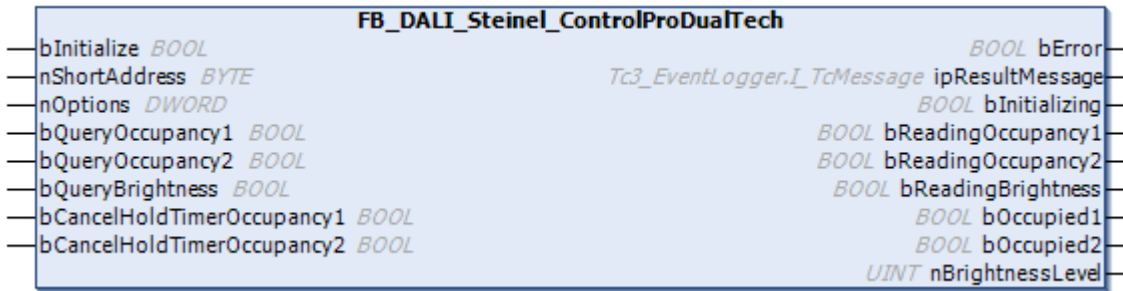
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
nDetectionRangeOccupancy	BYTE	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0



4.1.4.10.2 FB\_DALI\_Steinel\_ControlProDualTech



The function block represents the ControlPro Dual Tech US Dali sensor from Steinel.

This outputs the measured brightness value (instance 0) and two independent occupancy signals (instance 1 and instance 2).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at `bInitialize` and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (`bError = FALSE`). If the output `bError = TRUE`, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables `bQueryOccupancy1`, `bQueryOccupancy2` and `bQueryBrightness`, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy1     : BOOL := FALSE;
  bQueryOccupancy2     : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerMotion1 : BOOL := FALSE;
  bCancelHoldTimerMotion2 : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancyN	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerMotion	BOOL	A positive edge at this input prematurely ends the hold timer of the respective instance and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy1 : BOOL;
  bReadingOccupancy2 : BOOL;
  bReadingBrightness : BOOL;
  bOccupied1      : BOOL;
  bOccupied2      : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancyN	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

Some parameters of the function block are executed as properties whose values can be changed if necessary.

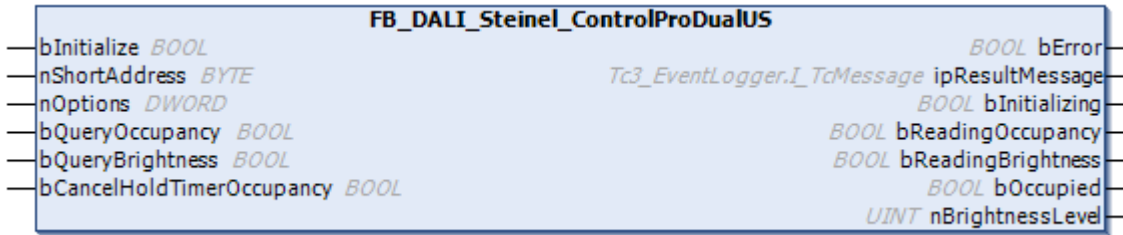
The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancyN	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancyN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancyN	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancyN	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyN	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
nDetectionRangeOccupancy	BYTE	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

### 4.1.4.10.3 FB\_DALI\_Steinel\_ControlProDualUS



The function block represents the ControlPro Dual US DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bInitializing    : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied        : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

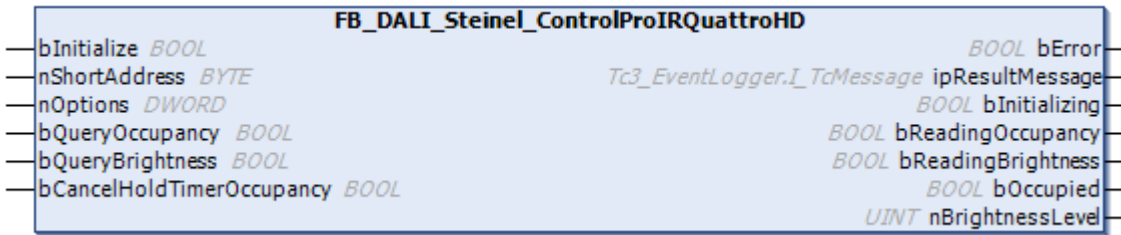
All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
nDetectionRangeOccupancy	BYTE	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

### 4.1.4.10.4 FB\_DALI\_Steinel\_ControlProIRQuattroHD



The function block represents the ControlPro IR Quattro HD DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .



## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

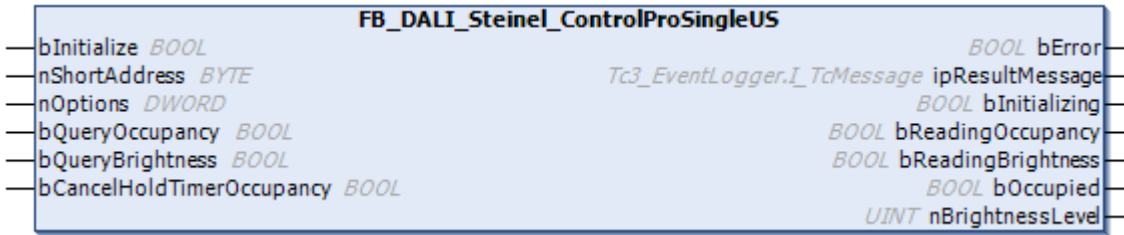


Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

### 4.1.4.10.5 FB\_DALI\_Steinel\_ControlProSingleUS



The function block represents the DALI ControlPro Single US DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
nDetectionRangeOccupancy	BYTE	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.4.10.6 FB\_DALI\_Steinel\_ControlProUS360



The function block represents the ControlPro US 360 DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy     : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bInitializing    : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied        : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

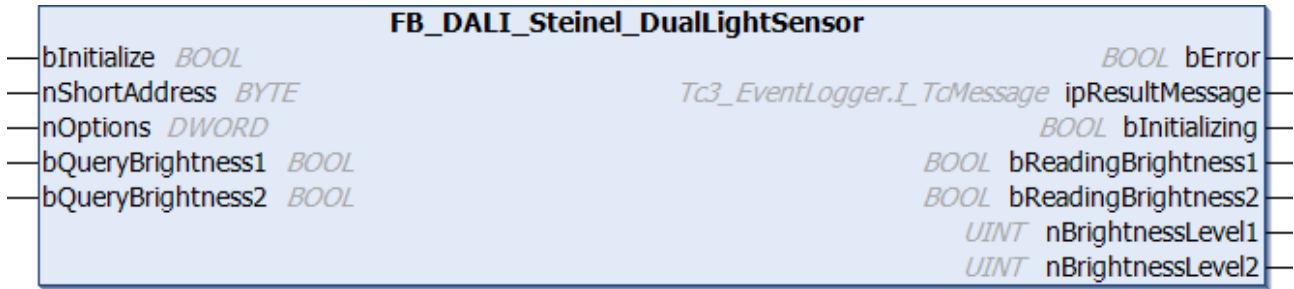
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
nDetectionRangeOccupancy	BYTE	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0



### 4.1.4.10.7 FB\_DALI\_Steinel\_DualLightSensor



The function block represents the Dual Light Sensor AP/UP DALI sensor from Steinel.

The function block has two light sensors (instance 0 and instance 1).

These two sensors each output a brightness value.

Further information on the supported instance type can be found here:

Part 304 (light sensors) - [Introduction](#) [▶ 561]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryBrightness1* and *bQueryBrightness2*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize      : BOOL := FALSE;
  nShortAddress    : BYTE := 0;
  nOptions         : DWORD := 0;
  bQueryBrightness1 : BOOL := FALSE;
  bQueryBrightness2 : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryBrightnessN	BOOL	The brightness is queried immediately by a positive edge at this input.

#### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bInitializing    : BOOL;
```



```

bReadingBrightness1      : BOOL;
bReadingBrightness2      : BOOL;
nBrightnessLevel1        : UINT;
nBrightnessLevel2        : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingBrightness N	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
nBrightnessLevelN	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnableBrightnessN	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
eEventPriorityBrightnessN	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightnessN	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightnessN	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightnessN	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightnessN	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.10.8 FB\_DALI\_Steinel\_Hallway\_IPD**

FB_DALI_Steinel_Hallway_IPD	
— bInitialize <i>BOOL</i>	<i>BOOL</i> bError
— nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
— nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
— bQueryOccupancyCombined <i>BOOL</i>	<i>BOOL</i> bReadingOccupancyCombined
— bQueryOccupancyDirectionS <i>BOOL</i>	<i>BOOL</i> bReadingOccupancyDirectionS
— bQueryOccupancyDirectionL <i>BOOL</i>	<i>BOOL</i> bReadingOccupancyDirectionL
— bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
— bQueryTemperature <i>BOOL</i>	<i>BOOL</i> bReadingTemperature
— bQueryHumidity <i>BOOL</i>	<i>BOOL</i> bReadingHumidity
— bCancelHoldTimerOccupancyCombined <i>BOOL</i>	<i>BOOL</i> bOccupiedCombined
— bCancelHoldTimerOccupancyDirectionS <i>BOOL</i>	<i>BOOL</i> bOccupiedDirectionS
— bCancelHoldTimerOccupancyDirectionL <i>BOOL</i>	<i>BOOL</i> bOccupiedDirectionL
	<i>UINT</i> nBrightnessLevel
	<i>LREAL</i> fTemperature
	<i>BYTE</i> nHumidity

The function block represents the Hallway IPD DALI sensor from Steinel.

This outputs the measured brightness value (instance 0), three occupancy signals (instances 1 to 3), a temperature value (instance 4) and a relative humidity value (instance 5).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) - [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The DALI sensor contains three occupancy sensors. One occupancy sensor (instance 2) is aligned in the direction "S" of the STEINEL lettering on the device, while the second occupancy sensor (instance 3) is aligned in the direction "L". The third occupancy sensor (instance 1) combines both sensors into a single instance.

Never activate all three instances at the same time, as this places an unnecessary load on the DALI bus. Either the two instances for direction "S" and "L" are active or the combined instance is active (default setting). Thus either the properties *nXyzDirectionS* / *nXyzDirectionL* or the properties *nXyzCombined* should be set, but not all at the same time. Setting the properties *nXyzCombined* overwrites the values of *nXyzDirectionS* / *nXyzDirectionL*.

The temperature measurement and humidity measurement are executed as Generic Inputs. These values can only be queried via a positive edge at the inputs *bQueryTemperature* and *bQueryHumidity*.

For the other instances, the reading can also be triggered directly via the inputs *bQueryBrightness*, *bQueryOccupancyCombined*, *bQueryOccupancyDirectionS* and *bQueryOccupancyDirectionL*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.

---



The function block cannot be used when using the KL6811.

---

### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancyCombined : BOOL := FALSE;
  bQueryOccupancyDirectionS : BOOL := FALSE;
  bQueryOccupancyDirectionL : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bQueryTemperature    : BOOL := FALSE;
  bQueryHumidity       : BOOL := FALSE;
  bCancelHoldTimerOccupancyCombined : BOOL := FALSE;
  bCancelHoldTimerOccupancyDirectionS : BOOL := FALSE;
  bCancelHoldTimerOccupancyDirectionL : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancyCombined	BOOL	The value of the combined occupancy sensor is queried immediately by a positive edge at this input.
bQueryOccupancyDirectionS	BOOL	The value of the occupancy sensor for direction "S" is queried immediately by a positive edge at this input.
bQueryOccupancyDirectionL	BOOL	The value of the occupancy sensor for direction "L" is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bQueryTemperature	BOOL	The temperature is queried immediately by a positive edge at this input. The DALI device does not support transmission of the temperature per event.
bQueryHumidity	BOOL	The relative humidity is queried immediately by a positive edge at this input. The DALI device does not support transmission of the relative humidity per event.
bCancelHoldTimerOccupancyCombined	BOOL	A positive edge at this input prematurely ends the hold timer of the combined occupancy sensor and resets the output <i>bOccupiedCombined</i> .
bCancelHoldTimerOccupancyDirectionS	BOOL	A positive edge at this input prematurely ends the hold timer of the occupancy sensor for direction "S" and resets the output <i>bOccupiedDirectionS</i> .
bCancelHoldTimerOccupancyDirectionL	BOOL	A positive edge at this input prematurely ends the hold timer of the occupancy sensor for direction "L" and resets the output <i>bOccupiedDirectionL</i> .

## Outputs

```

VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage      : I_TcMessage;
  bInitializing         : BOOL;
  bReadingOccupancyCombined : BOOL;
  bReadingOccupancyDirectionS : BOOL;
  bReadingOccupancyDirectionL : BOOL;
  bReadingBrightness    : BOOL;
  bReadingTemperature   : BOOL;
  bReadingHumidity      : BOOL;
  bOccupiedCombined     : BOOL;
  bOccupiedDirectionS  : BOOL;
  bOccupiedDirectionL  : BOOL;
  nBrightnessLevel     : UINT;
  fTemperature          : LREAL;
  nHumidity             : BYTE;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy Combined	BOOL	The output is TRUE as long as values of the instance of the combined movement sensor are being read by the DALI control device.
bReadingOccupancy DirectionS	BOOL	The output is TRUE as long as values of the instance of the movement sensor for direction "S" are being read by the DALI control device.
bReadingOccupancy DirectionL	BOOL	The output is TRUE as long as values of the instance of the movement sensor for direction "L" are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bReadingTemperature	BOOL	The output is TRUE as long as values of the instance of the temperature sensor are being read by the DALI control device.
bReadingHumidity	BOOL	The output is TRUE as long as values of the instance of the humidity sensor are being read by the DALI control device.
bOccupiedCombined	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the combined occupancy sensor.
bOccupiedDirectionS	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor for direction "S".
bOccupiedDirectionL	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor for direction "L".
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
fTemperature	LREAL	Measured temperature value (-5.0...60.0°C) of the temperature sensor.
nHumidity	BYTE	Measured value of the relative humidity (0...100%).



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancyCombined	BOOL	Get, Set	TRUE	A TRUE at this property enables the instance for the combined occupancy sensor. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyCombined</i> .
bEnableOccupancyDirectionS	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for motion detection in direction "S". FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyDirectionS</i> .
bEnableOccupancyDirectionL	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for motion detection in direction "L". FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyDirectionL</i> .
bEnableTemperature	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for temperature measurement. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryTemperature</i> .
bEnableHumidity	BOOL	Get, Set	FALSE	A TRUE on this property enables the instance to measure relative humidity. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryHumidity</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHysteresisBrightness	BYTE	Get, set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancyCombined	BYTE	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancyCombined	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.

Name	Type	Access	Initial value	Description
nDetectionRangeOccupancyCombined	UINT	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range). Setting this property overwrites <i>nDetectionRangeOccupancyDirectionS</i> and <i>nDetectionRangeOccupancyDirectionL</i> .
nHoldTimerOccupancyCombined	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyCombined	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancyCombined	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity). Setting this property overwrites <i>nSensitivityOccupancyDirectionS</i> and <i>nSensitivityOccupancyDirectionL</i> .
eEventPriorityOccupancyDirectionS	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancyDirectionS	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nDetectionRangeOccupancyDirectionS	UINT	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
nHoldTimerOccupancyDirectionS	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyDirectionS	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancyDirectionS	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
eEventPriorityOccupancyDirectionL	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancyDirectionL	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nDetectionRangeOccupancyDirectionL	UINT	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
nHoldTimerOccupancyDirectionL	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyDirectionL	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancyDirectionL	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).



**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_DALI from v3.10.5.0

**4.1.4.10.9 FB\_DALI\_Steinel\_Hallway\_IPD\_ECO**

FB_DALI_Steinel_Hallway_IPD_ECO			
– bInitialize <i>BOOL</i>			<i>BOOL</i> bError
– nShortAddress <i>BYTE</i>		<i>Tc3_EventLogger.I_TcMessage</i>	ipResultMessage
– nOptions <i>DWORD</i>			<i>BOOL</i> bInitializing
– bQueryOccupancyCombined <i>BOOL</i>		<i>BOOL</i>	bReadingOccupancyCombined
– bQueryOccupancyDirectionS <i>BOOL</i>		<i>BOOL</i>	bReadingOccupancyDirectionS
– bQueryOccupancyDirectionL <i>BOOL</i>		<i>BOOL</i>	bReadingOccupancyDirectionL
– bQueryBrightness <i>BOOL</i>			<i>BOOL</i> bReadingBrightness
– bCancelHoldTimerOccupancyCombined <i>BOOL</i>	<i>BOOL</i>		<i>BOOL</i> bOccupiedCombined
– bCancelHoldTimerOccupancyDirectionS <i>BOOL</i>	<i>BOOL</i>		<i>BOOL</i> bOccupiedDirectionS
– bCancelHoldTimerOccupancyDirectionL <i>BOOL</i>	<i>BOOL</i>		<i>BOOL</i> bOccupiedDirectionL
			<i>UINT</i> nBrightnessLevel

The function block represents the Hallway IPD ECO DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and three occupancy signals (instances 1 to 3).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) - [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The DALI sensor contains three occupancy sensors. One occupancy sensor (instance 2) is aligned in the direction "S" of the STEINEL lettering on the device, while the second occupancy sensor (instance 3) is aligned in the direction "L". The third occupancy sensor (instance 1) combines both sensors into a single instance.

Never activate all three instances at the same time, as this places an unnecessary load on the DALI bus. Either the two instances for direction "S" and "L" are active or the combined instance is active (default setting). Thus either the properties *nXYZDirectionS* / *nXYZDirectionL* or the properties *nXYZCombined* should be set, but not all at the same time. Setting the properties *nXYZCombined* overwrites the values of *nXYZDirectionS* / *nXYZDirectionL*.

The reading can be triggered immediately via the input variables *bQueryBrightness*, *bQueryOccupancyCombined*, *bQueryOccupancyDirectionS* and *bQueryOccupancyDirectionL*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.



 **Inputs**

```

VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancyCombined : BOOL := FALSE;
  bQueryOccupancyDirectionS : BOOL := FALSE;
  bQueryOccupancyDirectionL : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancyCombined : BOOL := FALSE;
  bCancelHoldTimerOccupancyDirectionS : BOOL := FALSE;
  bCancelHoldTimerOccupancyDirectionL : BOOL := FALSE;
END_VAR
    
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancyCombined	BOOL	The value of the combined occupancy sensor is queried immediately by a positive edge at this input.
bQueryOccupancyDirectionS	BOOL	The value of the occupancy sensor for direction "S" is queried immediately by a positive edge at this input.
bQueryOccupancyDirectionL	BOOL	The value of the occupancy sensor for direction "L" is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancyCombined	BOOL	A positive edge at this input prematurely ends the hold timer of the combined occupancy sensor and resets the output <i>bOccupiedCombined</i> .
bCancelHoldTimerOccupancyDirectionS	BOOL	A positive edge at this input prematurely ends the hold timer of the occupancy sensor for direction "S" and resets the output <i>bOccupiedDirectionS</i> .
bCancelHoldTimerOccupancyDirectionL	BOOL	A positive edge at this input prematurely ends the hold timer of the occupancy sensor for direction "L" and resets the output <i>bOccupiedDirectionL</i> .

 **Outputs**

```

VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage      : I_TcMessage;
  bInitializing        : BOOL;
  bReadingOccupancyCombined : BOOL;
  bReadingOccupancyDirectionS : BOOL;
  bReadingOccupancyDirectionL : BOOL;
  bReadingBrightness   : BOOL;
  bOccupiedCombined    : BOOL;
  bOccupiedDirectionS : BOOL;
  bOccupiedDirectionL : BOOL;
  nBrightnessLevel     : UINT;
  fTemperature         : LREAL;
  nHumidity            : BYTE;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy Combined	BOOL	The output is TRUE as long as values of the instance of the combined movement sensor are being read by the DALI control device.
bReadingOccupancy DirectionS	BOOL	The output is TRUE as long as values of the instance of the movement sensor for direction "S" are being read by the DALI control device.
bReadingOccupancy DirectionL	BOOL	The output is TRUE as long as values of the instance of the movement sensor for direction "L" are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bReadingTemperature	BOOL	The output is TRUE as long as values of the instance of the temperature sensor are being read by the DALI control device.
bOccupiedDirection S	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor for direction "S".
bOccupiedDirectionL	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor for direction "L".
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
fTemperature	LREAL	Measured temperature value (-5.0...60.0°C) of the temperature sensor.
nHumidity	BYTE	Measured value of the relative humidity (0...100%).



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

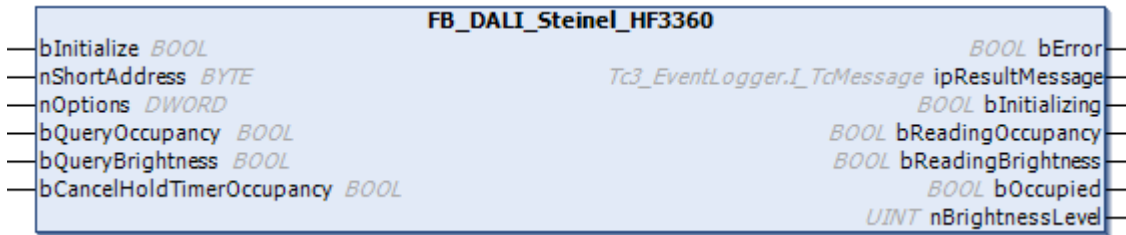
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancyCombined	BOOL	Get, Set	TRUE	A TRUE at this property enables the instance for the combined occupancy sensor. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyCombined</i> .
bEnableOccupancyDirectionS	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for motion detection in direction "S". FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyDirectionS</i> .
bEnableOccupancyDirectionL	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for motion detection in direction "L". FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyDirectionL</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHysteresisBrightness	BYTE	Get, set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancyCombined	BYTE	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancyCombined	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nDetectionRangeOccupancyCombined	UINT	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range). Setting this property overwrites <i>nDetectionRangeOccupancyDirectionS</i> and <i>nDetectionRangeOccupancyDirectionL</i> .
nHoldTimerOccupancyCombined	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyCombined	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.

Name	Type	Access	Initial value	Description
nSensitivityOccupancyCombined	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity). Setting this property overwrites <i>nSensitivityOccupancyDirectionS</i> and <i>nSensitivityOccupancyDirectionL</i> .
eEventPriorityOccupancyDirectionS	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancyDirectionS	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nDetectionRangeOccupancyDirectionS	UINT	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
nHoldTimerOccupancyDirectionS	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyDirectionS	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancyDirectionS	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
eEventPriorityOccupancyDirectionL	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancyDirectionL	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nDetectionRangeOccupancyDirectionL	UINT	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
nHoldTimerOccupancyDirectionL	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyDirectionL	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancyDirectionL	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_DALI from v3.10.5.0

4.1.4.10.10 FB\_DALI\_Steinel\_HF3360



The function block represents the HF 3360 AP/UP DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

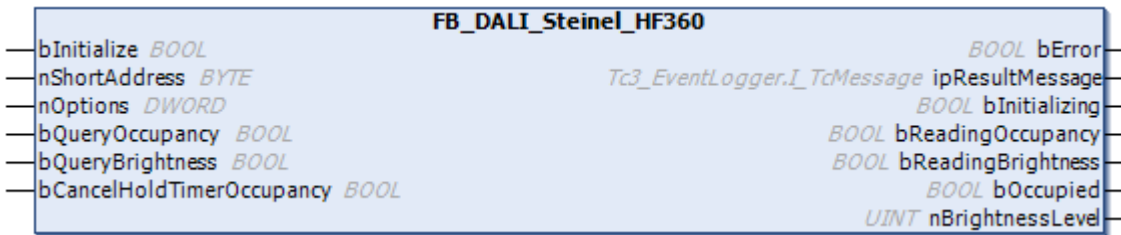
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
nDetectionRangeOccupancy	BYTE	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0



### 4.1.4.10.11 FB\_DALI\_Steinel\_HF360



The function block represents the HF 360 DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .



 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
nDetectionRangeOccupancy	BYTE	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.4.10.12 FB\_DALI\_Steinel\_HF360\_II\_IPD

FB_DALI_Steinel_HF360_II_IPD	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryOccupancy <i>BOOL</i>	<i>BOOL</i> bReadingOccupancy
bQueryBrightness <i>BOOL</i>	<i>BOOL</i> bReadingBrightness
bQueryTemperature <i>BOOL</i>	<i>BOOL</i> bReadingTemperature
bQueryHumidity <i>BOOL</i>	<i>BOOL</i> bReadingHumidity
bCancelHoldTimerOccupancy <i>BOOL</i>	<i>BOOL</i> bOccupied
	<i>UINT</i> nBrightnessLevel
	<i>LREAL</i> fTemperature
	<i>BYTE</i> nHumidity

The function block represents the HF 360 II IPD DALI sensor from Steinel.

This outputs the measured brightness value (instance 0), an occupancy signal (instance 1), a temperature value (instance 2) and a value for the relative humidity (instance 3).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) - [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The temperature measurement and humidity measurement are executed as Generic Inputs. These values can only be queried via a positive edge at the inputs *bQueryTemperature* and *bQueryHumidity*.

For the other instances, the reading can also be triggered directly via the inputs *bQueryBrightness* and *bQueryOccupancy*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```

VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bQueryTemperature    : BOOL := FALSE;
  bQueryHumidity       : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
    
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bQueryTemperature	BOOL	The temperature is queried immediately by a positive edge at this input. The DALI device does not support transmission of the temperature per event.
bQueryHumidity	BOOL	The relative humidity is queried immediately by a positive edge at this input. The DALI device does not support transmission of the relative humidity per event.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

### Outputs

```

VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage      : I_TcMessage;
  bInitializing        : BOOL;
  bReadingOccupancy    : BOOL;
  bReadingBrightness   : BOOL;
  bReadingTemperature  : BOOL;
  bReadingHumidity     : BOOL;
  bOccupied            : BOOL;
  nBrightnessLevel     : UINT;
  fTemperature         : LREAL;
  nHumidity            : BYTE;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bReadingTemperature	BOOL	The output is TRUE as long as values of the instance of the temperature sensor are being read by the DALI control device.
bReadingHumidity	BOOL	The output is TRUE as long as values of the instance of the humidity sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
fTemperature	LREAL	Measured temperature value (-5.0...60.0°C) of the temperature sensor.
nHumidity	BYTE	Measured value of the relative humidity (0...100%).



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

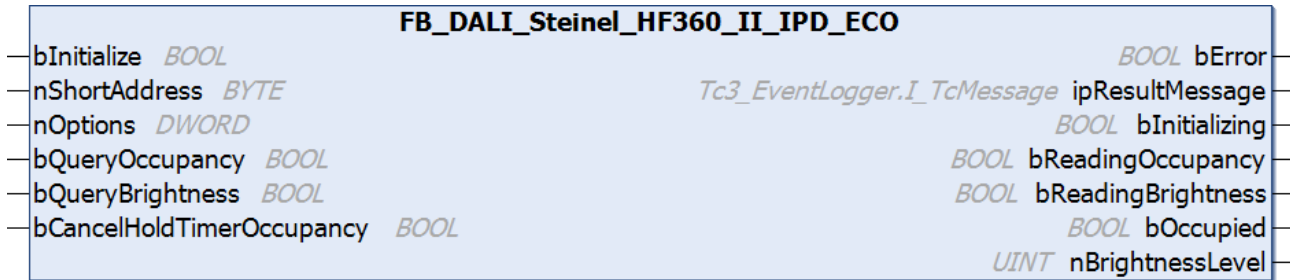
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	A TRUE at this property enables the instance for the occupancy sensor. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnableTemperature	BOOL	Get, Set	FALSE	TRUE at this property enables the instance for temperature measurement. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryTemperature</i> .
bEnableHumidity	BOOL	Get, Set	FALSE	A TRUE on this property enables the instance to measure relative humidity. FALSE at this property causes the instance not to be configured during initialization. The actual value of the instance can only be queried via the input <i>bQueryHumidity</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHysteresisBrightness	BYTE	Get, set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	BYTE	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nDetectionRangeOccupancy	UINT	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_DALI from v3.10.5.0

**4.1.4.10.13 FB\_DALI\_Steinel\_HF360\_II\_IPD\_ECO**



The function block represents the HF 360 II IPD ECO DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```

VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
    
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

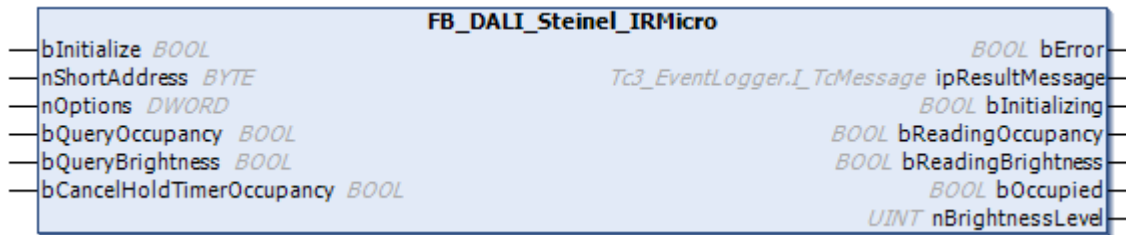


Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	A TRUE at this property enables the instance for the occupancy sensor. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHysteresisBrightness	BYTE	Get, set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	BYTE	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nDetectionRangeOccupancy	UINT	Get, Set	100 %	Setting of the detection range of the occupancy sensor from 0 % (small detection range) to 100 % (large detection range).
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_DALI from v3.10.5.0

### 4.1.4.10.14 FB\_DALI\_Steinel\_IRMicro



The function block represents the IR Micro DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness      : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

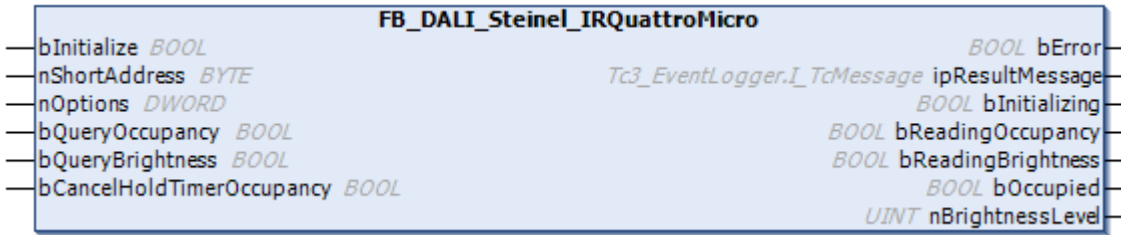
All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.4.10.15 FB\_DALI\_Steinel\_IRQuattroMicro



☉ The function block represents the IR Quattro Micro DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness      : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

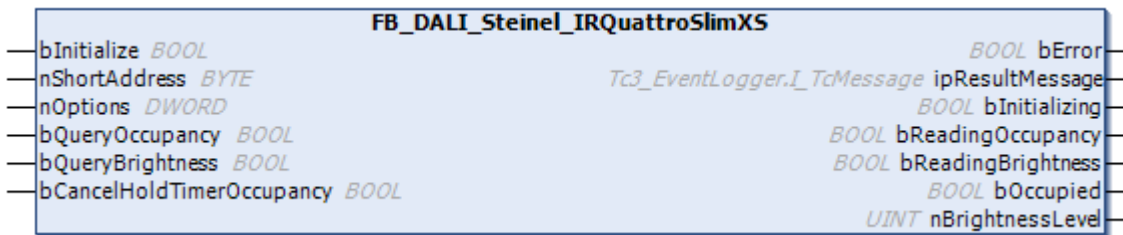
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0



### 4.1.4.10.16 FB\_DALI\_Steinel\_IRQuattroSlimXS



The function block represents the IR Quattro Slim XS DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .



 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

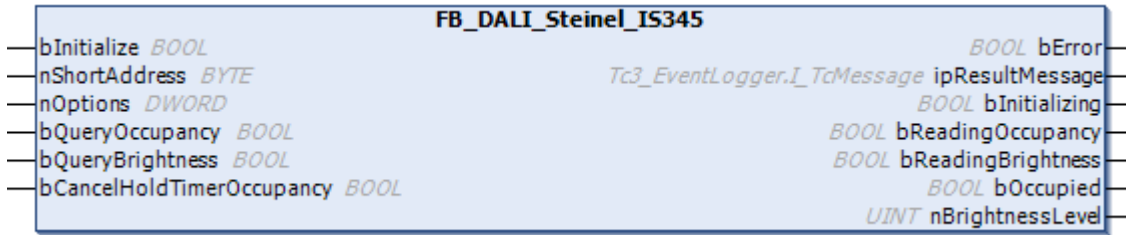
All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.4.10.17 FB\_DALI\_Steinel\_IS345



The function block represents the IS 345 AP/UP DALI<sup>®</sup> sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightness    : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

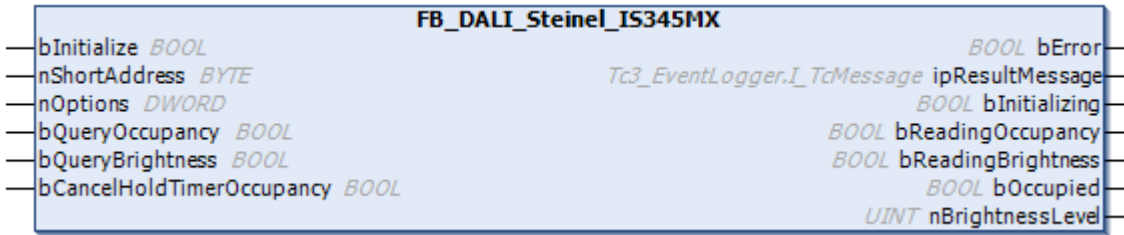
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

### 4.1.4.10.18 FB\_DALI\_Steinel\_IS345MX



The function block represents the IS 345 MX AP/UP DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage  : I_TcMessage;
  bInitializing    : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied        : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

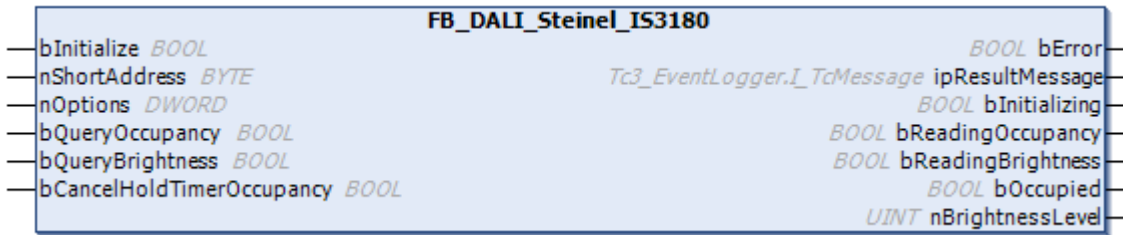
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0



4.1.4.10.19 FB\_DALI\_Steinel\_IS3180



The function block represents the IS 3180 AP/UP DALI<sup>®</sup> sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL := FALSE;
  bQueryBrightness     : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied        : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

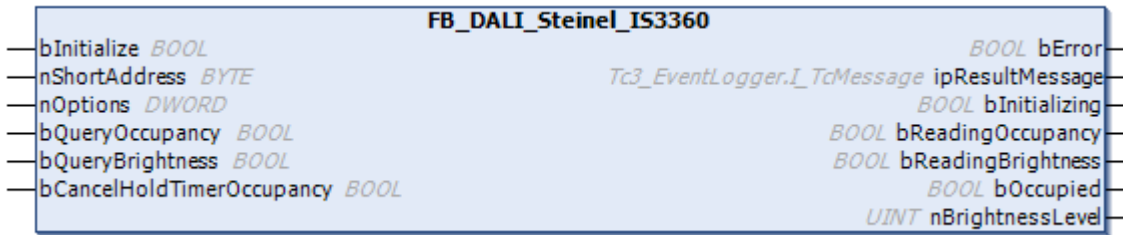
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

### 4.1.4.10.20 FB\_DALI\_Steinel\_IS3360



The function block represents the IS 3360 AP/UP DALI<sup>®</sup> sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[▶ 542\]](#)

Part 304 (light sensors) - [Introduction \[▶ 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL := FALSE;
  bQueryBrightness    : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

 **Properties**

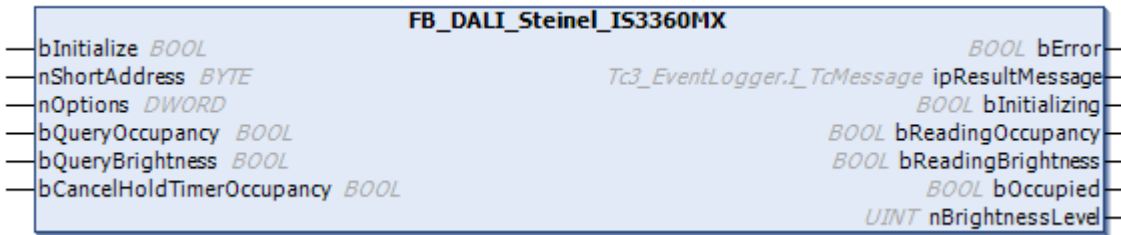
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.4.10.21 FB\_DALI\_Steinel\_IS3360MX



The function block represents the IS 3360 MX AP/UP DALI sensor from Steinel.

This outputs the measured brightness value (instance 0) and an occupancy signal (instance 1).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) - [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL := FALSE;
  bQueryBrightness    : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].



Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	20 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	100 %	Setting of the sensitivity of the occupancy sensor from 0 % (low sensitivity) to 100 % (high sensitivity).
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

#### 4.1.4.11 Sunricher

##### 4.1.4.11.1 FB\_DALI\_Sunricher\_PushButtonCoupler

FB_DALI_Sunricher_PushButtonCoupler	
bInitialize <i>BOOL</i>	<i>BOOL</i> bError
nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
bQueryInputValue1 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue1
bQueryInputValue2 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue2
bQueryInputValue3 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue3
bQueryInputValue4 <i>BOOL</i>	<i>BOOL</i> bReadingInputValue4
	<i>WORD</i> nPushButton1Event
	<i>WORD</i> nPushButton2Event
	<i>WORD</i> nPushButton3Event
	<i>WORD</i> nPushButton4Event

The function block represents the Push Button Coupler DALI push button interface from Sunricher.

Up to four push buttons (instances 0 to 3) can be connected directly via the device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress       : BYTE := 0;
  nOptions            : DWORD := 0;
  bQueryInputValue1  : BOOL := FALSE;
  bQueryInputValue2  : BOOL := FALSE;
  bQueryInputValue3  : BOOL := FALSE;
  bQueryInputValue4  : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

 **Outputs**

```

VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  bReadingInputValue3 : BOOL;
  bReadingInputValue4 : BOOL;
  nPushButton1Event : WORD;
  nPushButton2Event : WORD;
  nPushButton3Event : WORD;
  nPushButton4Event : WORD;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValueN	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as properties [► 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

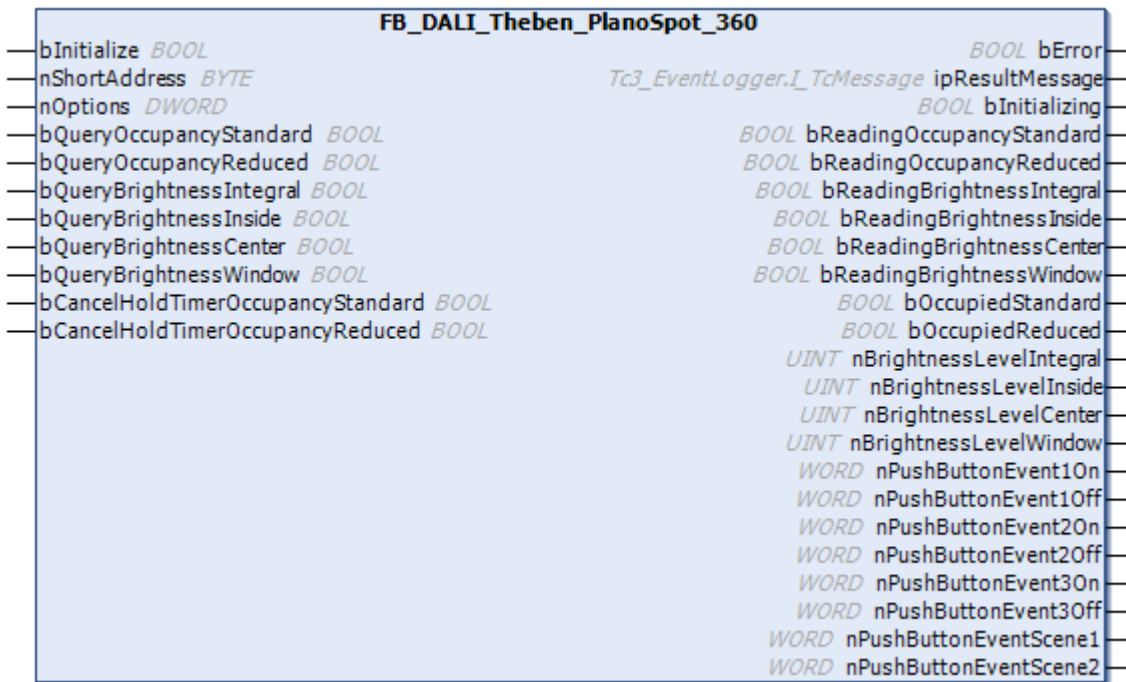
Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.22	Tc3_DALI from v3.9.0.0

4.1.4.12 Theben

4.1.4.12.1 FB\_DALI\_Theben\_PlanoSpot\_360



The function block represents the PlanoSpot 360 DALI sensor from Theben.

This outputs two occupancy signals (instances 0 and 1), four measured brightness values (instances 2 to 5) and eight push button signals (instances 6 to 13).

The push button instances can be used via an infrared remote control from the vendor.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancyStandard*, *bQueryOccupancyReduced*, *bQueryBrightnessIntegral*, *bQueryBrightnessInside*, *bQueryBrightnessCenter* and *bQueryBrightnessWindow*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancyStandard : BOOL := FALSE;
  bQueryOccupancyReduced  : BOOL := FALSE;
  bQueryBrightnessIntegral : BOOL := FALSE;
  bQueryBrightnessInside  : BOOL := FALSE;
  bQueryBrightnessCenter  : BOOL := FALSE;
  bQueryBrightnessWindow  : BOOL := FALSE;
  bCancelHoldTimerOccupancyStandard : BOOL := FALSE;
  bCancelHoldTimerOccupancyReduced  : BOOL := FALSE;
END_VAR
    
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancyN	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightnessN	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancyN	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```

VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingOccupancyStandard : BOOL;
  bReadingOccupancyReduced  : BOOL;
  bReadingBrightnessIntegral : BOOL;
  bReadingBrightnessInside  : BOOL;
  bReadingBrightnessCenter  : BOOL;
  bReadingBrightnessWindow  : BOOL;
  bOccupiedStandard      : BOOL;
  bOccupiedReduced      : BOOL;
  nBrightnessLevelIntegral : UINT;
  nBrightnessLevelInside  : UINT;
  nBrightnessLevelCenter  : UINT;
  nBrightnessLevelWindow  : UINT;
  nPushButtonEvent1On     : WORD;
  nPushButtonEvent1Off    : WORD;
  nPushButtonEvent2On     : WORD;
  nPushButtonEvent2Off    : WORD;
  nPushButtonEvent3On     : WORD;
  nPushButtonEvent3Off    : WORD;
  nPushButtonEventScene1  : WORD;
  nPushButtonEventScene2  : WORD;
END_VAR
    
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancyN	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightnessN	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupiedN	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevelN	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
nPushButtonEventN	WORD	The output outputs the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[► 840\]](#) can be used to convert the state of a push button into a BOOL variable.



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.



Name	Type	Access	Initial value	Description
bEnableBrightnessN	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightnessN</i> .
bEnableOccupancyN	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyN</i> .
bEnablePushButtonN	BOOL	Get, Set	FALSE	Push button input that can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
eEventPriorityBrightnessN	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nRoomCorrectionFactorN	BYTE	Get, Set	30	The room correction factor (5...200) specifies the difference in brightness measurement on the ceiling and at the workplace. It can be calculated using a luxmeter with the following equation:  $RoomCorrectionFactor = \frac{Brightness\ under\ ceiling}{Brightness\ on\ work\ surface}$ <p>The value can also be determined by a remote control and with an appropriate app. The preset value of 0.3 (30) is suitable for most applications.</p>
nReportTimerBrightnessN	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightnessN	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightnessN	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightnessN	BYTE	Get, Set	163	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancyN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancyN	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancyN	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyN	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	3	Adjustment of the sensitivity of both occupancy sensor instances from 0 (low sensitivity) to 5 (high sensitivity).



Name	Type	Access	Initial value	Description
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time [0, <i>tDoubleMin</i> ...2000 ms] to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.12.2 FB\_DALI\_Theben\_thePassa\_P360**

FB_DALI_Theben_thePassa_P360	
— bInitialize <i>BOOL</i>	<i>BOOL</i> bError
— nShortAddress <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> ipResultMessage
— nOptions <i>DWORD</i>	<i>BOOL</i> bInitializing
— bQueryOccupancyOverall <i>BOOL</i>	<i>BOOL</i> bReadingOccupancyOverall
— bQueryOccupancyZone1 <i>BOOL</i>	<i>BOOL</i> bReadingOccupancyZone1
— bQueryOccupancyZone2 <i>BOOL</i>	<i>BOOL</i> bReadingOccupancyZone2
— bQueryBrightnessZone1and2 <i>BOOL</i>	<i>BOOL</i> bReadingBrightnessZone1and2
— bQueryBrightnessZone1 <i>BOOL</i>	<i>BOOL</i> bReadingBrightnessZone1
— bQueryBrightnessZone2 <i>BOOL</i>	<i>BOOL</i> bReadingBrightnessZone2
— bCancelHoldTimerOccupancyOverall <i>BOOL</i>	<i>BOOL</i> bOccupiedOverall
— bCancelHoldTimerOccupancyZone1 <i>BOOL</i>	<i>BOOL</i> bOccupiedZone1
— bCancelHoldTimerOccupancyZone2 <i>BOOL</i>	<i>BOOL</i> bOccupiedZone2
	<i>UINT</i> nBrightnessLevelZone1and2
	<i>UINT</i> nBrightnessLevelZone1
	<i>UINT</i> nBrightnessLevelZone2
	<i>WORD</i> nPushButtonEvent1On
	<i>WORD</i> nPushButtonEvent1Off
	<i>WORD</i> nPushButtonEvent2On
	<i>WORD</i> nPushButtonEvent2Off
	<i>WORD</i> nPushButtonEvent3On
	<i>WORD</i> nPushButtonEvent3Off
	<i>WORD</i> nPushButtonEventScene1
	<i>WORD</i> nPushButtonEventScene2

The function block represents thePassa P360 DALI sensor from Theben.

This outputs three occupancy signals (instances 0 to 2), three measured brightness values (instances 3 to 5) and eight push button signals (instances 6 to 13).

The push button instances can be used via an infrared remote control from the vendor.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancyOverall*, *bQueryOccupancyZone1*, *bQueryOccupancyZone2*, *bQueryBrightnessZone1and2*, *bQueryBrightnessZone1* and *bQueryBrightnessZone2*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancyOverall : BOOL := FALSE;
  bQueryOccupancyZone1  : BOOL := FALSE;
  bQueryOccupancyZone2  : BOOL := FALSE;
  bQueryBrightnessZone1and2 : BOOL := FALSE;
  bQueryBrightnessZone1 : BOOL := FALSE;
  bQueryBrightnessZone2 : BOOL := FALSE;
  bCancelHoldTimerOccupancyOverall : BOOL := FALSE;
  bCancelHoldTimerOccupancyZone1 : BOOL := FALSE;
  bCancelHoldTimerOccupancyZone2 : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancyN	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightnessN	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancyN	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
  bReadingOccupancyStandard : BOOL;
  bReadingOccupancyReduced : BOOL;
```

```

bReadingBrightnessIntegral      : BOOL;
bReadingBrightnessInside       : BOOL;
bReadingBrightnessCenter       : BOOL;
bReadingBrightnessWindow       : BOOL;
bOccupiedStandard              : BOOL;
bOccupiedReduced               : BOOL;
nBrightnessLevelIntegral       : UINT;
nBrightnessLevelInside         : UINT;
nBrightnessLevelCenter         : UINT;
nBrightnessLevelWindow        : UINT;
nPushButtonEvent1On            : WORD;
nPushButtonEvent1Off          : WORD;
nPushButtonEvent2On            : WORD;
nPushButtonEvent2Off          : WORD;
nPushButtonEvent3On            : WORD;
nPushButtonEvent3Off          : WORD;
nPushButtonEventScene1        : WORD;
nPushButtonEventScene2        : WORD;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancyN	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightnessN	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupiedN	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevelN	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
nPushButtonEventN	WORD	The output outputs the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block `FB_DALI_ToPushButtonState` [► 840] can be used to convert the state of a push button into a BOOL variable.

### Properties

All parameters that are written to the DALI control device via `bInitialize` are available as properties [► 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

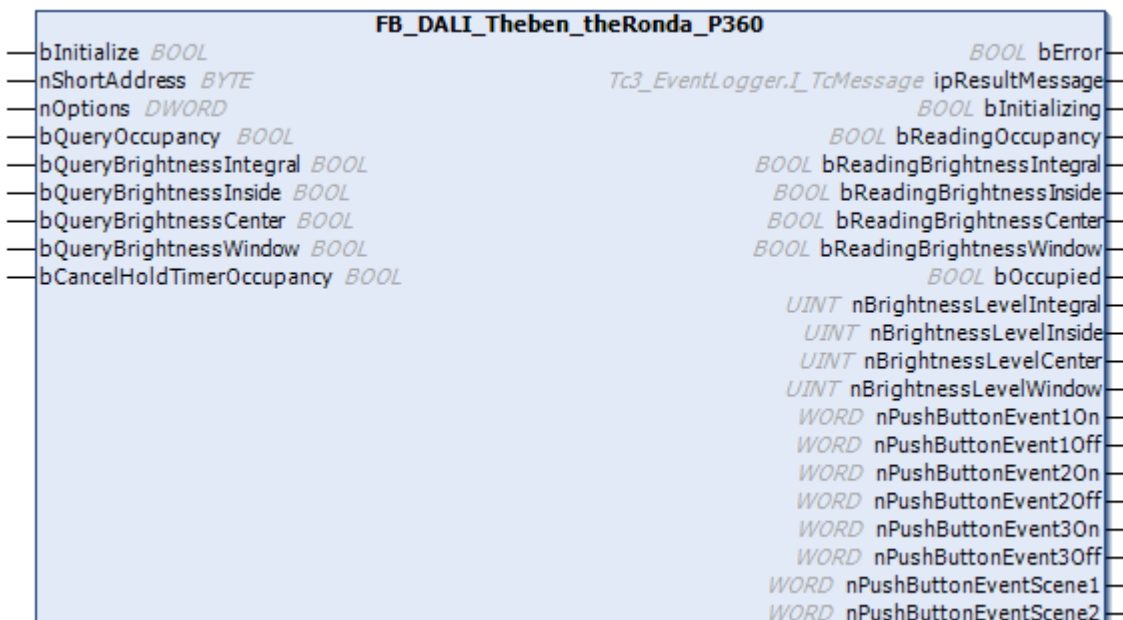
Name	Type	Access	Initial value	Description
bEnableBrightnessN	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightnessN</i> .
bEnableOccupancyN	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyN</i> .
bEnablePushButtonN	BOOL	Get, Set	FALSE	Push button input that can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
eEventPriorityBrightnessN	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nRoomCorrectionFactorN	BYTE	Get, Set	30	The room correction factor (5...200) specifies the difference in brightness measurement on the ceiling and at the workplace. It can be calculated using a luxmeter with the following equation:  $RoomCorrectionFactor = \frac{Brightness\ under\ ceiling}{Brightness\ on\ work\ surface}$ <p>The value can also be determined by a remote control and with an appropriate app.</p> <p>The preset value of 0.3 (30) is suitable for most applications.</p>
nReportTimerBrightnessN	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightnessN	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightnessN	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightnessN	BYTE	Get, Set	163	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancyN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancyN	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancyN	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancyN	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.

Name	Type	Access	Initial value	Description
nSensitivityOccupancy	BYTE	Get, Set	3	Adjustment of the sensitivity of both occupancy sensor instances from 0 (low sensitivity) to 5 (high sensitivity).
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.12.3 FB\_DALI\_Theben\_theRonda\_P360**



The function block represents theRonda P360 DALI sensor from Theben.

This outputs one occupancy signal (instance 0), four measured brightness values (instances 1 to 5) and eight push button signals (instances 5 to 12).

The push button instances can be used via an infrared remote control from the vendor.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy*, *bQueryBrightnessIntegral*, *bQueryBrightnessInside*, *bQueryBrightnessCenter* and *bQueryBrightnessWindow* , even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightnessIntegral : BOOL := FALSE;
  bQueryBrightnessInside : BOOL := FALSE;
  bQueryBrightnessCenter : BOOL := FALSE;
  bQueryBrightnessWindow : BOOL := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightnessN	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

**Outputs**

```
VAR_OUTPUT
  bError               : BOOL;
  ipResultMessage     : I_TcMessage;
  bInitializing       : BOOL;
  bReadingOccupancyStandard : BOOL;
  bReadingOccupancyReduced : BOOL;
```



```

bReadingBrightnessIntegral      : BOOL;
bReadingBrightnessInside       : BOOL;
bReadingBrightnessCenter       : BOOL;
bReadingBrightnessWindow       : BOOL;
bOccupiedStandard              : BOOL;
bOccupiedReduced               : BOOL;
nBrightnessLevelIntegral       : UINT;
nBrightnessLevelInside         : UINT;
nBrightnessLevelCenter         : UINT;
nBrightnessLevelWindow         : UINT;
nPushButtonEvent1On           : WORD;
nPushButtonEvent1Off          : WORD;
nPushButtonEvent2On           : WORD;
nPushButtonEvent2Off          : WORD;
nPushButtonEvent3On           : WORD;
nPushButtonEvent3Off          : WORD;
nPushButtonEventScene1        : WORD;
nPushButtonEventScene2        : WORD;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancyN	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightnessN	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupiedN	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevelN	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
nPushButtonEventN	WORD	The output outputs the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.



The function block `FB_DALI_ToPushButtonState` [► 840] can be used to convert the state of a push button into a BOOL variable.

### Properties

All parameters that are written to the DALI control device via `bInitialize` are available as properties [► 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnableBrightnessN	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightnessN</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancyN</i> .
bEnablePushButtonN	BOOL	Get, Set	FALSE	Push button input that can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
eEventPriorityBrightnessN	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nRoomCorrectionFactorN	BYTE	Get, Set	30	<p>The room correction factor (5...200) specifies the difference in brightness measurement on the ceiling and at the workplace. It can be calculated using a luxmeter with the following equation:</p> $RoomCorrectionFactor = \frac{Brightness\ under\ ceiling}{Brightness\ on\ work\ surface}$ <p>The value can also be determined by a remote control and with an appropriate app.</p> <p>The preset value of 0.3 (30) is suitable for most applications.</p>
nReportTimerBrightnessN	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightnessN	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightnessN	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightnessN	BYTE	Get, Set	163	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	3	Adjustment of the sensitivity of both occupancy sensor instances from 0 (low sensitivity) to 5 (high sensitivity).

Name	Type	Access	Initial value	Description
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.12.4 FB\_DALI\_Theben\_theRonda\_S360**



The function block represents the theRonda S360 DALI sensor from Theben.

This outputs an occupancy signal (instance 0), a measured brightness value (instance 1) and eight push button signals (instances 5 to 12).

The push button instances can be used via an infrared remote control from the vendor.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingOccupancy     : BOOL;
  bReadingBrightness    : BOOL;
  bOccupied             : BOOL;
  nBrightnessLevel      : UINT;
  nPushButtonEvent1On   : WORD;
  nPushButtonEvent1Off  : WORD;
  nPushButtonEvent2On   : WORD;
  nPushButtonEvent2Off  : WORD;
  nPushButtonEvent3On   : WORD;
  nPushButtonEvent3Off  : WORD;
  nPushButtonEventScene1 : WORD;
  nPushButtonEventScene2 : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
nPushButtonEventN	WORD	The output outputs the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[► 840\]](#) can be used to convert the state of a push button into a BOOL variable.



**Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

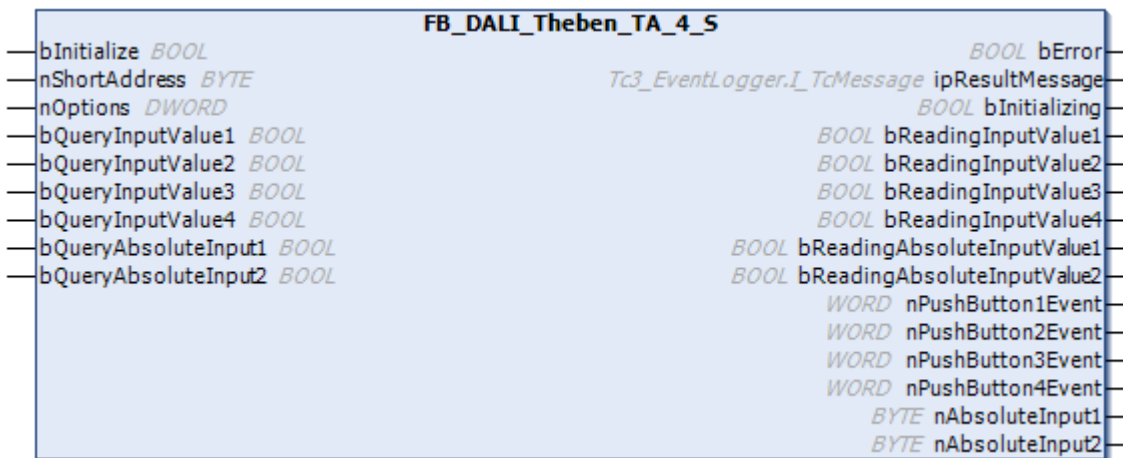
Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnablePushButtonN	BOOL	Get, Set	FALSE	Push button input that can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
eEventPriorityBrightness	E_DALI EventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nRoomCorrectionFactor	BYTE	Get, Set	30	The room correction factor (5...200) specifies the difference in brightness measurement on the ceiling and at the workplace. It can be calculated using a luxmeter with the following equation:  $RoomCorrectionFactor = \frac{Brightness\ under\ ceiling}{Brightness\ on\ work\ surface}$ <p>The value can also be determined by a remote control and with an appropriate app. The preset value of 0.3 (30) is suitable for most applications.</p>
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightnessN	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	163	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALI EventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
nSensitivityOccupancy	BYTE	Get, Set	3	Adjustment of the sensitivity of both occupancy sensor instances from 0 (low sensitivity) to 5 (high sensitivity).

Name	Type	Access	Initial value	Description
eEventPriorityPushButtonN	E_DALI EventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
ipDALICommunication	I_DALI Communication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.12.5 FB\_DALI\_Theben\_TA\_4\_S**



This function block represents the DALI push button interface TA 4 S DALI-2 from Theben.

Up to four push buttons (instances 2 to 5) and two sliders/temperature sensors (instances 0 and 1) can be connected via the device.

The push button inputs are configured as NO contacts in the delivery state. You can change the contact type to break contact (NC) via the properties [► 805].

You can set the parameters of the two absolute inputs in the properties [► 805]. 100 kOhm variable resistances can be directly connected in the delivery state.

For further information on configuration, please refer to the vendor's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) – Introduction [► 511]

Part 302 (absolute input devices) - Introduction [▶ 532]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

All inputs can be queried immediately via a positive edge at *bQueryInputValueN* and *bQueryAbsoluteInputN*, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress         : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryInputValue1    : BOOL  := FALSE;
  bQueryInputValue2    : BOOL  := FALSE;
  bQueryInputValue3    : BOOL  := FALSE;
  bQueryInputValue4    : BOOL  := FALSE;
  bQueryAbsoluteInput1 : BOOL  := FALSE;
  bQueryAbsoluteInput2 : BOOL  := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	Immediate query of the values at the push button inputs 1 to 4.
bQueryAbsoluteInputN	BOOL	Immediate query of the values at the absolute value inputs 1 to 2.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingInputValue1   : BOOL;
  bReadingInputValue2   : BOOL;
  bReadingInputValue3   : BOOL;
  bReadingInputValue4   : BOOL;
  bReadingAbsoluteInputValue1 : BOOL;
  bReadingAbsoluteInputValue2 : BOOL;
  nPushButton1Event     : WORD;
  nPushButton2Event     : WORD;
  nPushButton3Event     : WORD;
  nPushButton4Event     : WORD;
  nAbsoluteInput1       : BYTE;
  nAbsoluteInput2       : BYTE;
END_VAR
```



Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingValueN	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.

Name	Type	Description
nAbsoluteInputN	BYTE	Outputs that contain the measured values, [°C] or [Ω], of the connected variable resistances. Please refer to the manufacturer's documentation for the TA 4 S DALI-2 for the appropriate scaling of the output value.

### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[▶ 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	Push button input 1 to 4, which can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
eEventPriorityInputN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerAbsoluteInputN	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nReportTimerAbsoluteInputN	BYTE	Get, Set	0 s	Time [s] after which the pending event of the absolute encoder is repeated if no other event has occurred in the meantime.
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
nContactTypePushButtonN	BYTE	Get, Set	16#00	Push button contact type: 16#00: NO contact (delivery state) 16#FF: NC contact
bEnableAbsoluteInputN	BOOL	Get, Set	FALSE	Absolute value input 1 or 2, which can each be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryAbsoluteInputN</i> .
eEventPriorityAbsoluteInputN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerAbsoluteInputN	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nReportTimerAbsoluteInputN	BYTE	Get, Set	0 s	Time [s] after which the pending event of the absolute encoder is repeated if no other event has occurred in the meantime.
nSensorTypeAbsoluteInputN	BYTE	Get, Set	16#03	Sensor type: 16#00: Theben sensors (9070321, 9070459, 9070489, 9070496). 16#01: Theben sensor (9070191). 16#02: any 100 kΩ NTC with known B or Beta [K] (see also value range parameter Beta Low / Beta High below). 16#03: 100 kΩ variable resistance (delivery state).

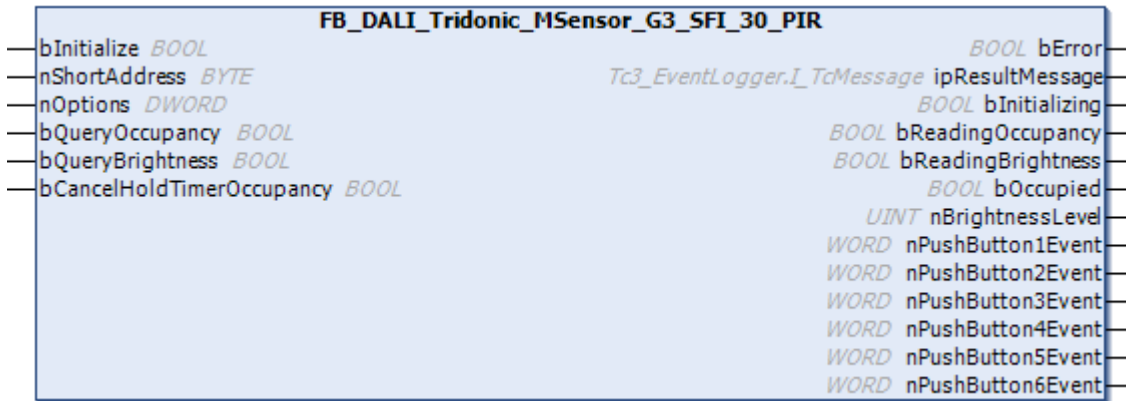
Name	Type	Access	Initial value	Description
nBetaHighAbsoluteInputN	BYTE	Get, Set	16#11	If the sensor type 16#02 was selected, the associated B or Beta [K] should be set here. The B or Beta is a four-digit number in [K], which can be found on the data sheet for the NTC sensor. For storage in the memory bank, the four-digit number must first be converted to hex format and then stored in the Beta Low and Beta High bytes. Example: B or Beta = 4580, this corresponds to 16#11E4 • Beta Low = 16#E4, delivery state • Beta High = 16#11, delivery state • The adjustment range is 16#00...16#FF.
nBetaLowAbsoluteInputN	BYTE	Get, Set	16#E4	See <i>nBetaHighAbsoluteInputN</i>
nFilterAbsoluteInputN	BYTE	Get, Set	16#00	A filter can be set for noise suppression. The adjustment range is 16#00...16#0B. 16#00: each measured value is transmitted unchanged (delivery state) 16#01: average of 2 measured values (2 <sup>1</sup> ). 16#02: average of 4 measured values (2 <sup>2</sup> ). 16#03: average of 8 measured values (2 <sup>3</sup> ). ... 16#09: average of 512 measured values (2 <sup>9</sup> ). 16#0A: average of 1024 measured values (2 <sup>10</sup> ). 16#0B: average of 2048 measured values (2 <sup>11</sup> ). <b>Attention:</b> The filter delays the measurement!
nHysteresisAbsoluteInputN	BYTE	Get, Set	16#00	Hysteresis value for the avoidance of frequent events. The adjustment range is 16#00...16#FF. This corresponds to a value range of 0.1 °C...25.5 °C for sensor types 16#00...16#02 or a value range of 100 Ω...25.5 kΩ for sensor type 16#03. The hysteresis step size is equivalent to 0.1 °C or 100 Ω.
nOffsetAbsoluteInputN	BYTE	Get, Set	16#80	The adjustment range of the offset is 16#01...16#FF. This corresponds to a value range of -12.7 °C...+12.7 °C for sensor types 16#00...16#02 or a value range of -12.7 kΩ...+12.7 kΩ for sensor type 16#03. The offset step size is equivalent to 0.1 °C or 100 Ω. The 16#80s in the delivery state correspond to 0 °C or 0 Ω respectively.
nRangeAbsoluteInputN	BYTE	Get, Set	16#00	Setting the parameter range: 16#00: Temperature sensors with a positive temperature range (delivery state). 16#01: Temperature sensors with a negative temperature range.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.20	Tc3_DALI from v3.8.0.0

### 4.1.4.13 Tridonic

#### 4.1.4.13.1 FB\_DALI\_Tridonic\_MSensorG3\_SFI\_30\_PIR



The function block represents the MSensor G3 SFI 30 PIR 5DPI, MSensor G3 SFI 30 PIR 10DPI or MSensor G3 SFI 30 PIR 16DPI DALI sensor from Tridonic.

This outputs an occupancy signal (instance 0), a measured brightness value (instance 1) and six push button signals (instances 2 to 7).

The push button instances can be used via an infrared remote control from the vendor.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at `bInitialize` and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (`bError = FALSE`). If the output `bError = TRUE`, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variables `bQueryOccupancy` and `bQueryBrightness`, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions             : DWORD := 0;
  bQueryOccupancy     : BOOL := FALSE;
```

```
bQueryBrightness      : BOOL := FALSE;
bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

 **Outputs**

```
VAR_OUTPUT
bError      : BOOL;
ipResultMessage : I_TcMessage;
bInitializing : BOOL;
bReadingOccupancy : BOOL;
bReadingBrightness : BOOL;
bOccupied   : BOOL;
nBrightnessLevel : UINT;
nPushButton1Event : WORD;
nPushButton2Event : WORD;
nPushButton3Event : WORD;
nPushButton4Event : WORD;
nPushButton5Event : WORD;
nPushButton6Event : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
nPushButtonNEvent	WORD	The output outputs the events of the push buttons via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[► 840\]](#) can be used to convert the state of a push button into a BOOL variable.



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnablePushButtonN	BOOL	Get, Set	FALSE	Push button input that can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	255	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nEventFilterPushButtonInternN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ]
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press. This property does not apply to the instances <i>PushButtonScene1</i> and <i>PushButtonScene2</i> .



Name	Type	Access	Initial value	Description
nShortTimerPush ButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPush ButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.13.2 FB\_DALI\_Tridonic\_MSensorG3\_SRC\_30\_PIR**



The function block represents the MSensor G3 SRC 30 PIR 5DPI DALI sensor from Tridonic.

This outputs an occupancy signal (instance 0), a measured brightness value (instance 1) and six push button signals (instances 2 to 7).

The push button instances can be used via an infrared remote control from the vendor.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variables *bQueryOccupancy* and *bQueryBrightness*, even if the corresponding instance is not enabled.





Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryOccupancy      : BOOL  := FALSE;
  bQueryBrightness     : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingOccupancy     : BOOL;
  bReadingBrightness    : BOOL;
  bOccupied             : BOOL;
  nBrightnessLevel      : UINT;
  nPushButton1Event     : WORD;
  nPushButton2Event     : WORD;
  nPushButton3Event     : WORD;
  nPushButton4Event     : WORD;
  nPushButton5Event     : WORD;
  nPushButton6Event     : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

Name	Type	Description
nPushButtonNEvent	WORD	The output outputs the events of the push buttons via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[► 840\]](#) can be used to convert the state of a push button into a BOOL variable.



### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
bEnablePushButtonN	BOOL	Get, Set	FALSE	Push button input that can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled.
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	255	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nEventFilterPushButtonInternN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ]
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press. This property does not apply to the instances <i>PushButtonScene1</i> and <i>PushButtonScene2</i> .

Name	Type	Access	Initial value	Description
nShortTimerPush ButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPush ButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck. This property does not apply to the instances PushButtonScene1 and PushButtonScene2.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.13.3 FB\_DALI\_Tridonic\_XC\_G3\_CWM30**

```

FB_DALI_Tridonic_XC_G3_CWM_30
-----
bInitialize BOOL
nShortAddress BYTE
nOptions DWORD
bQueryInputValue1 BOOL
bQueryInputValue2 BOOL
bQueryInputValue3 BOOL
bQueryInputValue4 BOOL
-----
Tc3_EventLogger.I_TcMessage ipResultMessage
BOOL bError
BOOL bInitializing
BOOL bReadingInputValue1
BOOL bReadingInputValue2
BOOL bReadingInputValue3
BOOL bReadingInputValue4
WORD nPushButton1Event
WORD nPushButton2Event
WORD nPushButton3Event
WORD nPushButton4Event
    
```

The function block represents the DALI XC G3 CWM 30 DA2 DALI push button interface from Tridonic.

Up to four push buttons (instances 0 to 3) can be connected directly via the device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.

---

**i** Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.

---



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
  nOptions              : DWORD := 0;
  bQueryInputValue1    : BOOL := FALSE;
  bQueryInputValue2    : BOOL := FALSE;
  bQueryInputValue3    : BOOL := FALSE;
  bQueryInputValue4    : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

**Outputs**

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingInputValue1   : BOOL;
  bReadingInputValue2   : BOOL;
  bReadingInputValue3   : BOOL;
  bReadingInputValue4   : BOOL;
  nPushButton1Event     : WORD;
  nPushButton2Event     : WORD;
  nPushButton3Event     : WORD;
  nPushButton4Event     : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValueN	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block `FB_DALI_ToPushButtonState` [► 840] can be used to convert the state of a push button into a BOOL variable.



### Properties

All parameters that are written to the DALI control device via `bInitialize` are available as properties [► 890].

The properties of instances of the same type are listed only once in the table and marked with N at the end.

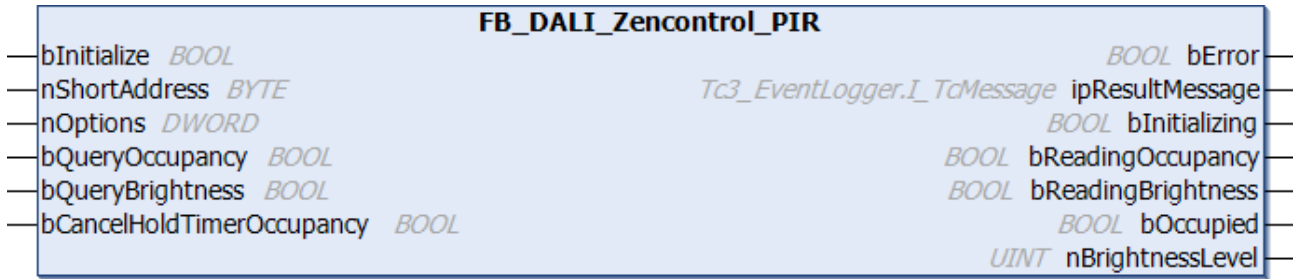
Name	Type	Access	Initial value	Description
<code>bEnablePushButtonN</code>	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <code>bQueryPushButtonN</code> .
<code>nEventFilterPushButtonN</code>	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
<code>eEventPriorityPushButtonN</code>	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
<code>nDoubleTimerPushButtonN</code>	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
<code>nRepeatTimerPushButtonN</code>	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
<code>nShortTimerPushButtonN</code>	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
<code>nStuckTimerPushButtonN</code>	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
<code>ipDALICommunication</code>	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

4.1.4.14 Zencontrol

4.1.4.14.1 FB\_DALI\_Zencontrol\_PIR



The function block represents the PIR DALI-2 sensor from Zencontrol.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

Part 304 (light sensors) – [Introduction \[► 561\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at `bInitialize` and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (`bError = FALSE`). If the output `bError = TRUE`, the outputs with the output values must not be evaluated.

The reading can be triggered immediately via the input variables `bQueryOccupancy` and `bQueryBrightness`, even if the corresponding instance is not enabled.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress       : BYTE  := 0;
  nOptions            : DWORD := 0;
  bQueryOccupancy     : BOOL  := FALSE;
  bQueryBrightness    : BOOL  := FALSE;
  bCancelHoldTimerOccupancy : BOOL := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryOccupancy	BOOL	The occupancy is queried immediately by a positive edge at this input.
bQueryBrightness	BOOL	The brightness is queried immediately by a positive edge at this input.
bCancelHoldTimerOccupancy	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .

## Outputs

```

VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied       : BOOL;
  nBrightnessLevel : UINT;
END_VAR

```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingOccupancy	BOOL	The output is TRUE as long as values of the instance of the movement sensor are being read by the DALI control device.
bReadingBrightness	BOOL	The output is TRUE as long as values of the instance of the light sensor are being read by the DALI control device.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

## Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

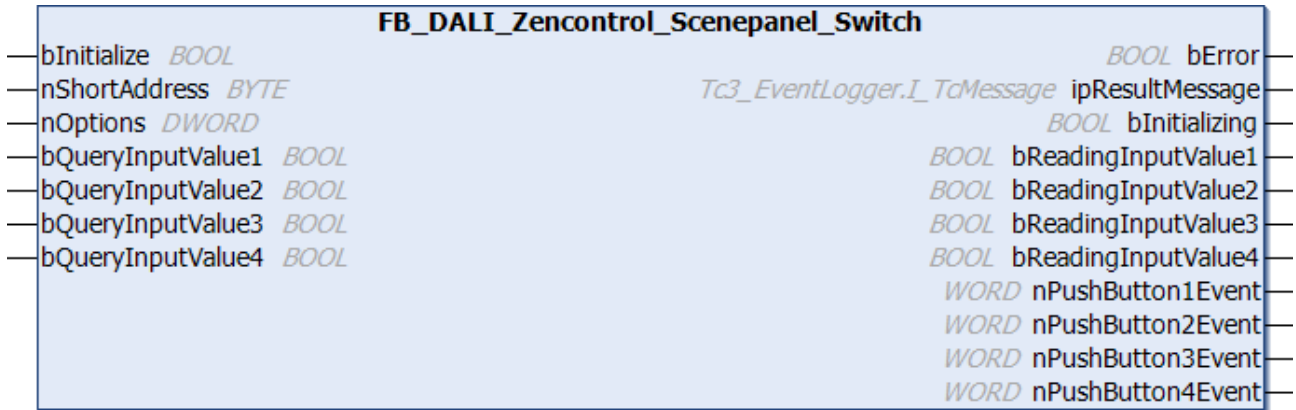


Name	Type	Access	Initial value	Description
bEnableBrightness	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for brightness measurement. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryBrightness</i> .
bEnableOccupancy	BOOL	Get, Set	TRUE	TRUE at this property enables the instance for motion detection. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryOccupancy</i> .
eEventPriorityBrightness	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
nReportTimerBrightness	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
nDeadtimeTimerBrightness	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHysteresisBrightness	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the brightness.
nHysteresisMinBrightness	BYTE	Get, Set	10	As the percentage hysteresis can also lead to large fluctuations at low brightness, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255.
eEventPriorityOccupancy	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimerOccupancy	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time after an event has been sent.
nHoldTimerOccupancy	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimerOccupancy	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.22	Tc3_DALI from v3.9.0.0

#### 4.1.4.14.2 FB\_DALI\_Zencontrol\_Scenepanel\_Switch



The function block represents the Scenepanel Switch DALI push button interface from Zencontrol.

Up to four push buttons (instances 0 to 3) can be connected directly via the device.

For information on the exact functions of the individual instances, please refer to the manufacturer's device documentation.

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction \[► 511\]](#)

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValueN*, even if the corresponding instance is not enabled.

The properties *bEnablePushButtonN* make it possible to disable individual instances.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

#### Inputs

```

VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions             : DWORD := 0;
  bQueryInputValue1   : BOOL  := FALSE;
  bQueryInputValue2   : BOOL  := FALSE;
  bQueryInputValue3   : BOOL  := FALSE;
  bQueryInputValue4   : BOOL  := FALSE;
END_VAR

```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValueN	BOOL	The state of the corresponding push button is queried immediately by a positive edge at this input.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingInputValue1 : BOOL;
  bReadingInputValue2 : BOOL;
  bReadingInputValue3 : BOOL;
  bReadingInputValue4 : BOOL;
  nPushButton1Event : WORD;
  nPushButton2Event : WORD;
  nPushButton3Event : WORD;
  nPushButton4Event : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValueN	BOOL	The outputs are set as soon as the manual reading of the corresponding input has been triggered.
nPushButtonNEvent	WORD	The outputs output the events for the corresponding push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState \[▶ 840\]](#) can be used to convert the state of a push button into a BOOL variable.

 **Properties**

All parameters that are written to the DALI control device via *bInitialize* are available as properties [[▶ 890](#)].

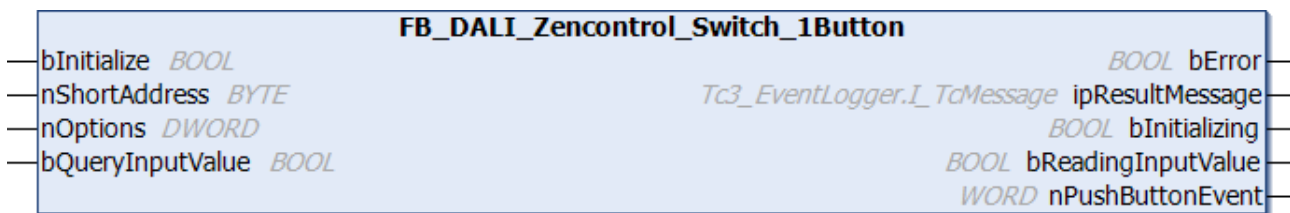
The properties of instances of the same type are listed only once in the table and marked with N at the end.

Name	Type	Access	Initial value	Description
bEnablePushButtonN	BOOL	Get, Set	TRUE	FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <i>bQueryPushButtonN</i> .
nEventFilterPushButtonN	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [ <a href="#">▶ 512</a> ].
eEventPriorityPushButtonN	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [ <a href="#">▶ 233</a> ].
nDoubleTimerPushButtonN	UINT	Get, Set	0 ms	Time (0, <i>tDoubleMin</i> ...2000 ms) to distinguish between a single and a double button press.
nRepeatTimerPushButtonN	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
nShortTimerPushButtonN	UINT	Get, Set	500 ms	Time ( <i>tShortMin</i> ...5100 ms) to distinguish between a long and a short button press. <i>tShortMin</i> is a manufacturer-specific value.
nStuckTimerPushButtonN	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.22	Tc3_DALI from v3.9.0.0

**4.1.4.14.3 FB\_DALI\_Zencontrol\_Switch\_1Button**



The function block represents the Switch 1 Button DALI push button interface from Zencontrol.

One push button (instance 0) can be connected directly via the device.

For information on the exact functions of the instance, please refer to the vendor's device documentation.

Further information on the supported instance type can be found here:

Part 301 (push buttons) - [Introduction](#) [[▶ 511](#)]

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.

The reading of the corresponding instance can be triggered immediately via the input variable *bQueryInputValue*, even if the corresponding instance is not enabled.

It is possible to disable the instance via the property *bEnablePushButton*.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nOptions              : DWORD := 0;
  bQueryInputValue     : BOOL  := FALSE;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nOptions	DWORD	Reserved for future extensions.
bQueryInputValue	BOOL	The state of the push button is queried immediately by a positive edge at this input.

Outputs

```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bInitializing         : BOOL;
  bReadingInputValue    : BOOL;
  nPushButtonEvent     : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue	BOOL	The output is set as soon as the manual reading of the input has been triggered.
nPushButtonEvent	WORD	The output outputs the events for the push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block `FB_DALI_ToPushButtonState` [▶ 840] can be used to convert the state of a push button into a BOOL variable.

 **Properties**

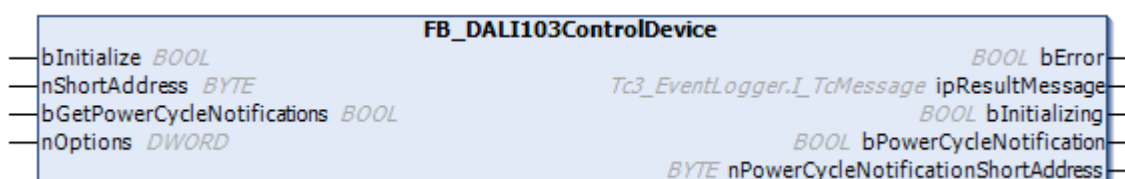
All parameters that are written to the DALI control device via `bInitialize` are available as `properties` [▶ 890].

Name	Type	Access	Initial value	Description
<code>bEnablePushButton</code>	BOOL	Get, Set	TRUE	Push button input that can be enabled or disabled on the DALI control device. FALSE on this property causes sending of events from this instance to be disabled. However, the actual value of the instance can still be queried via the input <code>bQueryPushButton</code> .
<code>nEventFilterPushButton</code>	BYTE	Get, Set	2#1000_0011	Setting of the <code>event filter</code> [▶ 512].
<code>eEventPriorityPushButton</code>	E_DALIEventPriority	Get, Set	Middle	Setting of the <code>event priority</code> [▶ 233].
<code>nDoubleTimerPushButton</code>	UINT	Get, Set	0 ms	Time (0, <code>tDoubleMin</code> ...2000 ms) to distinguish between a single and a double button press.
<code>nRepeatTimerPushButton</code>	UINT	Get, Set	160 ms	Interval (100...2000 ms) of repetitive events in the case of a long button press.
<code>nShortTimerPushButton</code>	UINT	Get, Set	500 ms	Time ( <code>tShortMin</code> ...5100 ms) to distinguish between a long and a short button press. <code>tShortMin</code> is a manufacturer-specific value.
<code>nStuckTimerPushButton</code>	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
<code>ipDALI_Communication</code>	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <code>Transfer of the reference to the communication block</code> [▶ 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.22	Tc3_DALI from v3.9.0.0

**4.1.4.15 FB\_DALI103ControlDevice**



The function block is used for the configuration and operation of a DALI control device that supports Part 103 (control devices).

The application controller can be activated, the operation mode specified and the event *Power Cycle Notification* enabled.

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
    bInitialize           : BOOL := FALSE;
    nShortAddress        : BYTE  := 0;
    bGetPowerCycleNotifs : BOOL := TRUE;
    nOptions              : DWORD := 0;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
bGetPowerCycleNotifs	BOOL	A TRUE at this input causes the received <i>Power Cycle Notification</i> to be output at the output.
nOptions	DWORD	Reserved for future extensions.

**Outputs**

```
VAR_OUTPUT
    bError                : BOOL;
    ipResultMessage       : I_TcMessage;
    bInitializing         : BOOL;
    bPowerCycleNotification : BOOL;
    nPowerCycleNotificationShortAddress : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI device has been started, and remains active until all DALI commands have been executed.
bPowerCycleNotification	BOOL	As soon as a <i>Power Cycle Notification</i> Event matches the desired filter criteria, this output is set to TRUE for one PLC cycle.
nPowerCycleNotificationShortAddress	BYTE	This output contains the short address (0..63) of the DALI control device that sent the Power Cycle Notification Event.



 **Properties**

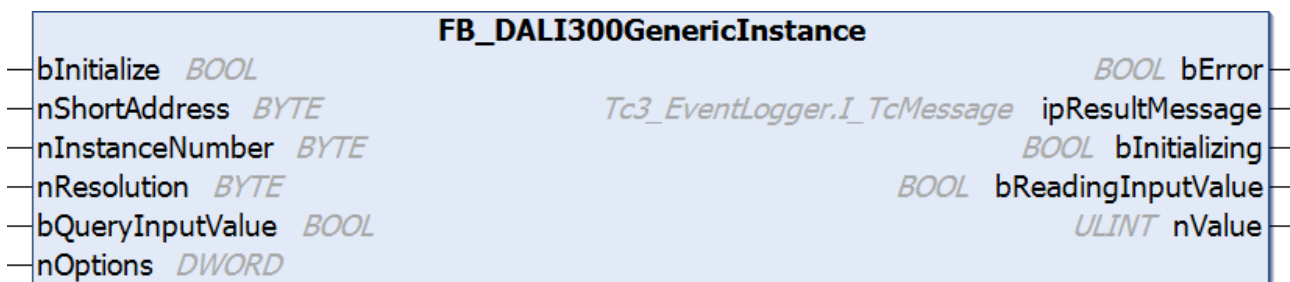
All parameters that are written to the DALI control device via *bInitialize* are available as properties [[▶ 890](#)].

Name	Type	Access	Initial value	Description
bEnableApplicationController	BOOL	Get, Set	FALSE	If this property is TRUE, the application controller (see <a href="#">applicationActive</a> [ <a href="#">▶ 230</a> ]) is enabled.
bEnablePowerCycleNotification	BOOL	Get, Set	FALSE	TRUE at this property enables the <i>Power Cycle Notification</i> event.
nOperatingMode	BYTE	Get, Set	0	Specification of the operation mode (0...255) of the DALI control device (see <a href="#">operatingMode</a> [ <a href="#">▶ 230</a> ]).
ipDALICommunication	I_DALI Communication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [ <a href="#">▶ 892</a> ]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.16 FB\_DALI300GenericInstance**



The function block is used for the direct reading of generic values of an instance of a DALI control device that supports Part 103 (instance type 0).

The input variable *bEnableInstance* can be used to configure whether the instance is enabled.

If necessary, the output value can also be queried immediately via the variable *bQueryInputValue*, even if the corresponding instance is not enabled.

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.



The function block cannot be used when using the KL6811.

 **Inputs**

```

VAR_INPUT
    bInitialize      : BOOL := FALSE;
    nShortAddress    : BYTE := 0;
    nInstanceNumber  : BYTE := 0;
    nResolution      : BYTE := 10;
    bQueryInputValue : BOOL := FALSE;
    nOptions         : DWORD := 0;
END_VAR
    
```



Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nInstanceNumber	BYTE	Number of the instance (0...31) of the DALI control device to be addressed.
nResolution	BYTE	Vendor-specific resolution (1...64) of the input value. The value specifies the number of bits with which the input value is scaled.
bQueryInputValue	BOOL	Immediate query of the measured value. A query is possible even if the instance is disabled via the <i>bEnableInstance</i> property.
nOptions	DWORD	Reserved for future extensions.

 **Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
  bReadingInputValue : BOOL;
  nValue           : ULINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">Error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue	BOOL	The output is set as soon as the manual reading of the measured value has been started and remains active until all DALI commands have been processed.
nValue	ULINT	Outputs the measured value.

 **Properties**

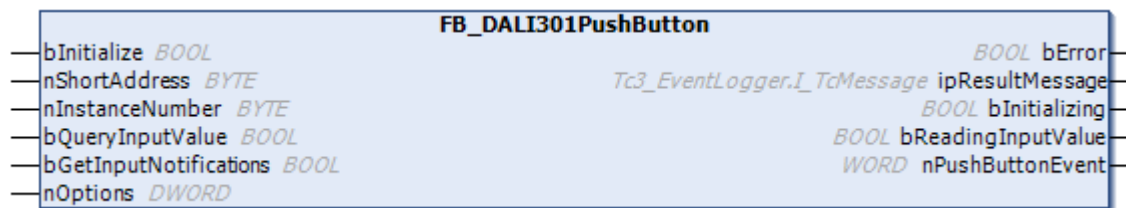
All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [▶ 890].

Name	Type	Access	Initial value	Description
bEnableInstance	BOOL	Get, Set	TRUE	The instance can be enabled or disabled on the DALI control device. If this property is FALSE, sending events from this instance is disabled. However, the actual value of the instance can still be queried via the input <i>bQueryInputValue</i> .
nEventFilter	DWORD	Get, Set	16#00_00_00_00	Setting of the event filter.
eEventPriority	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <u>event priority</u> [► 233].
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.17 FB\_DALI301PushButton**



The function block is used for the configuration and operation of an instance of a DALI control device that supports Part 301 (push buttons).

Further information on the supported instance types can be found here:

Part 301 (push buttons) - [Introduction](#) [► 511]

If necessary, the output value can also be queried immediately via the variable *bQueryInputValue*, even if the corresponding instance is not enabled.

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize          : BOOL := FALSE;
  nShortAddress        : BYTE := 0;
```

```
nInstanceNumber      : BYTE   := 0;
bQueryInputValue     : BOOL   := FALSE;
bGetInputNotifications : BOOL   := TRUE;
nOptions              : DWORD  := 0;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nInstanceNumber	BYTE	Number of the instance (0...31) of the DALI control device to be addressed.
bQueryInputValue	BOOL	The state of the input is queried immediately by a positive edge at this input. A query is possible even if the instance is disabled via the <i>bEnableInstance</i> property.
bGetInputNotifications	BOOL	A TRUE at this input causes the received <i>Input Notification Events</i> to be output at the output.
nOptions	DWORD	Reserved for future extensions.

 **Outputs**

```
VAR_OUTPUT
  bError          : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing   : BOOL;
  bReadingInputValue : BOOL;
  nPushButtonEvent : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue	BOOL	The output is set as soon as the manual reading of the input has been triggered.
nPushButtonEvent	WORD	The output outputs the events for the push button via one bit for one cycle.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block `FB_DALI_ToPushButtonState` [► 840] can be used to convert the state of a push button into a BOOL variable.

 **Properties**

All parameters that are written to the DALI control device via `bInitialize` are available as properties [► 890].

Name	Type	Access	Initial value	Description
bEnableInstance	BOOL	Get, Set	TRUE	The instance can be enabled or disabled on the DALI control device. If this property is FALSE, sending events from this instance is disabled. However, the actual value of the instance can still be queried via the input <code>bQueryInputValue</code> .
nEventFilter	BYTE	Get, Set	2#1000_0011	Setting of the <u>event filter</u> [► 512].
eEventPriority	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nRepeatTime	UINT	Get, Set	160 ms	Time (100...2000 ms) after which the pending push button event is repeated in case of a long button press.
nShortTimer	UINT	Get, Set	500 ms	Time ( <code>tShortMin</code> ...5100 ms) to distinguish between a long and a short button press.
nDoubleTimer	UINT	Get, Set	0 ms	Time (0, <code>tDoubleMin</code> ...2000 ms) to distinguish between a single and a double button press.
nStuckTimer	BYTE	Get, Set	20 s	Time (5...255 s) after whose expiry the push button no longer represents a long button press. The push button is stuck.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.18 FB\_DALI302AbsoluteInputDevice**

FB_DALI302AbsoluteInputDevice	
— <code>bInitialize</code> <i>BOOL</i>	<i>BOOL</i> <code>bError</code>
— <code>nShortAddress</code> <i>BYTE</i>	<i>Tc3_EventLogger.I_TcMessage</i> <code>ipResultMessage</code>
— <code>nInstanceNumber</code> <i>BYTE</i>	<i>BOOL</i> <code>bInitializing</code>
— <code>nResolution</code> <i>BYTE</i>	<i>BOOL</i> <code>bReadingInputValue</code>
— <code>bQueryInputValue</code> <i>BOOL</i>	<i>LWORD</i> <code>nInputLevel</code>
— <code>bGetInputNotifications</code> <i>BOOL</i>	
— <code>nOptions</code> <i>DWORD</i>	

The function block is used for the configuration and operation of an instance of a DALI control device that supports Part 302 (absolute encoder).

If necessary, the output value can also be queried immediately via the variable `bQueryInputValue`, even if the corresponding instance is not enabled.

The output variable `nInputLevel` is written depending on the specified resolution `nResolution` (application and vendor-specific).

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at `bInitialize` and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nInstanceNumber      : BYTE  := 0;
  nResolution          : BYTE  := 1;
  bQueryInputValue     : BOOL  := FALSE;
  bGetInputNotifications : BOOL := TRUE;
  nOptions             : DWORD := 0;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nInstanceNumber	BYTE	Number of the instance (0...31) of the DALI control device to be addressed.
nResolution	BYTE	Vendor-specific resolution (1...64) of the input value. The value specifies the number of bits with which the input value is scaled.
bQueryInputValue	BOOL	The state of the input is queried immediately by a positive edge at this input. A query is possible even if the instance is disabled via the <i>bEnableInstance</i> property.
bGetInputNotifications	BOOL	A TRUE at this input causes the received <i>Input Notification Events</i> to be output at the output.
nOptions	DWORD	Reserved for future extensions.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
  bReadingInputValue : BOOL;
  nInputLevel     : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation</a> [► 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [► 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue	BOOL	The output is set as soon as the manual reading of the input has been triggered.
nInputLevel	BYTE	Outputs the measured value of the resistance measurement.

### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties](#) [► 890].

Name	Type	Access	Initial value	Description
bEnableInstance	BOOL	Get, Set	TRUE	The instance can be enabled or disabled on the DALI control device. If this property is FALSE, sending events from this instance is disabled. However, the actual value of the instance can still be queried via the input <i>bQueryInputValue</i> .
eEventPriority	E_DALIEventPriority	Get, Set	Middle	Setting of the <a href="#">event priority</a> [► 233].
nReportTimer	BYTE	Get, Set	0	Time (0...255 s) after which the pending event of the measurement is repeated if no other event has occurred in the meantime.
nDeadtimeTimer	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block</a> [► 892]).

### Examples of the setting of the resolution:

#### Simple switch (make contact)

```
nResolution := 1;
```

When the contact is open, the output has the following value:

```
nInputLevel := 0;
```

When closed, the output has the following value:

```
nInputLevel := 1;
```

#### Switch with two positions

```
nResolution := 2;
```

When contact 1 is closed, the output has the following value:

```
nInputLevel := 1;
```

When contact 2 is closed, the output has the following value:

```
nInputLevel := 2;
```

**Switch with two positions and a neutral center position:**

```
nResolution := 2;
```

When contact 1 is closed, the output has the following value:

```
nInputLevel := 1;
```

In the center position, the output has the following value:

```
nInputLevel := 0
```

When contact 2 is closed, the output has the following value:

```
nInputLevel := 2;
```

**Rotary switch:**

When using a rotary switch with latching positions, the resolution *nResolution* depends on the number of available positions.

The output of the value *nInputLevel* also corresponds to this.

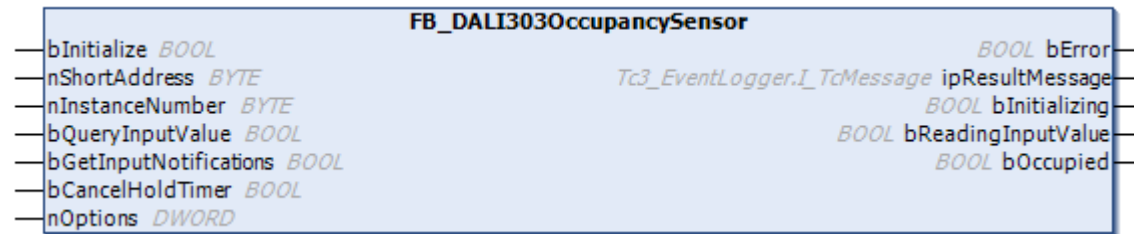
**Slide resistor, potentiometer (absolute encoder):**

When using an absolute encoder, the resolution *nResolution* and the output of the variable *nInputLevel* are vendor-dependent.

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.19 FB\_DALI303OccupancySensor**



The function block is used for the configuration and operation of an instance of a DALI control device that supports Part 303 (occupancy sensor).

Further information on the supported instance types can be found here:

Part 303 (occupancy sensors) – [Introduction \[► 542\]](#)

If necessary, the output value can also be queried immediately via the variable *bQueryInputValue*, even if the corresponding instance is not enabled.

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

### Inputs

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nInstanceNumber      : BYTE  := 0;
  bQueryInputValue     : BOOL  := FALSE;
  bGetInputNotifications : BOOL := TRUE;
  bCancelHoldTimer    : BOOL  := FALSE;
  nOptions              : DWORD := 0;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0..63) of the DALI control device.
nInstanceNumber	BYTE	Number of the instance (0...31) of the DALI control device to be addressed.
bQueryInputValue	BOOL	The state of the input is queried immediately by a positive edge at this input. A query is possible even if the instance is disabled via the <i>bEnableInstance</i> property.
bGetInputNotifications	BOOL	A TRUE at this input causes the received <i>Input Notification Events</i> to be output at the output.
bCancelHoldTimer	BOOL	A positive edge at this input prematurely ends the Hold Timer and resets the output <i>bOccupied</i> .
nOptions	DWORD	Reserved for future extensions.

### Outputs

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
  bReadingInputValue : BOOL;
  bOccupied        : BOOL;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">Error evaluation</a> [▶ 887]) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages</a> [▶ 873]). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue	BOOL	The output is set as soon as the manual reading of the measured value has been started and remains active until all DALI commands have been processed.
bOccupied	BOOL	If the output is TRUE, then occupancy has been detected in the detection range of the occupancy sensor.



 **Properties**

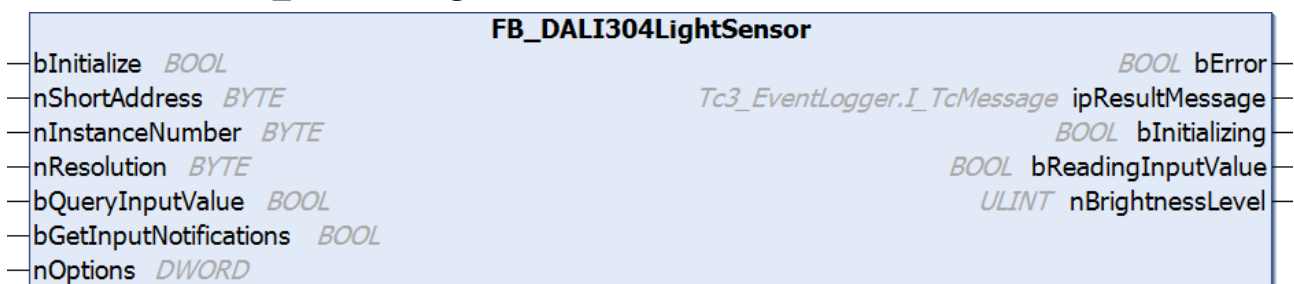
All parameters that are written to the DALI control device via *bInitialize* are available as properties [► 890].

Name	Type	Access	Initial value	Description
bEnableInstance	BOOL	Get, Set	TRUE	The instance can be enabled or disabled on the DALI control device. If this property is FALSE, sending events from this instance is disabled. However, the actual value of the instance can still be queried via the input <i>bQueryInputValue</i> .
nEventFilter	BYTE	Get, Set	2#0000_0111	Setting of the <u>event filter</u> [► 543].
eEventPriority	E_DALIEventPriority	Get, Set	Middle	Setting of the <u>event priority</u> [► 233].
nDeadtimeTimer	UINT	Get, Set	100 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHoldTimer	UINT	Get, Set	900 s	Hold time (1...2538 s) during which the occupancy is still active after no further motion was detected by the sensor.
nReportTimer	BYTE	Get, Set	60 s	Time (0...255 s) after which the pending event of the occupancy sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <u>Transfer of the reference to the communication block</u> [► 892]).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.20 FB\_DALI304LightSensor**



The function block is used for the configuration and operation of an instance of a DALI control device that supports Part 304 (light sensor).

Further information on the supported instance types can be found here:

Part 304 (light sensor) - Introduction [► 561]

If necessary, the output value can also be queried immediately via the variable *bQueryInputValue*, even if the corresponding instance is not enabled.

The output variable *nBrightnessLevel* is written depending on the specified resolution *nResolution* (application and vendor-specific).

The existing parameters can be overwritten by changing the properties listed below. All parameters are written to the DALI control device by a positive edge at *bInitialize* and stored there.

The outputs with the output values of the DALI device only contain valid values if the function block was executed without errors (*bError* = FALSE). If the output *bError* = TRUE, the outputs with the output values must not be evaluated.



Set the times so that as few events as possible are sent. If too many events are sent, this can have a negative effect on the behavior of the application.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bInitialize           : BOOL := FALSE;
  nShortAddress        : BYTE  := 0;
  nInstanceNumber     : BYTE  := 0;
  nResolution          : BYTE  := 10;
  bQueryInputValue    : BOOL  := FALSE;
  bGetInputNotificati : BOOL  := TRUE;
  nOptions             : DWORD := 0;
END_VAR
```

Name	Type	Description
bInitialize	BOOL	The configuration of the DALI control device is started by a positive edge at this input. During this time no DALI commands are processed.
nShortAddress	BYTE	Short address (0...63) of the DALI control device.
nInstanceNumber	BYTE	Number of the instance (0...31) of the DALI control device to be addressed.
nResolution	BYTE	Vendor-specific resolution (1...64) of the input value. The value specifies the number of bits with which the input value is scaled.
bQueryInputValue	BOOL	Immediate query of the measured value. A query is possible even if the instance is disabled via the <i>bEnableInstance</i> property.
bGetInputNotificati ns	BOOL	A TRUE at this input causes the received Input Notification Events to be output at the output.
nOptions	DWORD	Reserved for future extensions.

**Outputs**

```
VAR_OUTPUT
  bError           : BOOL;
  ipResultMessage : I_TcMessage;
  bInitializing    : BOOL;
  bReadingOccupancy : BOOL;
  bReadingBrightness : BOOL;
  bOccupied        : BOOL;
  nBrightnessLevel : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [► 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [► 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bInitializing	BOOL	The output is set as soon as the initialization of the DALI control device has been started, and remains active until all DALI commands have been executed.
bReadingInputValue	BOOL	The output is set as soon as the manual reading of the measured value has been started and remains active until all DALI commands have been processed.
nBrightnessLevel	UINT	Measured brightness value of the light sensor.



This measured value must be compared with the measured reference values at the place of operation of the DALI control device.

### Properties

All parameters that are written to the DALI control device via *bInitialize* are available as [properties \[► 890\]](#).

Name	Type	Access	Initial value	Description
bEnableInstance	BOOL	Get, Set	TRUE	The instance can be enabled or disabled on the DALI control device. If this property is FALSE, sending events from this instance is disabled. However, the actual value of the instance can still be queried via the input <i>bQueryInputValue</i> .
eEventPriority	E_DALIEventPriority	Get, Set	MiddleLow	Setting of the <a href="#">event priority [► 233]</a> .
nDeadtimeTimer	UINT	Get, Set	1500 ms	Dead time (0...12750 ms) to ensure that not too many events are sent in succession. An event is only sent after expiry of the dead time timer. The dead time timer is restarted each time an event is sent.
nHysteresis	BYTE	Get, Set	5 %	Value (0...25 %) for calculating a hysteresis value in order to prevent frequent and disturbing changes when measuring the illuminance.
nHysteresisMin	BYTE	Get, Set	10	Since the percentage hysteresis can lead to strong fluctuations when the illuminance is low, a minimum hysteresis can be specified via this property. The minimum hysteresis is an absolute value with a range from 0 to 255. The value depends on the resolution.
nReportTimer	BYTE	Get, Set	30 s	Time (0...255 s) after which the pending event of the light sensor is repeated if no other event has occurred in the meantime.
ipDALICommunication	I_DALICommunication	Get, Set	0	Interface pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [► 892]</a> ).

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.1.4.21 FB\_DALI\_ToPushButtonState**



Determines the static state from the events of a button.

For correct evaluation, the events *Button Released* (Bit 0) and *Button Pressed* (Bit 1) must be activated via the event filter.



Release only those events that you need for the application. If too many events are sent, this can have a negative effect on the behavior of the application.

**Inputs**

```
VAR_INPUT
  nPushButtonEvent      : UINT;
END_VAR
```

Name	Type	Description
nPushButtonEvent	UINT	This variable contains the individual events of the push button, which are represented by the respective bits.

Bit	Description
0	Push button released.
1	Push button pressed.
2	Short keystroke.
3	Double keystroke.
4	Start long keystroke.
5	Repeat long keystroke.
6	Stop long keystroke.
7	Push button free again; was previously blocked.
8	Push button blocked.

The function block [FB\\_DALI\\_ToPushButtonState](#) [[▶ 840](#)] can be used to convert the state of a push button into a BOOL variable.

**Outputs**

```
VAR_OUTPUT
  bPushButton          : BOOL;
END_VAR
```

Name	Type	Description
nPushButton	BOOL	This variable reflects the static state of the push button. The state was determined on the basis of the individual events.

**Example**

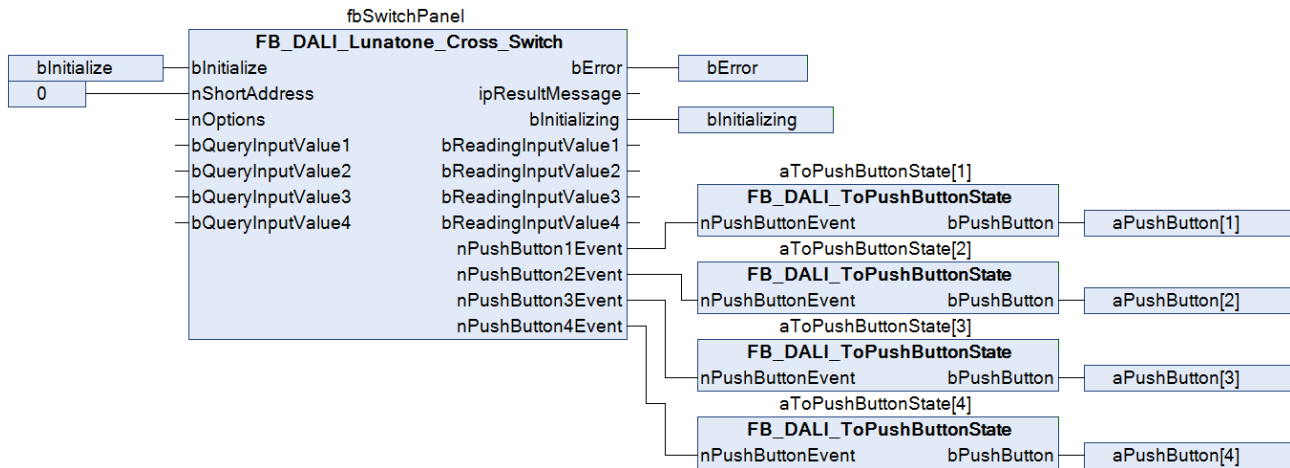
The example shows how the events of a 4-fold DALI push button are converted into four individual variables.

```
PROGRAM Demo_ST
VAR
  fbSwitchPanel      : FB_DALI_Lunatone_Cross_Switch(Communication.fbKL6821Communication);
  bInitialize        : BOOL;
  bError             : BOOL;
  bInitializing      : BOOL;
  aToPushButtonState : ARRAY [1..4] OF FB_DALI_ToPushButtonState();
  aPushButton        : ARRAY [1..4] OF BOOL;
END_VAR
```

Program part in structured text:

```
fbSwitchPanel(
  bInitialize := bInitialize,
  nShortAddress := 0,
  bInitializing => bInitializing,
  bError => bError);
aToPushButtonState[1](nPushButtonEvent := fbSwitchPanel.nPushButton1Event,
  bPushButton => aPushButton[1]);
aToPushButtonState[2](nPushButtonEvent := fbSwitchPanel.nPushButton2Event,
  bPushButton => aPushButton[2]);
aToPushButtonState[3](nPushButtonEvent := fbSwitchPanel.nPushButton3Event,
  bPushButton => aPushButton[3]);
aToPushButtonState[4](nPushButtonEvent := fbSwitchPanel.nPushButton4Event,
  bPushButton => aPushButton[4]);
```

Program part in the Continuous Function Chart (CFC):

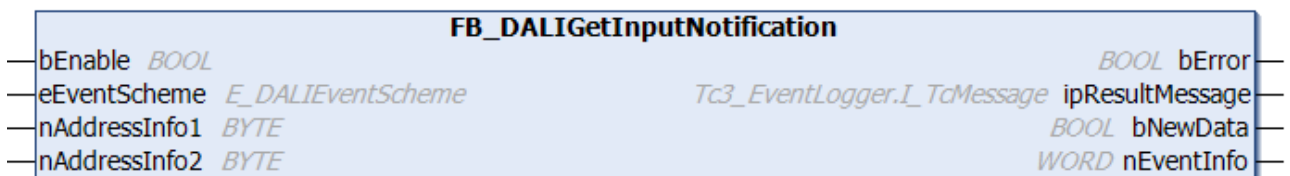


Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_DALI from v3.10.5.0

4.1.5 Events

4.1.5.1 FB\_DALIGetInputNotification



The function block filters all received *Input Notification Events* according to certain filter criteria.

Each *Input Notification* contains the address of the sender and further information about the event (output *nEventInfo*).

The address of the sender consists of two fields. The meaning of these fields is defined in the DALI control device by the event scheme (see [E\\_DALIEventScheme](#) [[▶ 859](#)]). The function block [FB\\_DALI103QueryEventScheme](#) [[▶ 191](#)] can be used to read the value of the event scheme; the function block [FB\\_DALI103SetEventScheme](#) [[▶ 174](#)] can be used to write to it. The two fields are a combination of the short address, instance number, instance type, instance group or device group.

The source of the *Input Notification* is defined by the inputs *eEventScheme*, *nAddressInfo1* and *nAddressInfo2*. As soon as an *Input Notification* is received from this source, the output *bNewData* is set and the event information is output on the output *nEventInfo*.

For each *Input Notification Event*, which is to be received and processed further, an instance of [FB\\_DALIGetInputNotification](#) must be created and configured with the correct filter criteria.

Moreover, this information can also be obtained at runtime with the help of the TwinCAT Communication Monitor extension.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
  bEnable      : BOOL;
  eEventScheme : E_DALIEventScheme := E_DALIEventScheme.DeviceInstance;
  nAddressInfo1 : BYTE;
  nAddressInfo2 : BYTE;
END_VAR
```

Name	Type	Description
bEnable	BOOL	If this input is TRUE, all received <i>Input Notification Events</i> of the DALI control devices are output according to the specified filter criteria. The filter criteria are defined by the inputs <i>eEventScheme</i> , <i>nAddressInfo1</i> and <i>nAddressInfo2</i> .
eEventScheme	<a href="#">E_DALIEventScheme</a> <a href="#">▶ 859</a>	This input specifies the event scheme that the expected event must have. The event scheme also determines the meaning of the two inputs <i>nAddressInfo1</i> and <i>nAddressInfo2</i> .
nAddressInfo1, nAddressInfo2	BYTE	see table below

eEventScheme	nAddressInfo1	nAddressInfo2
<a href="#">E_DALIEventScheme.Instance</a>	Instance type (0...31)	Instance number (0..31)
<a href="#">E_DALIEventScheme.Device</a>	Short address (0...63)	Instance type (0...31)
<a href="#">E_DALIEventScheme.DeviceInstance</a>	Short address (0...63)	Instance number (0...31)
<a href="#">E_DALIEventScheme.DeviceGroup</a>	Device group (0...31)	Instance type (0...31)
<a href="#">E_DALIEventScheme.InstanceGroup</a>	Instance group (0...31)	Instance type (0...31)

**Outputs**

```
VAR_OUTPUT
  bError      : BOOL;
  ipResultMessage : I_TcMessage;
  bNewData    : BOOL;
  nEventInfo  : WORD;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bNewData	BOOL	As soon as a <i>Input Notification Event</i> matches the desired filter criteria, this output is set to TRUE for one PLC cycle.
nEventInfo	WORD	If the output <i>bNewData</i> is TRUE, further information about the event can be found at this output. The exact meaning depends on the device type and is described in the respective Part 3xx of IEC 62386.

The possible values of the events for the respective instance types can be found here:

Part 301 (push buttons) - [Input Notification \[▶ 512\]](#)

Part 302 (absolute input devices) - [Input Notification \[▶ 533\]](#)

Part 303 (occupancy sensors) - [Input Notification \[▶ 544\]](#)

Part 304 (light sensors) - [Input Notification \[▶ 562\]](#)



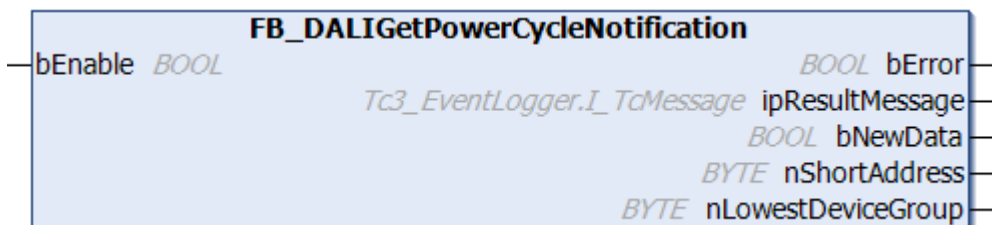
**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.1.5.2 FB\_DALIGetPowerCycleNotification**



The function block *B\_DALIGetPowerCycleNotification* filters out all received *Power Cycle Notification Events*.

Each *Power Cycle Notification Event* sent by a DALI control device contains two fields that provide information about the event source. These two fields are the short address and the smallest group address to which the DALI control device belongs.

An instance of FB\_DALIGetPowerCycleNotification must be created for each KL6821/EL6821 through which *Power Cycle Notification Events* are to be received and processed. If a *Power Cycle Notification Event* is detected, the output *bNewData* is set to TRUE for one PLC cycle. The two outputs *nShortAddress* and *nLowestDeviceGroup* provide information about the exact source of the event.



The function block cannot be used when using the KL6811.

**Inputs**

```
VAR_INPUT
    bEnable      : BOOL;
END_VAR
```

Name	Type	Description
bEnable	BOOL	If this input is TRUE, all received <i>Power Cycle Notification Events</i> of the DALI control devices are output.

**Outputs**

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
    bNewData        : BOOL;
    nShortAddress   : BYTE;
    nLowestDeviceGroup : BYTE;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bNewData	BOOL	As soon as a <i>Power Cycle Notification Event</i> matches the desired filter criteria, this output is set to TRUE for one PLC cycle.
nShortAddress	BYTE	This output contains the short address (0...63) of the DALI control device that sent the <i>Power Cycle Notification Event</i> . If the DALI control device does not have a short address, 255 (MASK) is output.
nLowestDeviceGroup	BYTE	This output contains the lowest group address (0...31) of the DALI control device that sent the <i>Power Cycle Notification Event</i> . If the DALI control device is not assigned to a group, 255 (MASK) is output.

**Properties**

Name	Type	Access	Initial value	Description
ipDALICommunication	I_DALICommunication	Get,Set	0	Interface Pointer to the communication block (see <a href="#">Transfer of the reference to the communication block [▶ 892]</a> ).

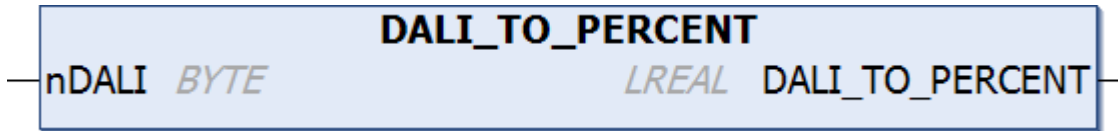


Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

4.1.6 Helper

4.1.6.1 DALI\_TO\_PERCENT



Conversion of the output value of a DALI control gear from 0...254 to 0...100 %.

The conversion is logarithmic, as described in IEC 62386. A special feature is the output value 255 (MASK). If the value 255 (MASK) is passed to the function, the value 255.0 is returned.

The following table shows some example values:

nDALI	Return value
0	0.0 %
1	0.100 %
10	0.128 %
85	0.991 %
100	1.492 %
128	3.206 %
150	5.845 %
200	22.892 %
254	100 %
255 (MASK)	255.0

 Inputs

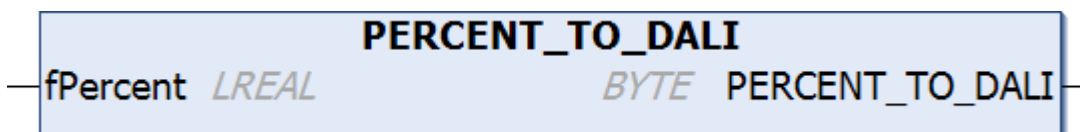
```
VAR_INPUT
  nDALI      : BYTE;
END_VAR
```

Name	Type	Description
nDALI	BYTE	Output value for the DALI control gear (0...254).

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.32	Tc3_DALI from v3.12.0.0

4.1.6.2 PERCENT\_TO\_DALI



Conversion of the output value of a DALI control gear from 0...100 % to 0...254.

The conversion is logarithmic, as described in IEC 62386. A special feature is the output value 255 (MASK). If the value 255.0 is passed to the function, the value 255 (MASK) is returned.

The following table shows some example values:

fPercent	Return value
<= 0.0	0
0.100	1
0.128	10
0.991	85
1.492	100
3.206	128
5.845	150
22.892	200
100.0 <= fPercent < 255.0	254
>= 255.0	255 (MASK)

### Inputs

```
VAR_INPUT
    fPercent          : LREAL;
END_VAR
```

Name	Type	Description
fPercent	LREAL	Output value for the DALI control gear (0...100%).

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.32	Tc3_DALI from v3.12.0.0

## 4.1.7 Simulation

Lighting technology applications have become increasingly extensive and complex in recent years. On the one hand, the demands for more flexibility have increased and, on the other hand, additional performance features such as Tunable White (color temperature control), dynamic scene management or diagnostic functions have been added. At the same time, the time and effort required to create the application program and commission the system should be kept to a minimum.

By simulating the DALI devices used, an application program can be completely created, extensively tested and optimized even before commissioning. Deficits in the design of the hardware are detected at an early stage and cost-intensive conversion measures are prevented. The effort for commissioning at the real plant is reduced.

In addition to the preparation of commissioning, the simulation blocks are also well suited for training or presentations. Thus, even larger DALI lines can be mapped without the need for additional hardware. The simulation blocks are also a prerequisite for automated testing of function blocks (unit tests).

The Tc3\_DALI library provides function blocks for simulating DALI lines with the associated DALI control gears. The simulation is completely transparent for the application program.

Simulated DALI lines are mapped by the function block [FB\\_DALIVirtualCommunication](#) [▶ 847]. As with [FB\\_KL6811Communication](#) [▶ 583], [FB\\_KL6821Communication](#) [▶ 586] and [FB\\_EL6821Communication](#) [▶ 578], [FB\\_DALIVirtualCommunication](#) also creates one instance per DALI line and calls it cyclically in a fast task.

```
fbDALISimulationCommunication : FB_DALIVirtualCommunication;
```

This instance is passed to the respective application function blocks.

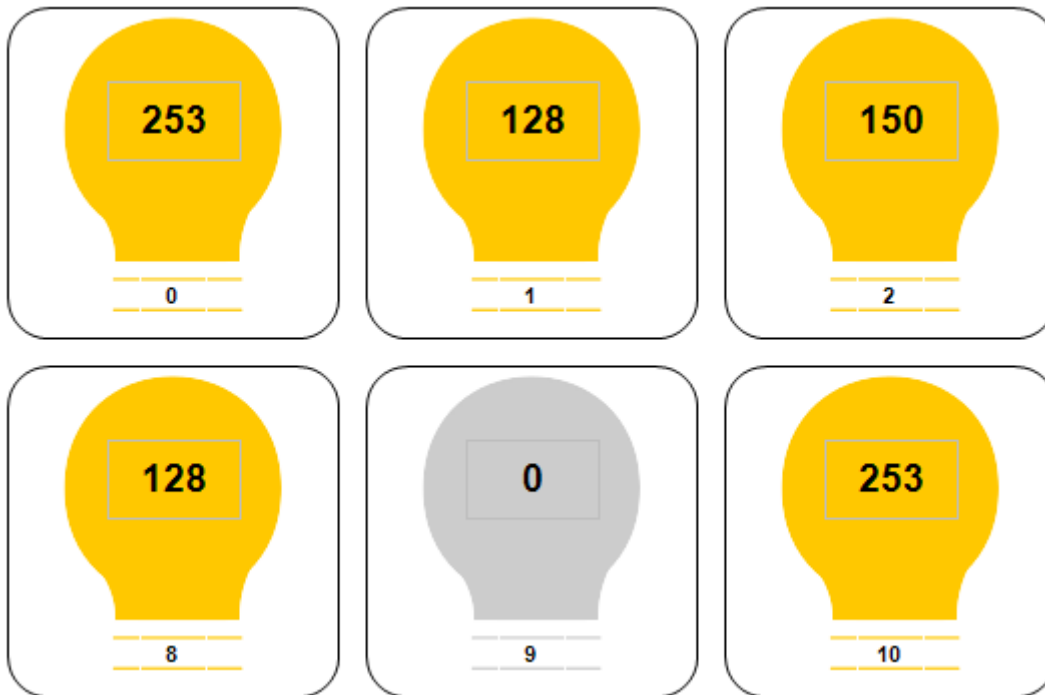
```
fbDimmer2Switch : FB_DALI102Dimmer2Switch(Communication.fbDALISimulatedCommunication);
```

The individual DALI control gears are represented by the function block [FB\\_DALIVirtualControlGear](#) [▶ 849]. One instance is required per DALI control gear. The individual parameters such as short address, group assignments, scenes, etc. are set via Properties. When the instances are declared, they are also assigned to the simulated DALI line.

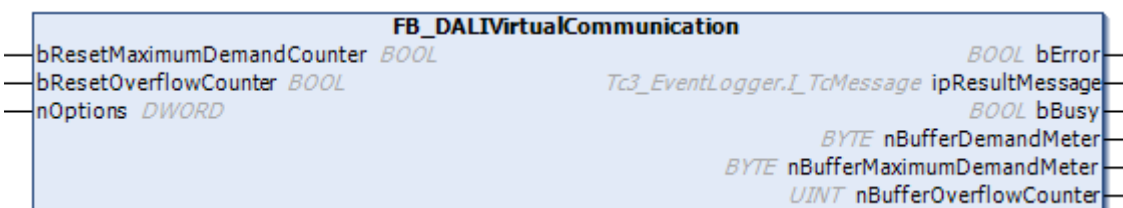
```
fbSUT : ARRAY [1..6] OF FB_DALIVirtualControlGear(fbVirtualCommunication) :=
  [(nShortAddress := 0, nGroups := 2#0010_0000_0100_1100, nMaxLevel := 254, nMinLevel := 85),
  (nShortAddress := 1, nGroups := 2#0000_0100_0000_0000, nMaxLevel := 254, nMinLevel := 85),
  (nShortAddress := 2, nGroups := 2#0000_0100_0000_0000, nMaxLevel := 254, nMinLevel := 85),
  (nShortAddress := 3, nGroups := 2#0000_0000_0100_1111, nMaxLevel := 254, nMinLevel := 85),
  (nShortAddress := 4, nGroups := 2#0010_0000_0000_1010, nMaxLevel := 254, nMinLevel := 85),
  (nShortAddress := 5, nGroups := 2#0010_0100_0000_0100, nMaxLevel := 254, nMinLevel := 85)];
```

The properties can be changed at runtime of the simulation. Hereby e.g. error states can be simulated (see property [bLampFailure](#) [▶ 849]) and the behavior of the application can be checked for these errors.

For the visual representation of the individual simulated DALI control gears, the TwinCAT PLC HMI can be used, for example. For this purpose, the ready-to-use sample application DALI PLC Simulation Tool is provided.



#### 4.1.7.1 FB\_DALIVirtualCommunication



The individual DALI commands are temporarily stored in a command buffer within the Tc3\_DALI library before further processing. The FB\_DALIVirtualCommunication function block sequentially reads the DALI commands from this command buffer and forwards the DALI commands to the associated instances of FB\_DALIVirtualControlGear [▶ 849].

One instance of the FB\_DALIVirtualCommunication function block must be created for each DALI line that is to be simulated. This instance must be called in a separate, faster task. This faster communication task must have a higher priority than the task in which the function blocks for the individual DALI commands are called.

The utilization rate of the command buffer can be determined using the outputs of the function block. If you find that the command buffer is overflowing regularly, you should take the following steps:

- How heavily are the individual PLC tasks utilized? TwinCAT provides suitable analysis tools.
- Try reducing the cycle time of the task in which the FB\_DALIVirtualCommunication function block is called. The value should not be greater than 6 ms. The optimum value is 2 ms or less.

- Check the cycle time of the PLC task in which the function blocks for the individual DALI commands are called. This value should not be greater than 10 ms. The optimum value is 8 ms or less.
- If possible avoid polling (regular reading) of values. Only read values when they are actually required.
- Distribute the individual simulated DALI devices evenly over several DALI lines. Since several DALI lines are processed simultaneously in each PLC cycle, this increases the data throughput.

 **Inputs**

```
VAR_INPUT
  bResetMaximumDemandCounter      : BOOL;
  bResetOverflowCounter           : BOOL;
  nOptions                        : DWORD := 0;
END_VAR
```

Name	Type	Description
bResetMaximumDemandCounter	BOOL	A positive edge resets the stored value for the maximum utilization of the command buffer, <i>nBufferMaximumDemandMeter</i> (0...100%).
ResetOverflowCounter	BOOL	A positive edge resets the stored value for the number of overflows of the command buffer, <i>nBufferOverflowCounter</i> .
nOptions	DWORD	Reserved for future extensions.

 **Outputs**

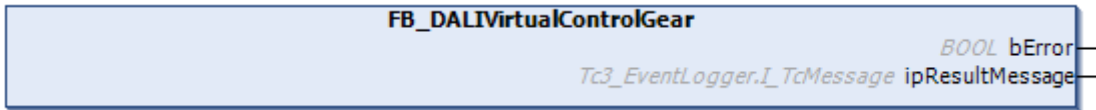
```
VAR_OUTPUT
  bError                : BOOL;
  ipResultMessage       : I_TcMessage;
  bBusy                 : BOOL;
  nBufferDemandMeter    : BYTE;
  nBufferMaximumDemandMeter : BYTE;
  nBufferOverflowCounter : UINT;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.
bBusy	BOOL	The output is set as soon as execution of the DALI commands has commenced. It remains active until all DALI commands have been processed.
nBufferDemandMeter	BYTE	Utilization rate of the command buffer (0...100%).
nBufferMaximumDemandMeter	BYTE	Maximum utilization rate of the command buffer reached so far (0...100%). The counter can be reset via the input <i>bResetMaximumDemandCounter</i> .
nBufferOverflowCounter	UINT	Number of command buffer overflows to date. The counter can be reset via the input <i>bResetOverflowCounter</i> .

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

### 4.1.7.2 FB\_DALIVirtualControlGear



The function block represents a DALI control gear on a virtual DALI line.

For each virtual DALI control gear an instance of FB\_DALIVirtualControlGear must be created. During the declaration the interface pointer of FB\_DALIVirtualCommunication [▶ 847] is passed, whereby the virtual DALI control gear is assigned to the virtual DALI line.

The individual variables for the virtual DALI control gear are set via Properties. The same function blocks as for real DALI control gears are used for access from the application program, e.g. FB\_DALI102GoToScene [▶ 87], FB\_DALI102SetMinLevel [▶ 73] or FB\_DALI102QueryContentDTR0 [▶ 100].

Currently FB\_DALIVirtualControlGear supports most DALI commands to Part 102.

The DALI PLC Simulation Tool is a TwinCAT 3 sample application that can be used to visualize virtual DALI control gears. The individual parameters of the virtual DALI control gears can be set via corresponding dialogs.

#### Outputs

```
VAR_OUTPUT
    bError          : BOOL;
    ipResultMessage : I_TcMessage;
END_VAR
```

Name	Type	Description
bError	BOOL	This output is switched to TRUE if an error occurs during the execution. Further information about the error can be queried via the variable <i>ipResultMessage</i> . The output is set to FALSE again as soon as <i>bBusy</i> switches to TRUE.
ipResultMessage	I_TcMessage	Interface pointer (see <a href="#">error evaluation [▶ 887]</a> ) that can be used to obtain detailed information about the processing of the function block (see <a href="#">runtime messages [▶ 873]</a> ). The interface pointer is valid after <i>bBusy</i> has changed from TRUE to FALSE.

#### Properties

Via Properties the [variables \[▶ 148\]](#) of the virtual DALI control gear can be set or also queried.

Name	Type	Access	Initial value	Description
aSceneLevels	REFERENCE TO ARRAY [0..15] OF BYTE	Get, Set	[16(255)]	See <a href="#">scene0...scene15</a> [► 154].
bControlGearFailure	BOOL	Get, Set	FALSE	See <a href="#">controlGearFailure</a> [► 153].
bLampFailure	BOOL	Get, Set	FALSE	See <a href="#">lampFailure</a> [► 153].
eExtendedFadeTimeBase	E_DALIExtendedFadeTimeBase	Get, Set	E_DALIExtendedFadeTimeBase.Base01	See <a href="#">extendedFadeTimeBase</a> [► 152].
eExtendedFadeTimeMultiplier	E_DALIExtendedFadeTimeMultiplier	Get, Set	E_DALIExtendedFadeTimeMultiplier.Disabled	See <a href="#">extendedFadeTimeMultiplier</a> [► 152].
eFadeRate	E_DALIFadeRate	Get, Set	E_DALIFadeRate.N045StepsPerSec	See <a href="#">fadeRate</a> [► 150].
eFadeTime	E_DALIFadeTime	Get, Set	E_DALIFadeTime.Disabled	See <a href="#">fadeTime</a> [► 151].
nActualLevel	BYTE	Get, Set	254	See <a href="#">actualLevel</a> [► 150].
nDTR0	BYTE	Get, Set	0	See <a href="#">DTR0...DTR2</a> [► 154].
nDTR1	BYTE	Get, Set	0	See <a href="#">DTR0...DTR2</a> [► 154].
nDTR2	BYTE	Get, Set	0	See <a href="#">DTR0...DTR2</a> [► 154].
nGroups	WORD	Get, Set	16#0000	See <a href="#">gearGroups</a> [► 154].
nMaxLevel	BYTE	Get, Set	254	See <a href="#">minLevel / maxLevel</a> [► 150].
nMinLevel	BYTE	Get, Set	85	See <a href="#">minLevel / maxLevel</a> [► 150].
nPhysicalMinimum	BYTE	Get, Set	85	See <a href="#">physicalMinLevel</a> [► 154].
nPowerOnLevel	BYTE	Get, Set	254	See <a href="#">powerOnLevel</a> [► 150]. The output value is set to <i>nPowerOnLevel</i> as soon as the function block is initialized (e.g. restart of the PLC program).
nShortAddress	BYTE	Get, Set	255	See <a href="#">shortAddress</a> [► 153].
nStatus	BYTE	Get, Set	0	Status of the DALI control gear. See also <a href="#">FB_DALI102QueryStatus</a> [► 126].
nSystemFailureLevel	BYTE	Get, Set	254	See <a href="#">systemFailureLevel</a> [► 150].

### Requirements

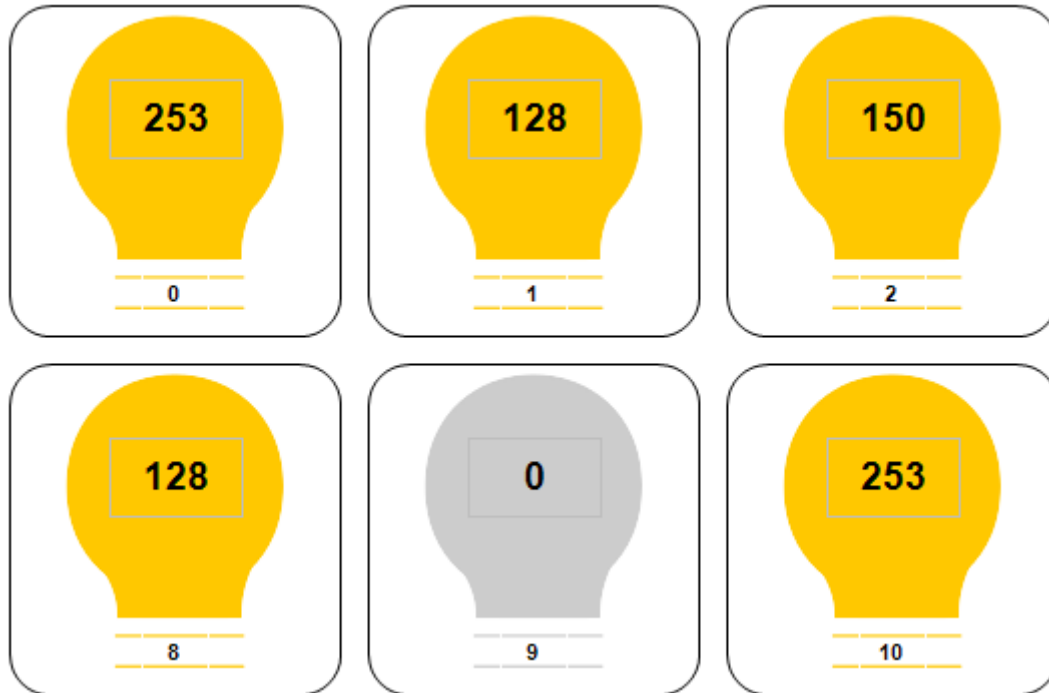
Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

#### 4.1.7.3 DALI PLC Simulation Tool

The DALI PLC Simulation Tool can be used to simulate DALI lines with up to 64 virtual DALI control gears (see [https://infosys.beckhoff.com/content/1033/TcPlcLib\\_Tc3\\_DALI/Resources/12051058443/.zip](https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc3_DALI/Resources/12051058443/.zip)).



The TwinCAT project is available for download as \*.zip file. This must first be unpacked locally so that the archive (\*.tnzip file) is available for import into the TwinCAT project.



Each virtual DALI control gear is represented by a lamp symbol. The number below the symbol is the DALI short address, while the number in the middle contains the current output value. If the output value is greater than 0, the color of the lamp symbol changes from gray to yellow.

If you click the Parameters button, an overview page with all parameters of the selected virtual DALI control gear opens.

### Operation of the parameter interface

The current parameters from the virtual DALI control gear are displayed here. It is also possible to modify them and thus change the behavior of the virtual DALI control gear.

### Parameters

<b>Short Address:</b>	2	<b>Max Level:</b> <small>(MinLevel...254)</small>	254	<b>DTR0:</b>	0
<b>Actual Level:</b> <small>(0, MinLevel...MaxLevel)</small>	150	<b>Min Level:</b> <small>(PhysicalMinimum...MaxLevel)</small>	85	<b>DTR1:</b>	0
<b>Power On Level:</b> <small>(0...255)</small>	0	<b>System Failure Level:</b> <small>(0...255)</small>	254	<b>DTR2:</b>	0
<b>Physical Minimum:</b> <small>(1...254)</small>	85				

### Groups

<input checked="" type="checkbox"/> 0	<input type="checkbox"/> 4	<input type="checkbox"/> 8	<input type="checkbox"/> 12
<input type="checkbox"/> 1	<input checked="" type="checkbox"/> 5	<input type="checkbox"/> 9	<input type="checkbox"/> 13
<input type="checkbox"/> 2	<input type="checkbox"/> 6	<input type="checkbox"/> 10	<input type="checkbox"/> 14
<input type="checkbox"/> 3	<input type="checkbox"/> 7	<input type="checkbox"/> 11	<input type="checkbox"/> 15

---

### Status

<input type="checkbox"/> Control Gear Failure	<input type="checkbox"/> Limit Error
<input type="checkbox"/> Lamp Failure	<input type="checkbox"/> Fade Running
<input checked="" type="checkbox"/> Lamp On	<input type="checkbox"/> Short Address is MASK

### Scene Level

0	255	4	255	8	255	12	255
1	255	5	150	9	255	13	255
2	255	6	255	10	255	14	255
3	100	7	255	11	255	15	255

---

### Fading

<b>Extended Fade Time Base:</b>	Base 01	▼
<b>Extended Fade Time Multiplier:</b>	Disabled	▼
<b>Fade Time:</b>	Disabled	▼
<b>Fade Rate:</b>	45 Steps/sec	▼

Back

For more information on the parameters used, see chapter [Variables](#) [▶ 148].

## Requirements

Required PLC library	DALI PLC Simulation Tool
Tc3_DALI from v3.11.0.0	Tc3_DALI_PLC_Simulation_Tool v1.0.0.0

## 4.2 DUTs

### 4.2.1 Structures

#### 4.2.1.1 ST\_DALICChangeAddressList

```

TYPE ST_DALICChangeAddressList :
STRUCT
  nOldAddress      : BYTE;
  nNewAddress      : BYTE;
  nRandomAddress   : UDINT;
  nErrors          : DWORD;
END_STRUCT
END_TYPE
    
```

## Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

#### 4.2.1.2 ST\_DALIVControlGearSettings

```

TYPE ST_DALIVControlGearSettings :
STRUCT
  nErrors          : DWORD;
  bPresent         : BOOL;
  nActualLevel     : BYTE;
END_STRUCT
    
```



```

nPowerOnLevel      : BYTE;
nSystemFailureLevel : BYTE;
nMinLevel          : BYTE;
nMaxLevel          : BYTE;
eFadeRate          : E_DALIFadeRate;
eFadeTime          : E_DALIFadeTime;
eFadeTimeBase      : E_DALIExtendedFadeTimeBase;
eFadeTimeMultiplier : E_DALIExtendedFadeTimeMultiplier;
nRandomAddress     : DWORD;
nGroups            : WORD;
aSceneLevels       : ARRAY [0..15] OF BYTE;
nStatus            : BYTE;
nMajorVersion      : BYTE;
nMinorVersion      : BYTE;
aDeviceTypes       : ARRAY [1..20] OF E_DALIDeviceType;
nPhysicalMinLevel  : BYTE;
END_STRUCT
END_TYPE

```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.2.1.3 ST\_EL6821InData**

```

TYPE ST_EL6821InData :
STRUCT
  nStatus           : WORD
  nSendErrorCode    : USINT;
  nNumberOfBufferEntries : USINT;
  nNumberOfAddressedDevices : USINT;
  nAddressingErrorCode : USINT;
  bInput01          : BIT;
  bInput02          : BIT;
  nRxBufferInfo     : WORD;
  nRxBufferFrame    : UDINT;
  stAdsAddr         : ST_AmsAddr;
END_STRUCT
END_TYPE

```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.47	Tc3_DALI from v3.16.1.0

**4.2.1.4 ST\_EL6821OutData**

```

TYPE ST_KL6821InData :
STRUCT
  nCtrl             : WORD;
  nAddressingStartAddress : USINT;
  nFrame            : UDINT
END_STRUCT
END_TYPE

```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.47	Tc3_DALI from v3.16.1.0

**4.2.1.5 ST\_KL6811InData**

```

TYPE ST_KL6811InData :
STRUCT
  nStatus           : BYTE;
  nData             : WORD;
END_STRUCT
END_TYPE

```

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.2.1.6 ST\_KL6811OutData**

```

TYPE ST_KL6811OutData :
STRUCT
  nCtrl          : BYTE;
  nData          : WORD;
END_STRUCT
END_TYPE

```

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.2.1.7 ST\_KL6821InData**

```

TYPE ST_KL6821InData :
STRUCT
  nStatus        : WORD
  aData          : ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE

```

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.2.1.8 ST\_KL6821OutData**

```

TYPE ST_KL6821OutData :
STRUCT
  nCtrl          : WORD;
  aData          : ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE

```

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.2.2 Enumerations****4.2.2.1 E\_DALIAddressType**

```

TYPE E_DALIAddressType :
(
  Short          := 0,
  Group          := 1,
  Broadcast      := 2,
  BroadcastUnaddr := 3
) BYTE := Short;
END_TYPE

```

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

### 4.2.2.2 E\_DALIColourTemperatureTcLimit

```

TYPE E_DALIColourTemperatureTcLimit :
(
  ColourTemperatureTcCoolest      := 0,
  ColourTemperatureTcWarmest     := 1,
  ColourTemperatureTcPhysicalCoolest := 2,
  ColourTemperatureTcPhysicalWarmest := 3
) USINT;
END_TYPE

```

#### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.2.2.3 E\_DALIColourType

```

TYPE E_DALIColourType :
(
  Unknown           := 16#00,
  XyCoordinate      := 16#10,
  ColourTemperatureTc := 16#20,
  PrimaryNDimLevel  := 16#40,
  RGBWAFControl     := 16#80,
  MASK              := 255
) USINT := MASK;
END_TYPE

```

#### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.2.2.4 E\_DALIColourValue

```

TYPE E_DALIColourValue :
(
  XCoordinate           := 0,
  YCoordinate           := 1,
  ColourTemperatureTc  := 2,
  PrimaryNDimLevel0    := 3,
  PrimaryNDimLevel1    := 4,
  PrimaryNDimLevel2    := 5,
  PrimaryNDimLevel3    := 6,
  PrimaryNDimLevel4    := 7,
  PrimaryNDimLevel5    := 8,
  RedDimLevel          := 9,
  GreenDimLevel        := 10,
  BlueDimLevel         := 11,
  WhiteDimLevel        := 12,
  AmberDimLevel        := 13,
  FreeColourDimLevel   := 14,
  RGBWAFControl        := 15,
  XCoordinatePrimaryN0 := 64,
  YCoordinatePrimaryN0 := 65,
  TYPrimaryN0          := 66,
  XCoordinatePrimaryN1 := 67,
  YCoordinatePrimaryN1 := 68,
  TYPrimaryN1          := 69,
  XCoordinatePrimaryN2 := 70,
  CCoordinatePrimaryN2 := 71,
  TYPrimaryN2          := 72,
  XCoordinatePrimaryN3 := 73,
  YCoordinatePrimaryN3 := 74,
  TYPrimaryN3          := 75,
  XCoordinatePrimaryN4 := 76,
  YCoordinatePrimaryN4 := 77,
  TYPrimaryN4          := 78,
  XCoordinatePrimaryN5 := 79,
  YCoordinatePrimaryN5 := 80,
  TYPrimaryN5          := 81,
  NumberOfPrimaries    := 82,
  ColourTemperatureTcCoolest := 128,

```

```

ColourTemperatureTcPhysicalCoolest := 129,
ColourTemperatureTcWarmest        := 130,
ColourTemperatureTcPhysicalWarmest := 131,
TemporaryXCoordinate              := 192,
TemporaryYCoordinate              := 193,
TemporaryColourTemperatureTc      := 194,
TemporaryPrimaryNDimLevel0        := 195,
TemporaryPrimaryNDimLevel1        := 196,
TemporaryPrimaryNDimLevel2        := 197,
TemporaryPrimaryNDimLevel3        := 198,
TemporaryPrimaryNDimLevel4        := 199,
TemporaryPrimaryNDimLevel5        := 200,
TemporaryRedDimLevel              := 201,
TemporaryGreenDimLevel            := 202,
TemporaryBlueDimLevel             := 203,
TemporaryWhiteDimLevel            := 204,
TemporaryAmberDimLevel            := 205,
TemporaryFreeColourDimLevel       := 206,
TemporaryRGBWAFControl            := 207,
TemporaryColourType               := 208,
ReportXCoordinate                  := 224,
ReportYCoordinate                  := 225,
ReportColourTemperatureTc         := 226,
ReportPrimaryNDimLevel0           := 227,
ReportPrimaryNDimLevel1           := 228,
ReportPrimaryNDimLevel2           := 229,
ReportPrimaryNDimLevel3           := 230,
ReportPrimaryNDimLevel4           := 231,
ReportPrimaryNDimLevel5           := 232,
ReportRedDimLevel                 := 233,
ReportGreenDimLevel               := 234,
ReportBlueDimLevel                := 235,
ReportWhiteDimLevel               := 236,
ReportAmberDimLevel               := 237,
ReportFreeColourDimLevel          := 238,
ReportRGBWAFControl               := 239,
ReportColourType                  := 240
) USINT;
END_TYPE

```

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

#### 4.2.2.5 E\_DALICommandPriority

```

TYPE E_DALICommandPriority :
(
  High           := 1,
  MiddleHigh     := 2,
  Middle         := 3,
  MiddleLow      := 4,
  Low            := 5,
  Unknown        := 255
) BYTE := Middle;
END_TYPE

```

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

#### 4.2.2.6 E\_DALIConfigurationCommand

```

TYPE E_DALIConfigurationCommand :
(
  DoNothing      := 0,
  Off            := 1,
  RecallMaxLevel := 2,
  RecallMinLevel := 3,
  GoToScene0     := 4,
  GoToScene1     := 5,
  GoToScene2     := 6,

```

```

GoToScene3      := 7,
GoToScene4      := 8,
GoToScene5      := 9,
GoToScene6      := 10,
GoToScene7      := 11,
GoToScene8      := 12,
GoToScene9      := 13,
GoToScene10     := 14,
GoToScene11     := 15,
GoToScene12     := 16,
GoToScene13     := 17,
GoToScene14     := 18,
GoToScene15     := 19
);
END_TYPE

```

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.2.2.7 E\_DALIDataFrameType**

```

TYPE E_DALIDataFrameType :
(
  Bit16          := 1,
  Bit24          := 3
) BYTE;
END_TYPE

```

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.2.2.8 E\_DALIDeviceType**

```

TYPE E_DALIDeviceType :
(
  DT00FluorescentLamp                := 0,
  DT01IndependentEmergencyLighting   := 1,
  DT02DischargeLamp                  := 2,
  DT03LowVoltageHalogenLamp          := 3,
  DT04IncandescentLamp               := 4,
  DT05ConversionOfDigitalSignalToDCVoltage := 5,
  DT06LEDModule                      := 6,
  DT07SwitchingFunction              := 7,
  DT08ColourControl                  := 8,
  DT09Sequencer                      := 9,
  DT15LoadReferencing                := 15,
  DT16ThermalGearProtection          := 16,
  DT17DimmingCurveSelection          := 17,
  DT19CentrallySuppliedEmergencyOperation := 19,
  DT20LoadShedding                  := 20,
  DT21ThermalLampProtection          := 21,
  DT23IntegratedLightSource          := 23,
  DT49IntegratedBusPowerSupply       := 49,
  DT50MemoryBank1Extension          := 50,
  DT51EnergyReporting                := 51,
  DT52DiagnosticsMaintenance        := 52,
  DT53ExtendedEmergencyData         := 53,
  Unknown                            := 255
) BYTE := Unknown;
END_TYPE

```

**DT00FluorescentLamp:** Part 201: Standard device (device type 0)

**DT01IndependentEmergencyLighting:** Part 202: Device for emergency lighting (device type 1)

**DT02DischargeLamp:** Part 203: Device for discharge lamps (device type 2)

**DT03LowVoltageHalogenLamp:** Part 204: Device for low-voltage halogen lamps (device type 3)

**DT04IncandescentLamp:** Part 205: Supply voltage controller for incandescent lamps (device type 4)

**DT05ConversionOfDigitalSignalToDCVoltage:** Part 206: Device for converting digital signals into DC voltage signals (device type 5)

**DT06LEDModule:** Part 207: Device for LED modules (device type 6)

**DT07SwitchingFunction:** Part 208: Device for switching functions (device type 7)

**DT08ColourControl:** Part 209: Device for color/color temperature control (device type 8)

**DT09Sequencer:** Part 210: Sequencer (device type 9)

**DT15LoadReferencing:** Part 216: Load referencing (device type 15)

**DT16ThermalGearProtection:** Part 217: Thermal gear protection (device type 16)

**DT17DimmingCurveSelection:** Part 218: Dimming curve selection (device type 17)

**DT19CentrallySuppliedEmergencyOperation:** Part 220: Centrally supplied emergency operation (device type 19)

**DT20LoadShedding:** Part 221: Load shedding (device type 20)

**DT21ThermalLampProtection:** Part 222: Thermal lamp protection (device type 21)

**DT23IntegratedLightSource:** Part 224: Non-replaceable light sources (device type 23)

**DT49IntegratedBusPowerSupply:** Part 250: Devices with integrated DALI bus power supply (device type 49)

**DT50MemoryBank1Extension:** Part 251: Further information and parameters for DALI control gears in memory bank 1 (device type 50)

**DT51EnergyReporting:** Part 252: Further parameters for the creation of an energy report (device type 51)

**DT52DiagnosticsMaintenance:** Part 253: Further parameters with diagnostic and maintenance information for DALI control gears (device type 52)

**DT53ExtendedEmergencyData:** Part 254: Extended information for DALI control gears for emergency lighting (device type 53)

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

### 4.2.2.9 E\_DALIDimmingCurve

```
TYPE E_DALIDimmingCurve :
(
  Standard      := 0,
  Linear        := 1,
  Unknown       := 255
) BYTE := Unknown;
END_TYPE
```

## Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

### 4.2.2.10 E\_DALIEventPriority

```
TYPE E_DALIEventPriority :
(
  MiddleHigh    := 2,
  Middle         := 3,
```

```

MiddleLow      := 4,
Low            := 5,
Unknown       := 255
) BYTE := Middle;
END_TYPE

```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.2.2.11 E\_DALIEventScheme**

```

TYPE E_DALIEventScheme :
(
  Instance           := 0,
  Device            := 1,
  DeviceInstance    := 2,
  DeviceGroup       := 3,
  InstanceGroup     := 4,
  Unknown           := 255
) BYTE := Unknown;
END_TYPE

```

**Instance:** Instance addressing with instance type and instance number.

**Device:** Device addressing with short address and instance type.

**DeviceInstance:** Device/instance addressing with short address and instance number.

**DeviceGroup:** Addressing of device groups with device group and instance type.

**InstanceGroup:** Addressing of instance groups with instance group and instance type.

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.2.2.12 E\_DALIEventType**

```

TYPE E_DALIEventType :
(
  InputNotification := 0,
  PowerNotification := 1,
  Unknown           := 255
) BYTE := Unknown;
END_TYPE

```

**Requirements**

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

**4.2.2.13 E\_DALIExtendedFadeTimeBase**

```

TYPE E_DALIExtendedFadeTimeBase :
(
  Base01 := 0,
  Base02 := 1,
  Base03 := 2,
  Base04 := 3,
  Base05 := 4,
  Base06 := 5,
  Base07 := 6,
  Base08 := 7,
  Base09 := 8,
  Base10 := 9,
  Base11 := 10,
  Base12 := 11,

```

```

Base13      := 12,
Base14      := 13,
Base15      := 14,
Base16      := 15,
Unknown     := 255
) BYTE := Unknown;
END_TYPE

```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

**4.2.2.14 E\_DALIExtendedFadeTimeMultiplier**

```

TYPE E_DALIExtendedFadeTimeMultiplier :
(
  Disabled      := 0,
  Multiplier100ms := 1,
  Multiplier1s  := 2,
  Multiplier10s := 3,
  Multiplier1min := 4,
  Unknown       := 255
) BYTE := Unknown;
END_TYPE

```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.2.2.15 E\_DALIFadeRate**

```

TYPE E_DALIFadeRate :
(
  N358StepsPerSec := 1,
  N253StepsPerSec := 2,
  N179StepsPerSec := 3,
  N127StepsPerSec := 4,
  N089StepsPerSec := 5,
  N063StepsPerSec := 6,
  N045StepsPerSec := 7,
  N032StepsPerSec := 8,
  N022StepsPerSec := 9,
  N016StepsPerSec := 10,
  N011StepsPerSec := 11,
  N008StepsPerSec := 12,
  N006StepsPerSec := 13,
  N004StepsPerSec := 14,
  N003StepsPerSec := 15,
  Unknown          := 255
) BYTE := Unknown;
END_TYPE

```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

**4.2.2.16 E\_DALIFadeTime**

```

TYPE E_DALIFadeTime :
(
  Disabled      := 0,
  T00707ms     := 1,
  T01000ms     := 2,
  T01400ms     := 3,
  T02000ms     := 4,
  T02800ms     := 5,
  T04000ms     := 6,
  T05700ms     := 7,

```



```
T08000ms      := 8,
T11300ms      := 9,
T16000ms      := 10,
T22600ms      := 11,
T32000ms      := 12,
T45300ms      := 13,
T64000ms      := 14,
T90500ms      := 15,
Unknown       := 255
) BYTE := Unknown;
END_TYPE
```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.4	Tc3_DALI from v3.2.0.0

**4.2.2.17 E\_DALIFastFadeTime**

```
TYPE E_DALIFastFadeTime :
(
  Disabled      := 0,
  T100ms        := 4,
  T200ms        := 8,
  T225ms        := 9,
  T300ms        := 12,
  T400ms        := 16,
  T500ms        := 20,
  T600ms        := 24,
  T700ms        := 27,
  Unknown       := 255
) BYTE := Unknown;
END_TYPE
```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

**4.2.2.18 E\_DALIInstanceAddressType**

```
TYPE E_DALIInstanceAddressType :
(
  InstanceNumber      := 0,
  InstanceGroup       := 1,
  InstanceType        := 2,
  FeatureNumber       := 3,
  FeatureGroup        := 4,
  FeatureType         := 5,
  FeatureBroadcast    := 6,
  InstanceBroadcast   := 7,
  FeatureDevice       := 8,
  Device              := 9
) BYTE := InstanceNumber;
END_TYPE
```

- InstanceNumber:** Instance number (0...31).
- InstanceGroup:** Instance group (0...31).
- InstanceType:** Instance type (0...31).
- FeatureNumber:** Feature at instance number level (0...31).
- FeatureGroup:** Feature at instance group level (0...31).
- FeatureType:** Feature at instance type level (0...31).
- FeatureBroadcast:** Feature at instance broadcast level.
- InstanceBroadcast:** Instance broadcast.

**FeatureDevice:** Feature at device level.

**Device:** Device

### Requirements

Development environment	required PLC library
TwinCAT from v3.1.4022.29	Tc3_DALI from v3.1.4.0

#### 4.2.2.19 E\_DALIInstanceType

```

TYPE E_DALIInstanceType :
(
  IT00GenericInstance           := 0,
  IT01PushButton                := 1,
  IT02AbsoluteInputDevice      := 2,
  IT03OccupancySensor          := 3,
  IT04LightSensor              := 4,
  IT05ColourSensor             := 5,
  IT06GeneralPurposeSensor     := 6,
  IT32InputControlDeviceFeedback := 32,
  IT33ManualConfiguration      := 33,
  IT51LuminaireMountedControlDevice := 51,
  Unknown                       := 255
) BYTE := Unknown;
END_TYPE

```

**IT00GenericInstance:** Part 103: Control device (instance type 0)

**IT01PushButton:** Part 301: Push button (instance type 1)

**IT02AbsoluteInputDevice:** Part 302: Absolute input device (instance type 2)

**IT03OccupancySensor** Part 303: Occupancy sensor (instance type 3)

**IT04LightSensor:** Part 304: Light sensor (instance type 4)

**IT05ColourSensor:** Part 305: Color sensor (instance type 5)

**IT06GeneralPurposeSensor:** Part 306: Universal sensor (instance type 6)

**IT32InputControlDeviceFeedback:** Part 332: Feedback for input devices (instance type 32)

**IT33ManualConfiguration:** Part 333: Manual configuration (instance type 33)

**IT51LuminaireMountedControlDevice:** Part 351: Control device integrated in the luminaire (instance type 51)

### Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.11	Tc3_DALI from v3.5.0.0

#### 4.2.2.20 E\_DALILightSourceType

```

TYPE E_DALILightSourceType :
(
  LowPressureFluorescent := 0,
  HID                    := 2,
  LowVoltageHalogen     := 3,
  Incandescent          := 4,
  LED                    := 6,
  OLED                  := 7,
  Other                  := 252,
  Unknown                := 253,
  NoLightSource         := 254,
  Multiple                := 255
) BYTE := Unknown;
END_TYPE

```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.29	Tc3_DALI from v3.11.0.0

**4.2.2.21 E\_DALILoytecInputConfiguration**

```
TYPE E_DALILoytecInputConfiguration :
(
  PushButton      := 0,
  GenericInput    := 1,
  AbsoluteValue   := 2
) BYTE := PushButton;
END_TYPE
```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.2.2.22 E\_DALILoytecResistanceConfiguration**

```
TYPE E_DALILoytecResistanceConfiguration :
(
  Resistance1kOhm := 0,
  Resistance10kOhm := 1,
) BYTE := Resistance10kOhm;
END_TYPE
```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.10	Tc3_DALI from v3.4.0.0

**4.2.2.23 E\_DALITestTiming**

```
TYPE E_DALITestTiming :
(
  FunctionTestDelayTime := 0,
  DurationTestDelayTime := 2,
  FunctionTestInterval := 4,
  DurationTestInterval := 5,
  TestExecutionTimeout := 6,
  ProlongTime := 7
) BYTE;
END_TYPE
```

**Requirements**

Development environment	Required PLC library
TwinCAT from v3.1.4024.35	Tc3_DALI from v3.13.0.0

**4.3 GVLs**

**4.3.1 Constants**

```
VAR_GLOBAL
  eEventTraceLevel      : TcEventSeverity := TcEventSeverity.Critical;
END_VAR
VAR_GLOBAL_CONSTANT
  cMemoryMode           : DWORD := 16#0000_0001;

  cCompleteNewInstallation : DWORD := 16#0000_0001;
  cDeleteAllGroupAssignments : DWORD := 16#0000_0002;
  cDeleteAllSceneAssignments : DWORD := 16#0000_0004;
  cOpticalFeedback       : DWORD := 16#0000_0008;
```

```

cCheckLampFailure           : DWORD := 16#0000_0010;
cCheckControlGearFailure   : DWORD := 16#0000_0020;
cCheckUnaddressedControlGears : DWORD := 16#0000_0040;

cOptionActualDimLevel      : DWORD := 16#0000_0001;
cOptionPowerOnLevel        : DWORD := 16#0000_0002;
cOptionSystemFailureLevel  : DWORD := 16#0000_0004;
cOptionMinLevel            : DWORD := 16#0000_0008;
cOptionMaxLevel            : DWORD := 16#0000_0010;
cOptionFadeRate            : DWORD := 16#0000_0020;
cOptionFadeTime            : DWORD := 16#0000_0040;
cOptionFadeRateFadeTime    : DWORD := 16#0000_0080;
cOptionExtendedFadeTime    : DWORD := 16#0000_0100;
cOptionRandomAddress        : DWORD := 16#0000_0200;
cOptionGroups              : DWORD := 16#0000_0400;
cOptionSceneLevels         : DWORD := 16#0000_0800;
cOptionStatusInformation    : DWORD := 16#0000_1000;
cOptionVersionNumber       : DWORD := 16#0000_2000;
cOptionDeviceType          : DWORD := 16#0000_4000;
cOptionPhysicalMinLevel    : DWORD := 16#0000_8000;
cOptionAll                  : DWORD := 16#00FF_FFFF;
END_VAR

```

## 4.3.2 Parameter

```

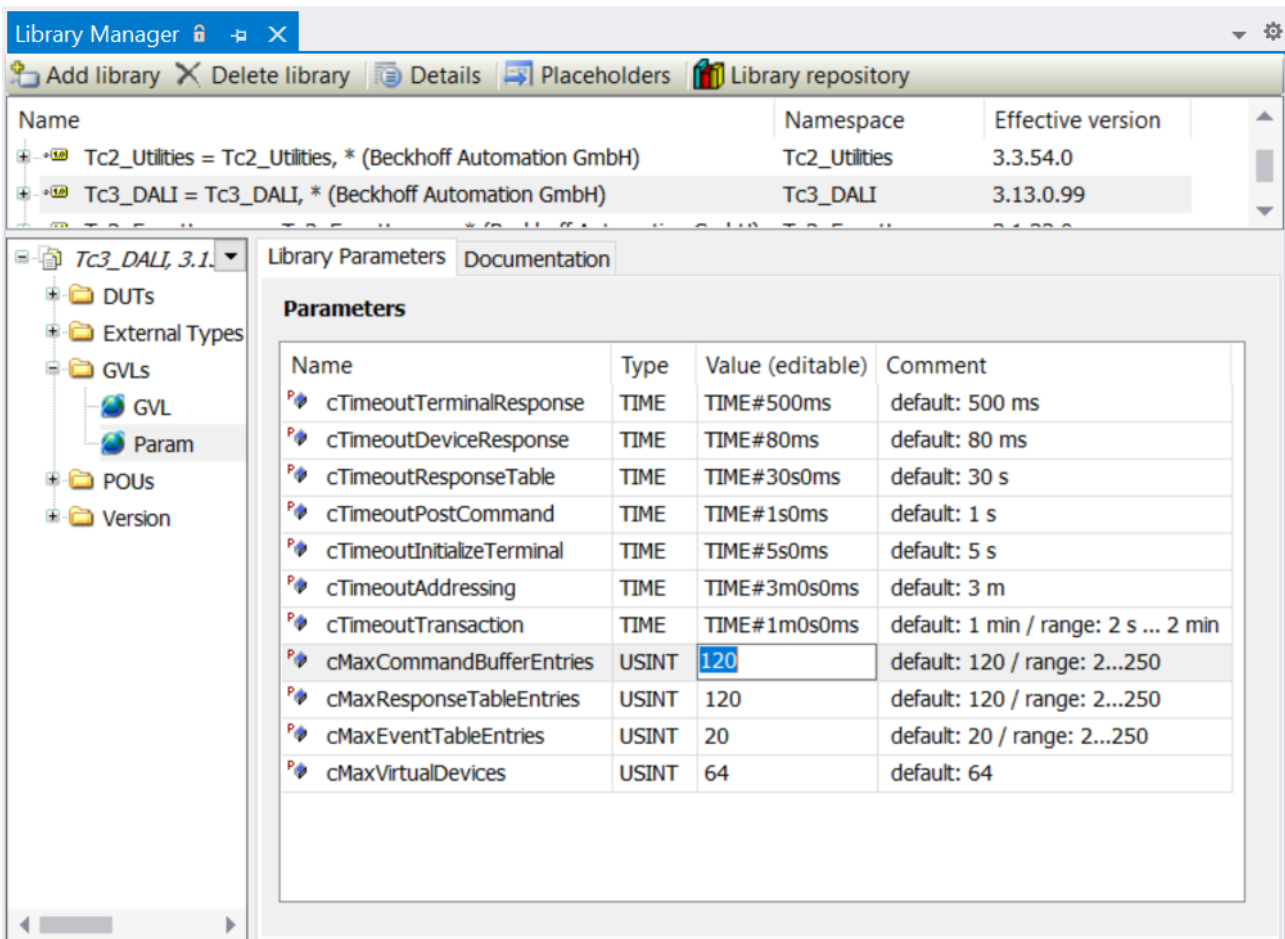
VAR_GLOBAL CONSTANT
cTimeoutTerminalResponse   : TIME := T#500MS;
cTimeoutDeviceResponse     : TIME := T#80MS;
cTimeoutResponseTable      : TIME := T#30S;
cTimeoutPostCommand        : TIME := T#1S;
cTimeoutInitializeTerminal : TIME := T#5S;
cTimeoutAddressing         : TIME := T#3M;
cTimeoutTransaction        : TIME := T#1M;
cMaxCommandBufferEntries   : USINT := 120;
cMaxResponseTableEntries   : USINT := 120;
cMaxEventTableEntries      : USINT := 20;
cMaxVirtualDevices         : USINT := 64;
END_VAR

```

The Tc3\_DALI library contains some parameters which can be used to change general settings. The parameter list can be accessed via the Library Manager. The values of the individual parameters can also be adjusted there.

### NOTICE

Note that some parameters have a profound effect on the way the Tc3\_DALI library works.



## 4.4 Integration into TwinCAT

### 4.4.1 EL6821 with CX5120

This sample explains how to write a simple PLC program for DALI in TwinCAT and how to link it with the hardware.

A single lamp is to be controlled and switched to the maximum output value or switched off with a push button.

Sample: [https://infosys.beckhoff.com/content/1033/TcPlcLib\\_Tc3\\_DALI/Resources/13899865611/.zip](https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc3_DALI/Resources/13899865611/.zip)



The TwinCAT project is available for download as \*.zip file. This must first be unpacked locally so that the archive (\*.tnzip file) is available for import into the TwinCAT project.

#### Hardware

##### Setting up the components

- 1x CX5120 Embedded PC
- 1x EL1008 digital 8-channel input terminal (for the switch-on/switch-off function)
- 1 x power supply terminal EL9562
- 1x DALI terminal EL6821
- 1 x end terminal EL9011

Set up the hardware and the DALI components as described in the documentation.

This sample assumes that an On button was connected to the first EL1008 input and an Off button to the second. There is a lamp at DALI address 0.

**Software**

**Creation of the PLC program**

Create a new **TwinCAT XAE Project** and a **Standard PLC Project**. Add the Tc3\_DALI library in the PLC project under **References**. Generate a global variable list with the name *GVL\_DALI* and create the following variables:

```
VAR_GLOBAL
  bSwitchOn          AT %I* : BOOL;
  bSwitchOff         AT %I* : BOOL;
  stEL6821InData     AT %I* : ST_EL6821InData;
  stEL6821OutData    AT %Q* : ST_EL6821OutData;
END_VAR
```

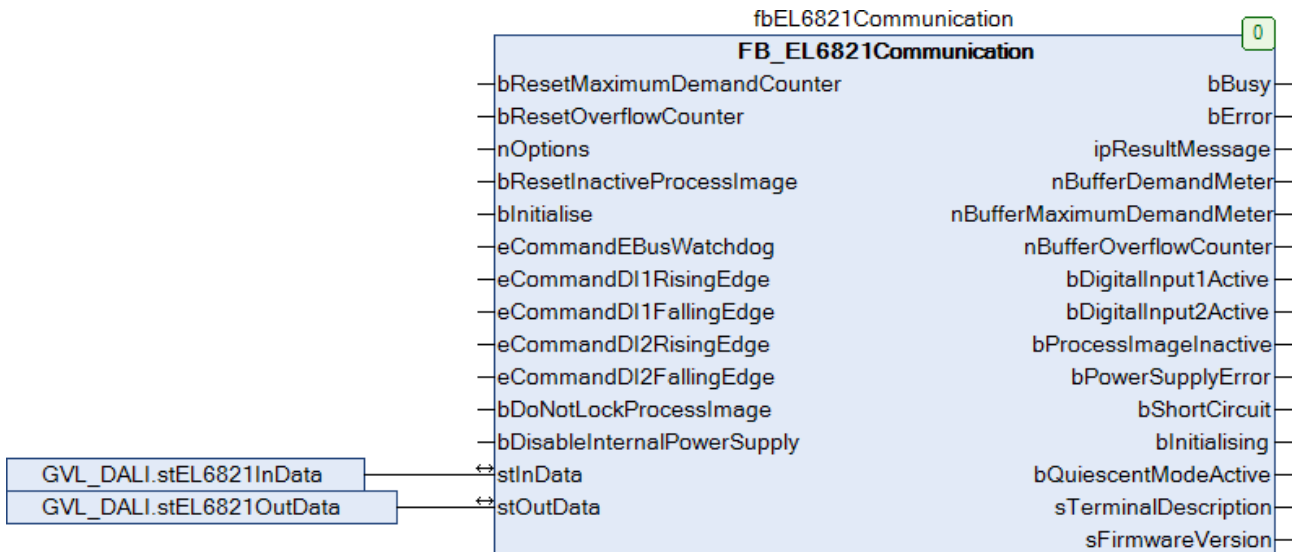
bSwitchOn: Input variable for the On button.

bSwitchOff: Input variable for the Off button.

stEL6821InData Input variable for the DALI terminal (see [ST\\_EL6821InData \[► 853\]](#)).

stEL6821OutData: Output variable for the DALI terminal (see [ST\\_EL6821OutData \[► 853\]](#)).

Create a program (CFC) for the background communication with the DALI terminal. The function block [FB\\_EL6821Communication \[► 578\]](#) is called in the program. In the communication block ensure that the structures *stInData* and *stOutData* are linked.



Create a MAIN program (CFC) in which the function blocks [FB\\_DALI102RecallMaxLevel \[► 91\]](#) and [FB\\_DALI102Off \[► 88\]](#) are declared as follows.

```
PROGRAM MAIN
VAR
  fb102RecallMaxLevel : FB_DALI102RecallMaxLevel (Communication.fbEL6821Communication);
  fb102Off            : FB_DALI102Off (Communication.fbEL6821Communication);
END_VAR
```

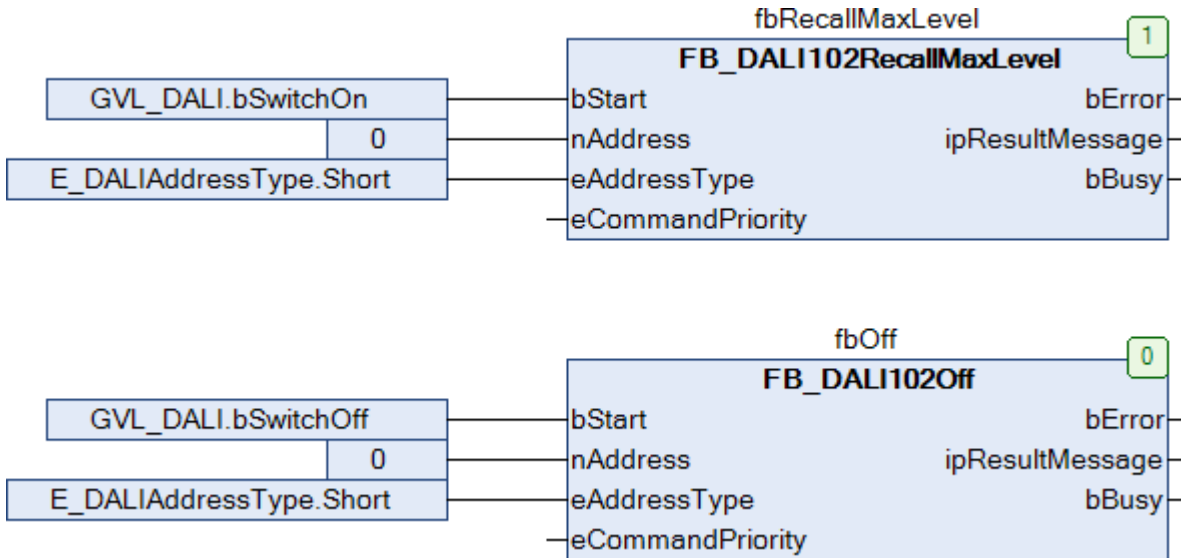
The communication block is specified in the round brackets after the declaration. The reference to the desired DALI terminal is defined via this specification.

For more information, see [Transfer of the reference to the communication block \[► 892\]](#).

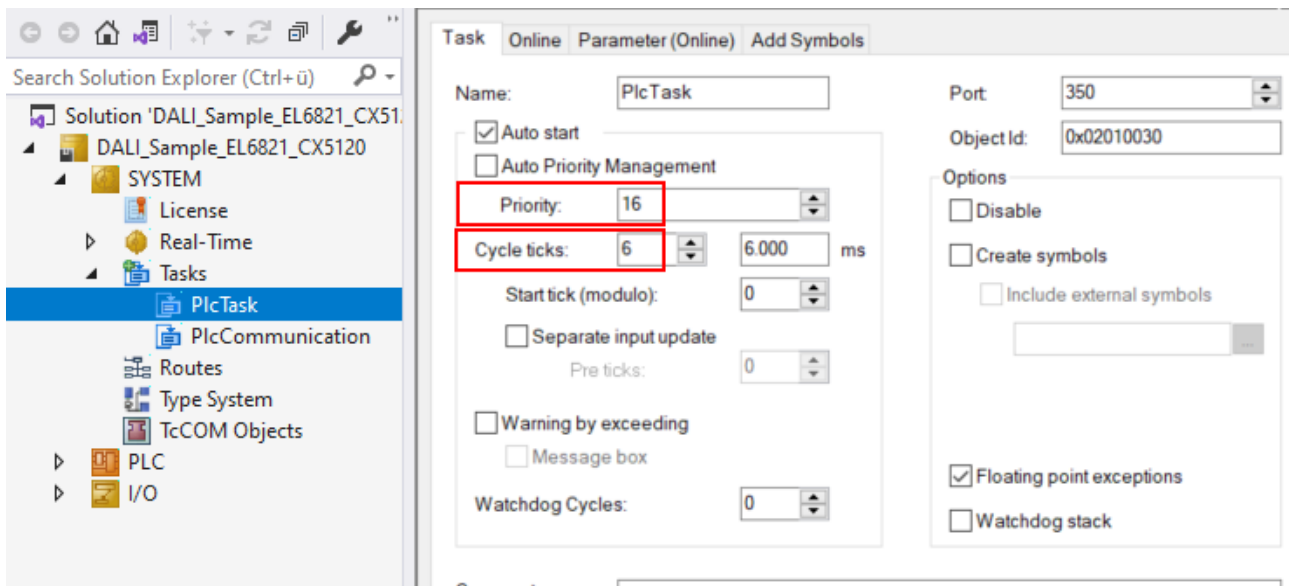
Call the two instances of the function blocks [FB\\_DALI102RecallMaxLevel \[► 91\]](#) and [FB\\_DALI102Off \[► 88\]](#) with the following variables.

The input *bStart* of the function block for switching on a lamp with the maximum output value is linked to the global variable *bSwitchOn*.

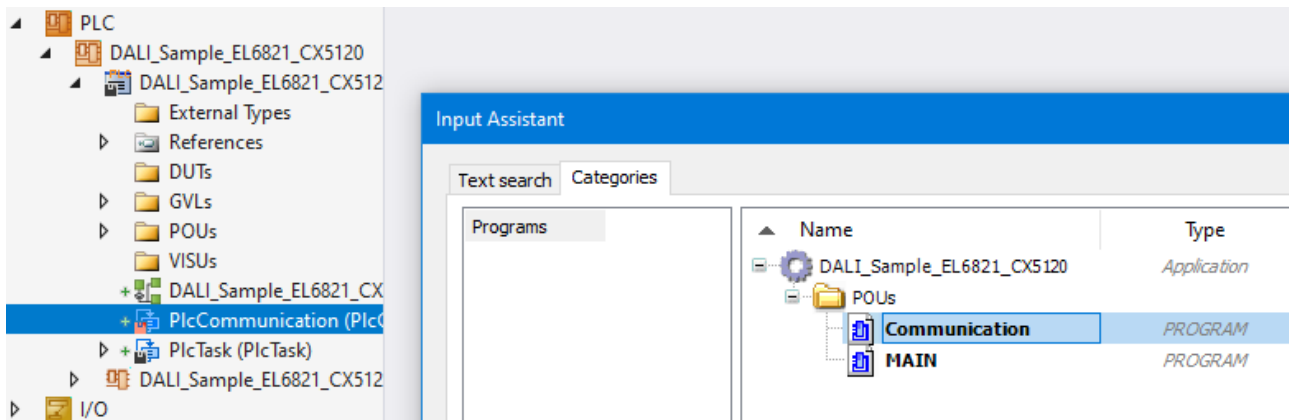
The input *bStart* of the function block for switching off a lamp is linked to the global variable *bSwitchOff*.



Navigate to the task configuration section and configure the **PicTask**. By way of example, the task is assigned priority 16 and a cycle time of 6 ms.



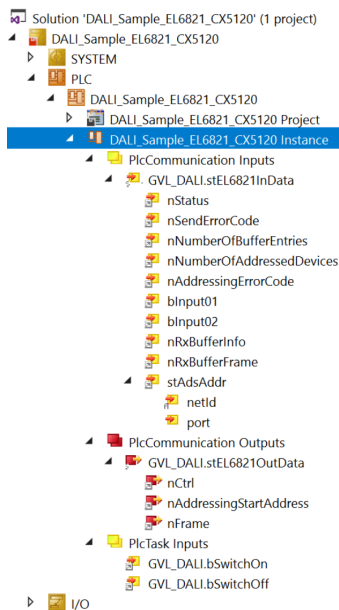
Add the program for the communication to this task. Further information on task configuration can be found in the description of the function block [FB\\_EL6821Communication](#) [▶ 578].



## I/O configuration

Select the CX as target system and initiate a search for its hardware.

In the instance (**DALI\_Sample\_EL6821\_CX5120 Instance**) of the PLC project, you can see that the input and output variables of the PLC program are assigned to the corresponding tasks (**PlcCommunication** and **PlcTask**).



Now link the global variables of PLC program with the inputs and outputs of the bus terminals. Create the Solution and enable the configuration.

The lamp with the maximum brightness value is switched on by pressing the first push button. The second push button can be used to switch it off again.

### 4.4.2 KL6821 with CX5120

This sample explains how to write a simple PLC program for DALI in TwinCAT and how to link it with the hardware.

A single lamp is to be controlled and switched to the maximum output value or switched off with a push button.

Sample: [https://infosys.beckhoff.com/content/1033/TcPlcLib\\_Tc3\\_DALI/Resources/6011100683/.zip](https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc3_DALI/Resources/6011100683/.zip)



The TwinCAT project is available for download as \*.zip file. This must first be unpacked locally so that the archive (\*.tzip file) is available for import into the TwinCAT project.

## Hardware

### Setting up the components

- 1x CX5120 Embedded PC
- 1x KL1104 digital 4-channel input terminal (for the switch-on/switch-off function)
- 1x [KL6821](#) DALI terminal
- 1x KL9010 end terminal

Set up the hardware and the DALI components as described in the documentation.

This sample assumes that an On button was connected to the first KL1104 input and an Off button to the second. There is a lamp at DALI address 0.



Software

Creation of the PLC program

Create a new **TwinCAT XAE Project** and a **Standard PLC Project**. Add the Tc3\_DALI library in the PLC project under **References**. Generate a global variable list with the name *GVL\_DALI* and create the following variables:

```
VAR_GLOBAL
  bSwitchOn           AT %I* : BOOL;
  bSwitchOff          AT %I* : BOOL;
  stKL6821InData      AT %I* : ST_KL6821InData;
  stKL6821OutData     AT %Q* : ST_KL6821OutData;
END_VAR
```

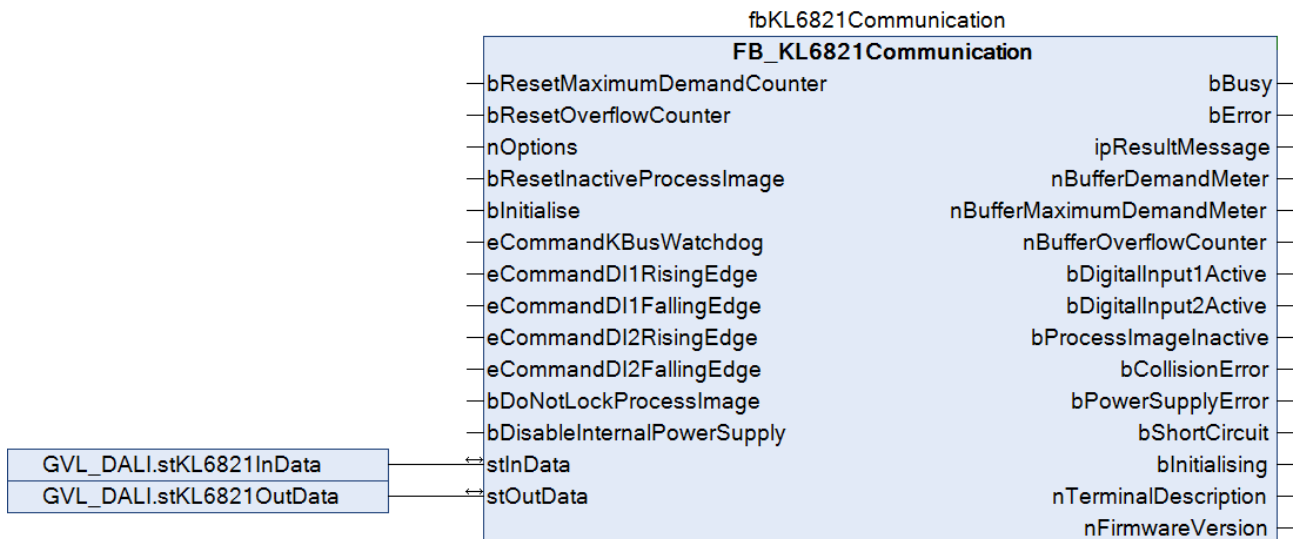
**bSwitchOn:** Input variable for the On button.

**bSwitchOff:** Input variable for the Off button.

**stKL6821InData:** Input variable for the DALI terminal (see [ST\\_KL6821InData \[► 854\]](#)).

**stKL6821OutData:** Output variable for the DALI terminal (see [ST\\_KL6821OutData \[► 854\]](#)).

Create a program (CFC) for the background communication with the DALI terminal. The function block [FB\\_KL6821Communication \[► 586\]](#) is called in the program. In the communication block ensure that the structures *stInData* and *stOutData* are linked.



Create a MAIN program (CFC) in which the [FB\\_DALI102RecallMaxLevel \[► 91\]](#) and [FB\\_DALI102Off \[► 88\]](#) function blocks are declared as follows.

```
PROGRAM MAIN
VAR
  fb102RecallMaxLevel : FB_DALI102RecallMaxLevel (Communication.fbKL6821Communication);
  fb102Off            : FB_DALI102Off (Communication.fbKL6821Communication);
END_VAR
```

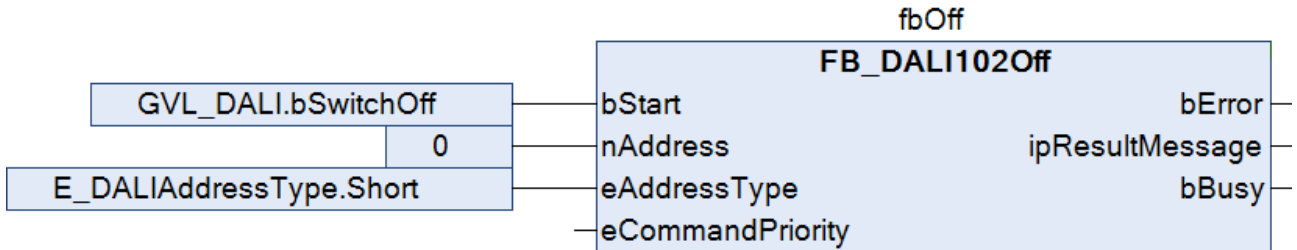
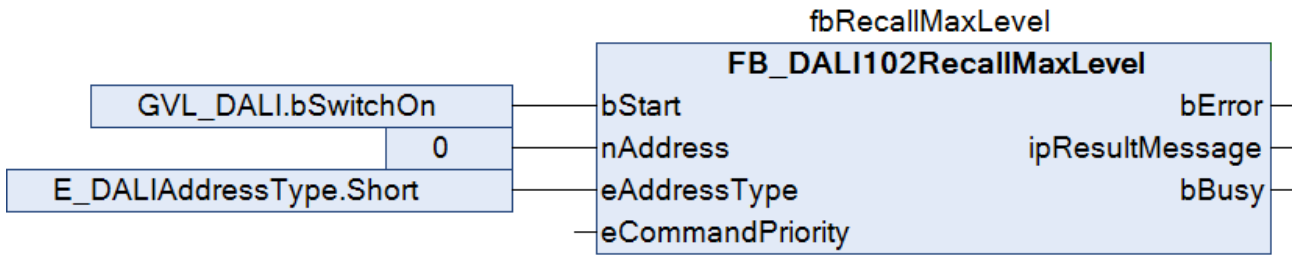
The communication block is specified in the round brackets after the declaration. The reference to the desired DALI terminal is defined via this specification.

For more information see: [Transfer of the reference to the communication block \[► 892\]](#).

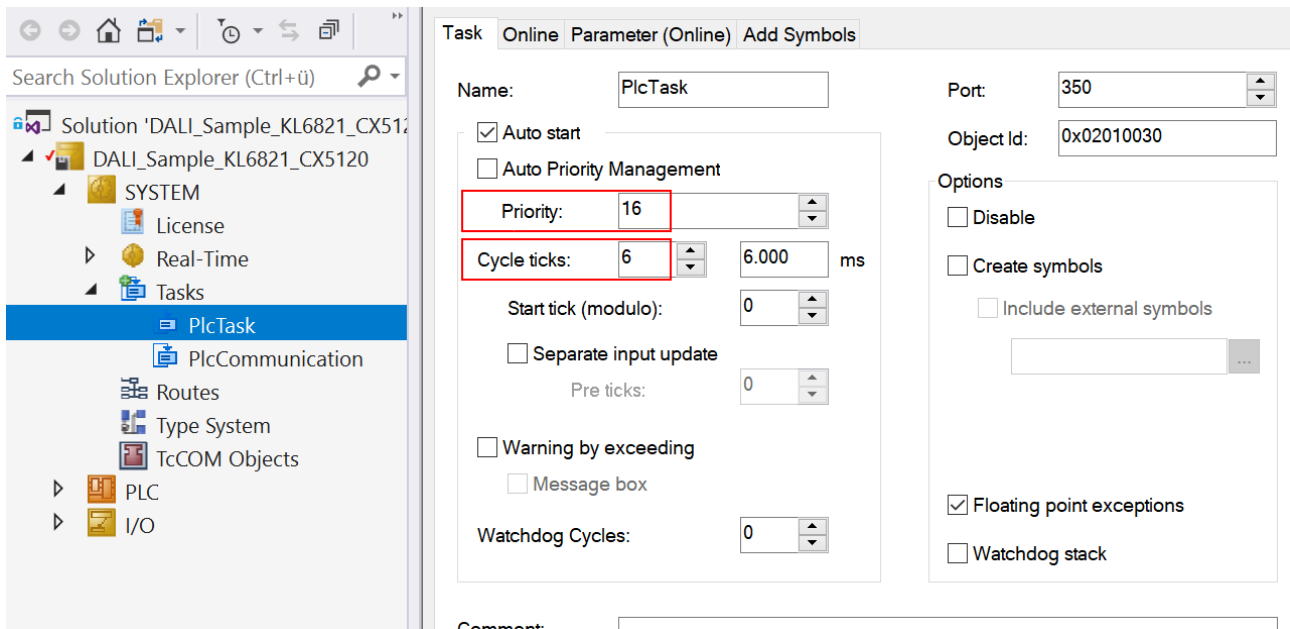
Call the two instances of the function blocks [FB\\_DALI102RecallMaxLevel \[► 91\]](#) and [FB\\_DALI102Off \[► 88\]](#) with the following variables.

The input *bStart* of the function block for switching on a lamp with the maximum output value is linked to the global variable *bSwitchOn*.

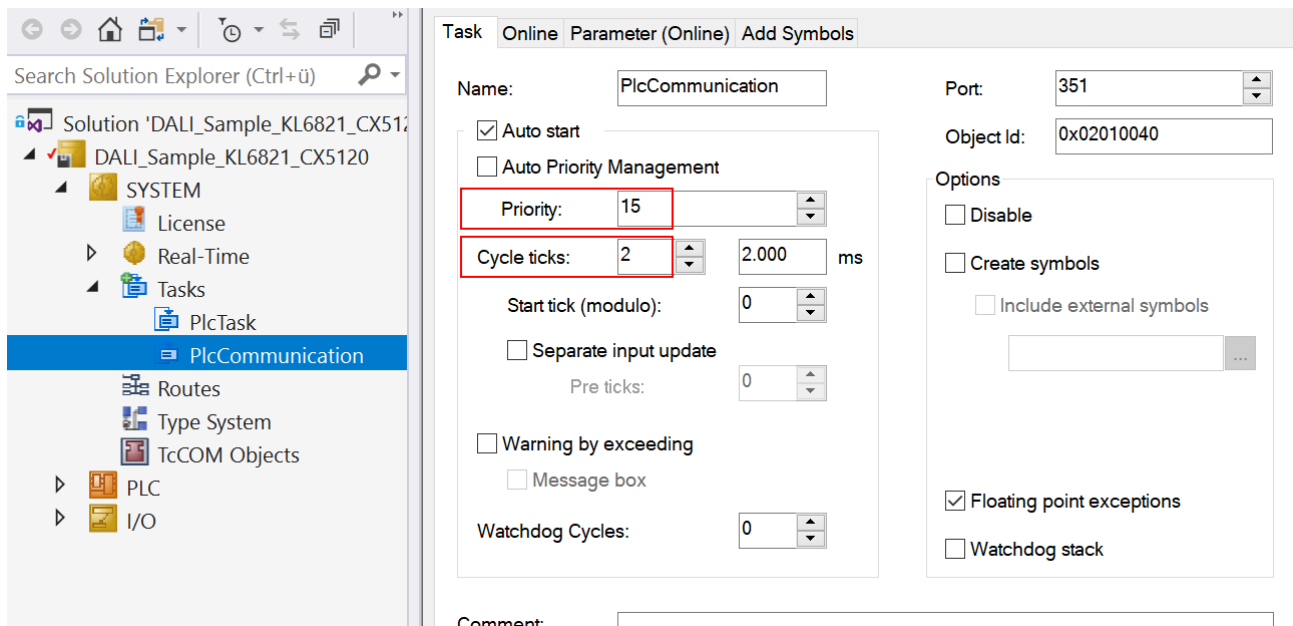
The input *bStart* of the function block for switching off a lamp is linked to the global variable *bSwitchOff*.



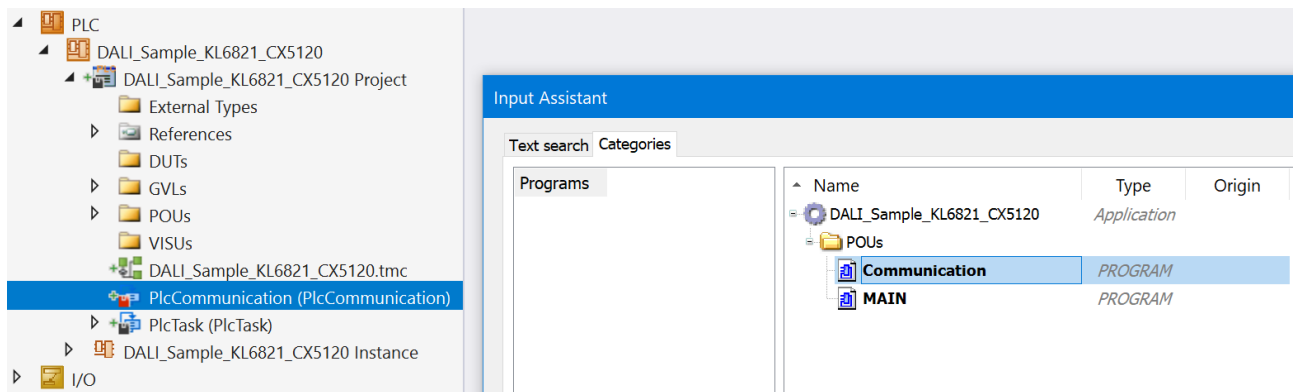
Navigate to the task configuration section and configure the **PlcTask**. By way of example, the task is assigned priority 16 and a cycle time of 6 ms.



Create a further task for the background communication. Assign a higher priority (smaller number) and a lower interval time to this task than the PlcTask.



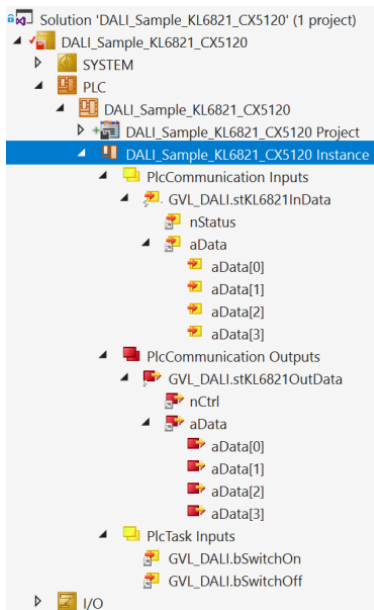
Add the program for the communication to this task. Further information on task configuration can be found in the description of the function block [FB\\_KL6821Communication](#) [► 586].



### I/O configuration

Select the CX as target system and initiate a search for its hardware.

In the instance (**DALI\_Sample\_KL6821\_CX5120 Instance**) of the PLC project, you can see that the input and output variables of the PLC program are assigned to the corresponding tasks (**PlcCommunication** and **PlcTask**).



Now link the global variables of PLC program with the inputs and outputs of the bus terminals. Create the Solution and enable the configuration.

The lamp with the maximum brightness value is switched on by pressing the first push button. The second push button can be used to switch it off again.

## 4.5 Runtime messages

Value (hex)	Value (dec)	Text ID	Description
16#0000	0	NoError	No error.
16#0001	1	NoResponseFromTheDALITerminal	No response from the DALI terminal.
16#0002	2	NoResponseFromTheDALIDevice	No response from the DALI device.
16#0003	3	CommandBufferOverflow	Overflow of the command buffer.
16#0004	4	NoAnswerFromTheCommunicationBlock	No response from the communication block.
16#0005	5	DALICollisionDetectedOnTheBackwardChannel	DALI collision detected on the backward channel: during the transfer of a DALI frame, a collision with the send data of another DALI device was detected.
16#0006	6	DALICollisionDetectedOnTheForwardChannel	DALI collision detected on the forward channel: during the transfer of a DALI frame, a collision with the send data of another DALI control device was detected. The error also occurs if the 24 V supply is missing at the power contacts of the KL6811.
16#0007	7	OverloadOfTheInternalDALIPowerSupplyUnitOfTheBusTerminal_BusUnderVoltage	When using the internal DALI power supply unit of the bus terminal: overload of the internal DALI power supply unit of the bus terminal (bus undervoltage).
16#0008	8	PowerSupplyUnitFaultDetected	When using the internal DALI power supply unit: Power supply unit fault detected.
16#0009	9	TheProcessImageWasDisabledByDI1OrDI2	The process image was disabled by the DI1 or DI2 inputs of the terminal.
16#000A	10	ShortCircuitDetectedOnTheDALIBus	The 24 V DC supply voltage at connections 1 and 5 of the KL6821/EL6821 is missing, or a short circuit has been detected on the DALI bus.
16#000B	11	CollisionErrorOnTheDALIBus	Due to collisions on the DALI bus, the DALI command could not be sent.
16#000C	12	ItemInReceiveBufferIsInvalid	The entry in the receive buffer of the KL6821/EL6821 is invalid.
16#000D	13	TheInterfaceToTheCommunicationBufferIsNotInitialized	The interface pointer on the communication block is not initialized.
16#000E	14	TheCommandBuffersHaveBeenBlockedForLongerThanPermitted	The command buffer was blocked for longer than is permitted.
16#000F	15	TheTerminalHasReturnedAnErrorDuringInternalAddressing	The terminal has returned an error during addressing.
16#0010	16	DuringInternalAddressingTheTerminalHasDetectedAShortCircuitOnTheBus	During addressing the terminal has detected a short circuit on the bus.
16#0011	17	DuringInternalAddressingTheTerminalHasDetectedThatThereIsNoFurtherShortAddressAvailable	During addressing the terminal has detected that there is no free short address available.
16#0012	18	DuringInternalAddressingTheTerminalHasDetectedThatSeveralDevicesHaveTheSameLongAddress	During addressing the terminal has detected that several devices have the same long address.
16#0013	19	InternalAddressingHasFailed3Times	The addressing failed three times.
16#0014	20	TimeoutDuringInternalAddressing	Timeout for addressing. The terminal has not sent a response following the start of addressing.
16#0015	21	NoResponseFromTheDALIDeviceInstance	No response from the instance of the DALI device.
16#0016	22	TheKL6811DoesNotSupportThisDALICommand	The KL6811 supports only 16-bit commands (IEC 62386-102 and IEC 62386-2xx).
16#0017	23	ErrorDuringTheConfigurationOfTheTerminal	An error occurred during configuration of the terminal.
16#0018	24	NoResponseFromTheDALIDevice2	No response from the DALI device.
16#0019	25	NoFreeShortAddressAvailable	No more free short addresses are available.
16#001A	26	ErrorDuringAddressing	An error occurred during addressing.
16#001B	27	InvalidResponseFromTheDALIDevice	The response from the DALI device is invalid.
16#001C	28	InvalidResponseFromTheDALIDeviceInstance	Invalid response from the instance of the DALI device.
16#001D	29	ThisDALICommandIsNotSupportedByTheSimulation	This DALI command is not supported by the simulation.
16#001E	30	CommandBuffersBlockedByMutex	Access to the command buffer is not possible due to blocking by the mutex.
16#001F	31	ItemInReceiveBufferErrorBitTiming	The entry in the receive buffer of the KL6821/EL6821 has a bit timing error.
16#0020	32	ItemInReceiveBufferIncorrectNumberOfBits	The entry in the receive buffer of the KL6821/EL6821 has an invalid number of bits.
16#0021	33	ItemInReceiveBufferEncodingError	An error has occurred in the entry in the receive buffer of the KL6821/EL6821 during decoding.

Value (hex)	Value (dec)	Text ID	Description
16#0022	34	DuringAddressingTheTerminalHasDetectedMissingResponse	During addressing, the terminal has detected a missing response from a device.
16#00B5	181	WritingToTheOffsetOfTheMemoryBankWasNotSuccessful	Writing to the offset of a memory bank of a DALI device was not successful.
16#00B6	182	ParameterColourTypeLiesOutsideOfTheValidRange	Parameter <i>eColourType</i> [► 855] lies outside of the valid range (E_DALIColourType.XyCoordinate...E_DALIColourType.RGBWAFControl).
16#00B7	183	ParameterEmergencyLevelLiesOutsideOfTheValidRange	Parameter <i>nEmergencyLevel</i> is outside the valid range ( <i>emergencyMinLevel</i> ... <i>emergencyMaxLevel</i> , 255).
16#00B8	184	ParameterDurationTestIntervalLiesOutsideOfTheValidRange	Parameter <i>nDurationTestInterval</i> is outside the valid range (0...97).
16#00B9	185	ParameterSensitivityOccupancyPartial3Part303LiesOutsideOfTheValidRange	Parameter <i>nSensitivityN</i> lies outside of the valid range (1...100).
16#00BA	186	ParameterSensitivityOccupancyPartial2Part303LiesOutsideOfTheValidRange	Parameter <i>nSensitivityN</i> lies outside of the valid range (0...4).
16#00BB	187	ParameterExtendedFadeTimeBaseLiesOutsideOfTheValidRange	Parameter <i>eExtendedFadeTimeBase</i> is outside the valid range (E_DALIExtendedFadeTimeBase.Base01...E_DALIExtendedFadeTimeBase.Base16).
16#00BC	188	ParameterExtendedFadeTimeMultiplierLiesOutsideOfTheValidRange	Parameter <i>eExtendedFadeTimeMultiplier</i> is out of valid range (E_DALIExtendedFadeTimeMultiplier.Disabled...E_DALIExtendedFadeTimeMultiplier.Multiplier1min).
16#00BD	189	ParameterDeadtimeLiesOutsideOfTheValidRange	Parameter <i>tDeadtime</i> lies outside of the valid range (2...3600 s).
16#00BE	190	ParameterHysteresisLiesOutsideOfTheValidRange2	Parameter <i>nHysteresis</i> lies outside of the valid range (0...25 %).
16#00BF	191	ParameterDeadtimeTimerLiesOutsideOfTheValidRange	Parameter <i>nDeadtimeTimer</i> lies outside of the valid range (0...12750 ms).
16#00C0	192	ParameterHoldTimerLiesOutsideOfTheValidRange	Parameter <i>nHoldTimer</i> lies outside of the valid range (1...2538 ms).
16#00C1	193	ParameterStuckTimerLiesOutsideOfTheValidRange	Parameter <i>nStuckTimer</i> lies outside of the valid range (5...255 s).
16#00C2	194	ParameterShortLiesOutsideOfTheValidRange	Parameter <i>nShort</i> lies outside of the valid range (tShortMin...255).
16#00C3	195	ParameterShortTimerPart301LiesOutsideOfTheValidRange	Parameter <i>nShortTimerN</i> lies outside of the valid range (tShortMin...5100 ms).
16#00C4	196	ParameterShortTimerLiesOutsideOfTheValidRange	Parameter <i>nShortTimer</i> lies outside of the valid range (tShortMin...5100 ms).
16#00C5	197	ParameterDoubleTimerPart301LiesOutsideOfTheValidRange	Parameter <i>nDoubleTimerN</i> lies outside of the valid range (0, tDoubleMin...2000 ms).
16#00C6	198	ParameterDoubleTimerLiesOutsideOfTheValidRange	Parameter <i>nDoubleTimer</i> lies outside of the valid range (0, tDoubleMin...2000 ms).
16#00C7	199	ParameterRepeatTimerLiesOutsideOfTheValidRange	Parameter <i>nRepeatTimer</i> lies outside of the valid range (100...2000 ms).
16#00C8	200	ParameterEventSchemeLiesOutsideOfTheValidRange	Parameter <i>eEventScheme</i> is outside the valid range (see E_DALIEventScheme [► 859]).
16#00C9	201	ParameterAddressIsAGroupAddressAndLiesOutsideOfTheValidRange2	Parameter <i>nAddress</i> is a group address and lies outside of the valid range (0...31).
16#00CA	202	ParameterEventPriorityDeviceLiesOutsideOfTheValidRange	Parameter <i>eEventPriorityN</i> lies outside of the valid range (E_DALIEventPriority.Low...E_DALIEventPriority.MiddleHigh).
16#00CB	203	ParameterALSCorrectionFactorPart304LiesOutsideOfTheValidRange	Parameter <i>nALSxCorrectionFactorN</i> lies outside of the valid range (0.1...5.0).
16#00CC	204	ParameterRoomCorrectionFactorPart304LiesOutsideOfTheValidRange	Parameter <i>nRoomCorrectionFactorN</i> lies outside of the valid range (5...200).
16#00CD	205	ParameterSwitchOnLevelLiesOutsideOfTheValidRange	Parameter <i>nSwitchOnLevel</i> lies outside of the valid range (minLevel...maxLevel [► 150]).
16#00CE	206	ParameterHysteresisPart304LiesOutsideOfTheValidRange	Parameter <i>nHysteresisN</i> lies outside of the valid range (0...25 %).
16#00CF	207	ParameterDeadtimeTimerPart304LiesOutsideOfTheValidRange	Parameter <i>nDeadtimeTimerN</i> lies outside of the valid range (0...12750 ms).

Value (hex)	Value (dec)	Text ID	Description
16#00D0	208	ParameterSensitivityOccupancyPartialPart303LiesOutsideOfTheValidRange	Parameter <i>nSensitivityN</i> lies outside of the valid range (1...5).
16#00D1	209	ParameterSensitivityOccupancyPercentPart303LiesOutsideOfTheValidRange	Parameter <i>nSensitivityN</i> lies outside of the valid range (0...100 %).
16#00D2	210	ParameterDetectionRangeOccupancyPart303LiesOutsideOfTheValidRange	Parameter <i>nDetectionRangeN</i> lies outside of the valid range (0...100 %).
16#00D3	211	ParameterHoldTimerPart303LiesOutsideOfTheValidRange	Parameter <i>nHoldTimerN</i> lies outside of the valid range (1...2538 ms).
16#00D4	212	ParameterDeadtimeTimerPart303LiesOutsideOfTheValidRange	Parameter <i>nDeadtimeTimerN</i> lies outside of the valid range (0...12750 ms).
16#00D5	213	ParameterHysteresisLiesOutsideOfTheValidRange3	Parameter <i>nHysteresis</i> lies outside of the valid range (1...65535).
16#00D6	214	ParameterDeadtimeTimerPart302LiesOutsideOfTheValidRange	Parameter <i>nDeadtimeTimerN</i> lies outside of the valid range (0...12750 ms).
16#00D7	215	ParameterStuckTimerPart301LiesOutsideOfTheValidRange	Parameter <i>nStuckTimerN</i> lies outside of the valid range (5...255 ms).
16#00D8	216	ParameterRepeatTimerPart301LiesOutsideOfTheValidRange	Parameter <i>nRepeatTimerN</i> lies outside of the valid range (100...2000 ms).
16#00D9	217	ParameterUpSwitchOffThresholdLiesOutsideOfTheValidRange	Parameter <i>nUpSwitchOffThreshold</i> lies outside of the valid range (1...255).
16#00DA	218	ParameterUpSwitchOnThresholdLiesOutsideOfTheValidRange	Parameter <i>nUpSwitchOnThreshold</i> lies outside of the valid range (1...255).
16#00DB	219	ParameterLevelMemoryModeLiesOutsideOfTheValidRange	Parameter <i>nLevelMemoryMode</i> lies outside of the valid range (1...254).
16#00DC	220	ParameterMinLevelsGreaterThanMaxLevel	Parameter <i>nMinLevel</i> is greater than <i>nMaxLevel</i> .
16#00DD	221	ParameterReferenceDeviceAddressLiesOutsideOfTheValidRange	Parameter <i>nReferenceDeviceAddress</i> lies outside of the valid range (0...63).
16#00DE	222	ErrorDuringReadingOffsetFromMemoryBank	An error occurred while reading an offset from the memory bank.
16#00DF	223	ParameterSubRangeStartLiesOutsideOfTheValidRange	Parameter <i>nSubRangeStart</i> lies outside of the valid range.
16#00E0	224	ErrorReadingMemoryBankDTR0IsWrong	An error occurred while reading out a memory bank. The value of DTR0 is not as expected.
16#00E1	225	ParameterSubRangeEndLiesOutsideOfTheValidRange	Parameter <i>nSubRangeEnd</i> lies outside of the valid range.
16#00E2	226	ReadCommandReturnedTMASK	The DALI device returns TMASK. The value is not currently available.
16#00E3	227	ReadCommandReturnedMASK	The DALI device returns MASK. The value is not available.
16#00E4	228	ParameterInternalControlGearReferenceTemperatureLiesOutsideOfTheValidRange	Parameter <i>nInternalControlGearReferenceTemperature</i> lies outside of the valid range (-60...193 °C).
16#00E5	229	ParameterLightSourceOnTimeResetableLiesOutsideOfTheValidRange	Parameter <i>nLightSourceOnTimeResetable</i> lies outside of the valid range (0...4294967293 s).
16#00E6	230	ParameterLightSourceStartCounterResetableLiesOutsideOfTheValidRange	Parameter <i>nLightSourceStartCounterResetable</i> lies outside of the valid range (0...16777213).
16#00E7	231	ParameterRatedMedianUsefulLifeOfLuminaireLiesOutsideOfTheValidRange	Parameter <i>nRatedMedianUsefulLifeOfLuminaire</i> lies outside of the valid range (0...253).
16#00E8	232	ParameterRatedMedianUsefulLightSourceStartsLiesOutsideOfTheValidRange	Parameter <i>nRatedMedianUsefulLightSourceStarts</i> lies outside of the valid range (0...65533).
16#00E9	233	ParameterInstanceNumberLiesOutsideOfTheValidRange	Parameter <i>nInstanceNumber</i> lies outside of the valid range (0...31).
16#00EA	234	ParameterShortAddressLiesOutsideOfTheValidRange2	Parameter <i>nShortAddress</i> lies outside of the valid range (0...63).
16#00EB	235	ParameterSubRangeEndIsLessThanSubRangeStart	Parameter <i>nSubRangeEnd</i> is smaller than <i>nSubRangeStart</i> .
16#00EC	236	ParameterFastFadeTimeLiesOutsideOfTheValidRange	Parameter <i>eFastFadeTime</i> is out of valid range ( <i>E_DALIFastFadeTime.Disabled...E_DALIFastFadeTime.T700ms</i> ).
16#00ED	237	ParameterDimmingCurveLiesOutsideOfTheValidRange	Parameter <i>eDimmingCurve</i> is outside the valid range (see <a href="#">E_DALIDimmingCurve</a> [► 858]).
16#00EE	238	ParameterEventPriorityLiesOutsideOfTheValidRange	Parameter <i>eEventPriority</i> lies outside of the valid range ( <i>E_DALIEventPriority.Low...E_DALIEventPriority.MiddleHigh</i> ).



Value (hex)	Value (dec)	Text ID	Description
16#00EF	239	ParameterShortAddressLiesOutsideOfTheValidRange	Parameter <i>nShortAddress</i> lies outside of the valid range (0...63, 255).
16#00F0	240	ParameterAddressInfoInstanceGroupLiesOutsideOfTheValidRange	Parameter <i>nAddressInfo1/nAddressInfo2</i> defines an instance group and lies outside of the valid range (0...31).
16#00F1	241	ParameterAddressInfoDeviceGroupLiesOutsideOfTheValidRange	Parameter <i>nAddressInfo1/nAddressInfo2</i> defines a device group and lies outside of the valid range (0...31).
16#00F2	242	ParameterAddressInfoShortAddressLiesOutsideOfTheValidRange	Parameter <i>nAddressInfo1/nAddressInfo2</i> defines a short address and lies outside of the valid range (0...63).
16#00F3	243	ParameterAddressInfoInstanceNumberLiesOutsideOfTheValidRange	Parameter <i>nAddressInfo1/nAddressInfo2</i> defines an instance number and lies outside of the valid range (0...31).
16#00F4	244	ParameterAddressInfoInstanceTypeLiesOutsideOfTheValidRange	Parameter <i>nAddressInfo1/nAddressInfo2</i> defines an instance type and lies outside of the valid range (0...31).
16#00F5	245	ParameterHysteresisLiesOutsideOfTheValidRange	Parameter <i>nHysteresis</i> lies outside of the valid range (0...25).
16#00F6	246	ParameterHoldLiesOutsideOfTheValidRange	Parameter <i>nHold</i> lies outside of the valid range (0...254).
16#00F7	247	ParameterDoubleLiesOutsideOfTheValidRange	Parameter <i>nDouble</i> lies outside of the valid range (0, <i>tDoubleMin</i> ...100).
16#00F8	248	ParameterRepeatLiesOutsideOfTheValidRange	Parameter <i>nRepeat</i> lies outside of the valid range (5...100).
16#00F9	249	ParameterStuckLiesOutsideOfTheValidRange	Parameter <i>nStuck</i> lies outside of the valid range (5...255).
16#00FA	250	ParameterResolutionLiesOutsideOfTheValidRange	Parameter <i>nResolution</i> lies outside of the valid range (1...64).
16#00FB	251	ParameterEventFilterLiesOutsideOfTheValidRange	Parameter <i>nEventFilter</i> lies outside of the valid range (16#00_0000...16#FF_FFFF).
16#00FC	252	ParameterChangeAddressListIsEmpty	Parameter <i>aChangeAddressList</i> is empty.
16#00FD	253	ParameterChangeAddressListContainsAnInvalidShortAddressEntry	Parameter <i>aChangeAddressList</i> contains an invalid short address entry.
16#00FE	254	ParameterChangeAddressListContainsADoubleListEntryInTheShortAddresses	Parameter <i>aChangeAddressList</i> contains a double list entry in the short addresses.
16#00FF	255	ParameterChangeAddressListContainsAnEntryForANewShortAddressWhichIsAlreadyAssigned	Parameter <i>aChangeAddressList</i> contains an entry for a new short address which is already assigned to a device that is not affected by changes. The addresses have been changed back.
16#0100	256	ParameterMaxResponseTableEntriesLiesOutsideOfTheValidRange	Parameter <i>cMaxResponseTableEntries</i> lies outside of the valid range (2...250).
16#0101	257	ParameterMaxEventTableEntriesLiesOutsideOfTheValidRange	Parameter <i>cMaxEventTableEntries</i> lies outside of the valid range (2...250).
16#0102	258	ParameterTimeoutLockCommandBuffersLiesOutsideOfTheValidRange	Parameter <i>cTimeoutLockCommandBuffers</i> lies outside of the valid range (2...120 s).
16#0103	259	ParameterMaxCommandBufferEntriesLiesOutsideOfTheValidRange	Parameter <i>cMaxCommandBufferEntries</i> lies outside of the valid range (2...250).
16#0104	260	ParameterDataFrameTypeLiesOutsideTheValidRange	Parameter <i>eDataFrameType</i> is outside the valid range (see <a href="#">E_DALIDataFrameType</a> [► 857]).
16#0105	261	ParameterAddressTypeLiesOutsideOfTheValidRange	Parameter <i>eAddressType</i> is outside the valid range (see <a href="#">E_DALIAddressType</a> [► 854]).
16#0106	262	ParameterAddressIsAShortAddressAndLiesOutsideOfTheValidRange	Parameter <i>nAddress</i> is a short address and lies outside of the valid range (0...63).
16#0107	263	ParameterAddressIsAGroupAddressAndLiesOutsideOfTheValidRange	Parameter <i>nAddress</i> is a group address and lies outside of the valid range (0...15).
16#0108	264	ParameterCommandPriorityLiesOutsideOfTheValidRange	Parameter <i>eCommandPriority</i> is outside the valid range ( <a href="#">E_DALICommandPriority.Low</a> ... <a href="#">E_DALICommandPriority.High</a> ).
16#0109	265	ParameterInstanceAddressTypeLiesOutsideTheValidRange	Parameter <i>eInstanceAddressType</i> is outside the valid range (see <a href="#">E_DALIInstanceAddressType</a> [► 861]).
16#010A	266	ParameterInstanceAddressIsAnInstanceNumberAndLiesOutsideOfTheValidRange	Parameter <i>nInstanceAddress</i> is an instance number and lies outside of the valid range (0...31).

Value (hex)	Value (dec)	Text ID	Description
16#010B	267	ParameterInstanceAddressIsAnInstanceGroupAndLiesOutsideOfTheValidRange	Parameter <i>nInstanceAddress</i> is an instance group and lies outside of the valid range (0...31).
16#010C	268	ParameterInstanceAddressIsAnInstanceTypeAndLiesOutsideOfTheValidRange	Parameter <i>nInstanceAddress</i> is an instance type and lies outside of the valid range (0...31).
16#010D	269	ParameterInstanceAddressIsAFeatureOnInstanceNumberLevelAndLiesOutsideOfTheValidRange	Parameter <i>nInstanceAddress</i> is a feature on instance number level and lies outside of the valid range (0...31).
16#010E	270	ParameterInstanceAddressIsAFeatureOnInstanceGroupLevelAndLiesOutsideOfTheValidRange	Parameter <i>nInstanceAddress</i> is a feature on instance group level and lies outside of the valid range (0...31).
16#010F	271	ParameterInstanceAddressIsAFeatureOnInstanceTypeLevelAndLiesOutsideOfTheValidRange	Parameter <i>nInstanceAddress</i> is a feature on instance type level and lies outside the valid range (0...31).
16#0110	272	ParameterGroupLiesOutsideOfTheValidRange	Parameter <i>nGroup</i> lies outside of the valid range (0...15).
16#0111	273	ParameterSceneLiesOutsideOfTheValidRange	Parameter <i>nScene</i> lies outside of the valid range (0...15).
16#0112	274	ParameterFadeTimeLiesOutsideOfTheValidRange	Parameter <i>eFadeTime</i> is out of valid range ( <i>E_DALIFadeTime.Disabled...E_DALIFadeTime.T90500ms</i> ).
16#0113	275	ParameterFadeRateLiesOutsideOfTheValidRange	Parameter <i>eFadeRate</i> is out of valid range ( <i>E_DALIFadeRate.N003StepsPerSec...E_DALIFadeRate.N358StepsPerSec</i> ).
16#0114	276	ParameterNewShortAddressLiesOutsideOfTheValidRange	Parameter <i>nNewShortAddress</i> lies outside of the valid range (0...63, 255).
16#0115	277	ParameterStartWithShortAddressLiesOutsideOfTheValidRange	Parameter <i>nStartWithShortAddress</i> lies outside of the valid range (0...63).

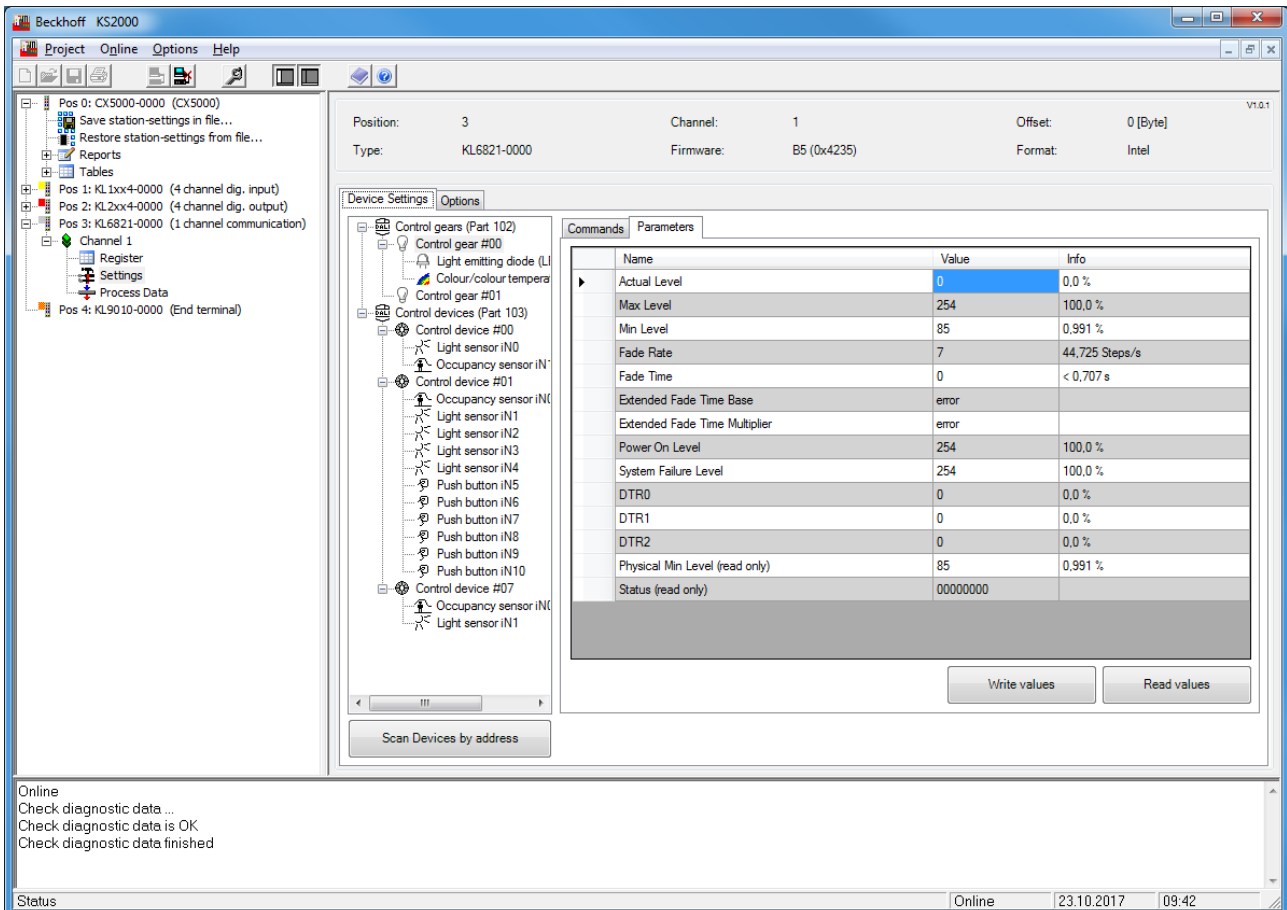
## 5 Appendix

### 5.1 Commissioning and diagnosis

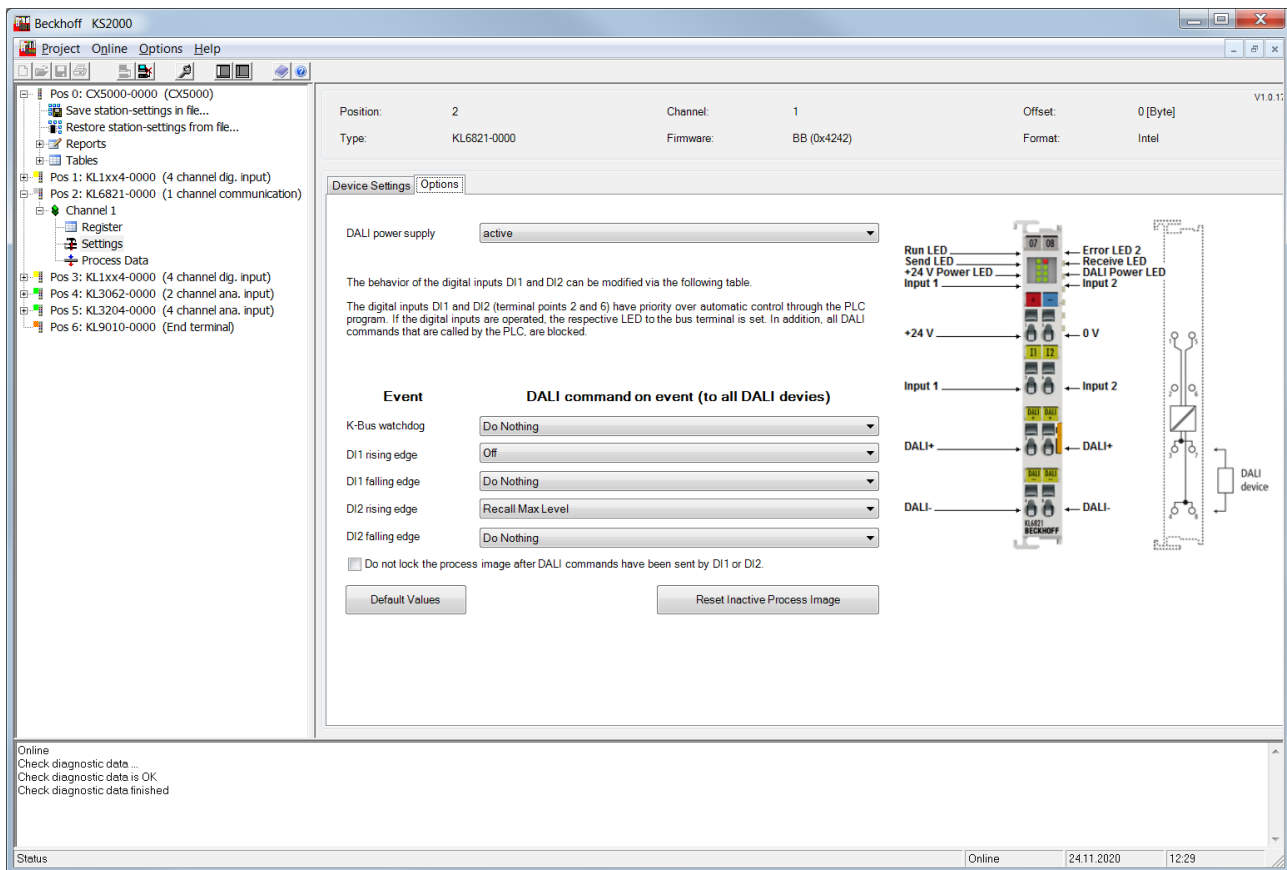
#### 5.1.1 Commissioning KL6821

The KS2000 provides dialogs for commissioning and diagnosis of DALI devices on the KL6821.

The dialogs offer not only functions for addressing DALI control gears and DALI control devices, but also for writing and reading parameters. The search function automatically detects the DALI device types and displays them in a tree structure.



All parameters of the KL6821 can also be set via the KS2000. This makes it possible, for example, to define the DALI commands that are sent when the digital inputs on the KL6821 are actuated.



Overview of the most important functions in the KS2000 for the KL6821:

- Find DALI devices
- Addressing the DALI devices, including adapting the short addresses
- Scene and group assignment of the DALI control gears
- Configuration of the DALI control gears, including the parameters for the different device types
- Configuration of the DALI control gears, including the parameters for the different sensor types
- Writing/reading of the memory banks
- Definition of DALI commands for DI1 and DI2 (separate for rising and falling edge)
- Definition of the DALI command for the K-bus watchdog (K-bus failure)
- Switching the internal DALI power supply unit on/off
- Activation/deactivation of the blocking of the process image for the PLC as soon as a DALI command is sent by actuating a digital input on the KL6821.

### 5.1.2 Commissioning EL6821

TwinCAT XAE provides dialogs for commissioning and diagnosis of DALI devices on the EL6821.

The dialogs offer not only functions for addressing DALI control gears and DALI control devices, but also for writing and reading parameters. The search function automatically detects the DALI device types and displays them in a tree structure.

Name	Value	Info
Actual Level	0	0,0 %
Max Level	254	100,0 %
Min Level	254	100,0 %
Fade Rate	44.7 steps/s	
Fade Time	Disabled	
Extended Fade Time Base	1	
Extended Fade Time Multiplier	Disabled	
Power On Level	254	100,0 %
System Failure Level	254	100,0 %
DTR0	255	MASK
DTR1	0	0,0 %
DTR2	0	0,0 %
Physical Minimum (read only)	254	100,0 %
Status (read only)	10000010	Power cycle seen; Lamp failure
Light Source Types (read only)	6	LED

Buttons: Write parameters, Read parameters

Furthermore, all parameters of the EL6821 can be set via TwinCAT XAE. This makes it possible, for example, to define the DALI commands that are sent when the digital inputs on the EL6821 are actuated.

DALI power supply

The behavior of the digital inputs DI1 and DI2 can be modified via the following table.

The digital inputs DI1 and DI2 (terminal points 2 and 6) have priority over automatic control through the PLC program. If the digital inputs are operated, the respective LED to the bus terminal is set. In addition, all DALI commands that are called by the PLC, are blocked.

Event	DALI command on event (to all DALI devies)
E-Bus watchdog	Do Nothing
DI1 rising edge	Off
DI1 falling edge	Do Nothing
DI2 rising edge	Recall Max Level
DI2 falling edge	Do Nothing

Do not lock the process image after DALI commands have been sent by DI1 or DI2.

Buttons: Default Values, Reset Inactive Process Image

[Product Database of the DALI Alliance...](#)  
[Product Information...](#)

Overview of the most important functions in the TwinCAT XAE for the EL6821:

- Find DALI devices
- Addressing the DALI devices, including adapting the short addresses
- Scene and group assignment of the DALI control gears
- Configuration of the DALI control gears, including the parameters for the different device types
- Configuration of the DALI control devices, including the parameters for the different sensor types
- Writing/reading of the memory banks
- Definition of DALI commands for DI1 and DI2 (separate for rising and falling edge)

- Definition of the DALI command for the E-bus watchdog (E-bus failure)
- Switching the DALI power supply unit on/off
- Activation/deactivation of the blocking of the process image for the PLC as soon as a DALI command is sent by actuating a digital input on the EL6821.

### 5.1.3 DALI PLC Commissioning Tool

The PLC project **DALI PLC Commissioning Tool** offers the user the possibility to configure DALI devices and up to ten KL6821/EL6821 with the aid of the TwinCAT PLC HMI (see [https://infosys.beckhoff.com/content/1033/TcPlcLib\\_Tc3\\_DALI/Resources/13219898251/.zip](https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc3_DALI/Resources/13219898251/.zip)).

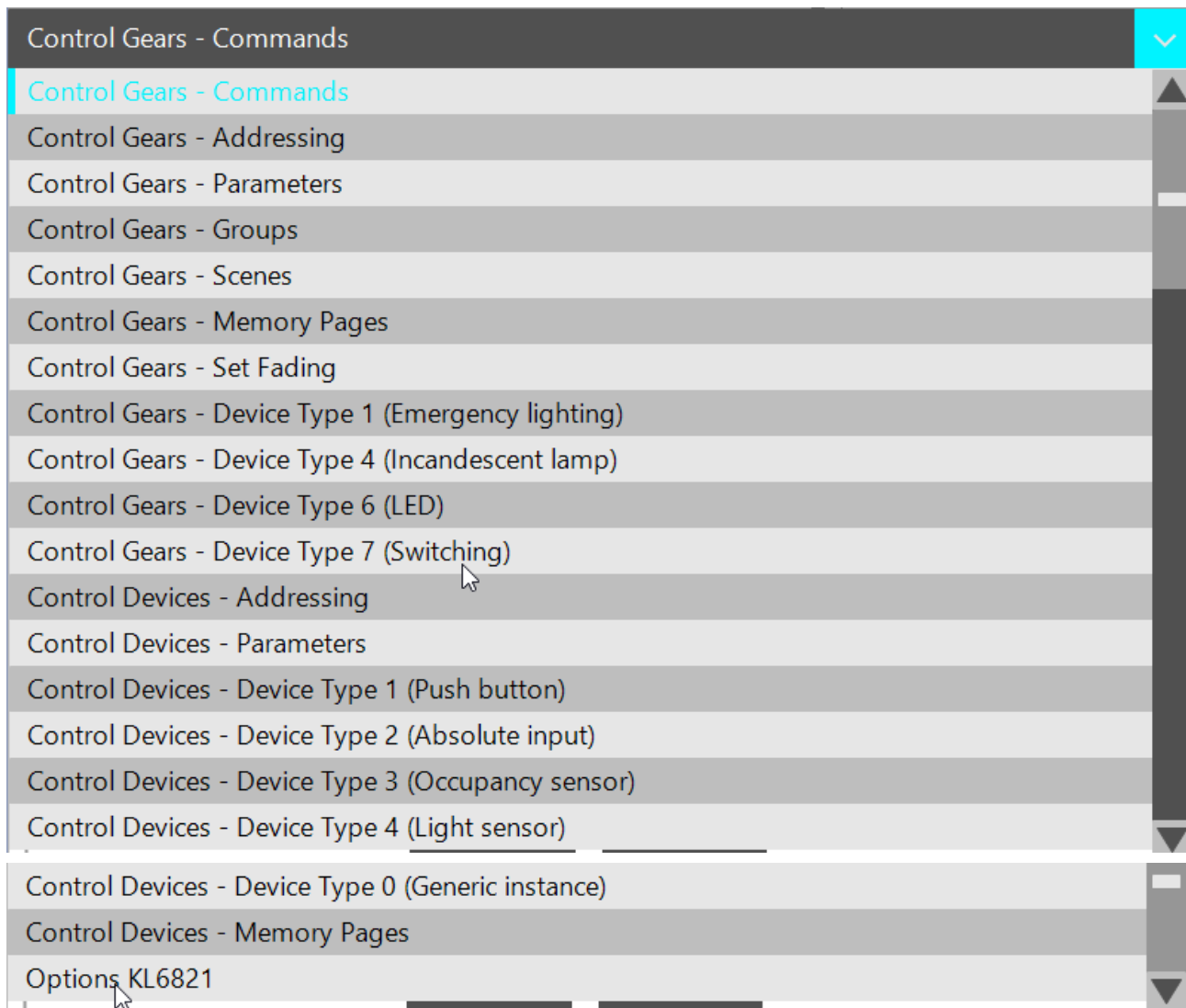


The TwinCAT project is available for download as \*.zip file. This must first be unpacked locally so that the archive (\*.tnzip file) is available for import into the TwinCAT project.

This project can be extended for specific applications, or it can only be used for configuration purposes. It initially supports ten KL6821/EL6821 via the global variable *nNumberOfDALILines*, but can be extended by the user to include more lines or the KL6811.

The dialogs offer not only functions for addressing DALI control gears and DALI control devices, but also for writing and reading parameters.

Using the drop-down menu, the user can select whether to parameterize DALI control gears or DALI control devices.



Addressing of the DALI devices and querying of the addresses that are already assigned takes place under the **Addressing** tab, which is available for both control gears and sensors.

Control Gears - Addressing
▼

**DALI Line:** 1

**Random Addressing**

Complete new installation  
(all addresses will be removed)

Optical feedback

**Start with address:** 0

Start

**Change Addresses**

OldAddress	NewAddress
0	0
3	3

Scan devices by address
Start

Parameters can be read based on selected short addresses and can be written, as far as possible.

Control Gears - Parameters

DALI Line: **1**

	ActualLevel	MinLevel	MaxLevel	FadeRate	FadeTime	PowerOnLevel	SystemFailureLevel	PhysicalMinimum	
0	0	126	254	7	0	0	50	3	
1	0	0	0	255	255	0	0	0	
2	0	0	0	255	255	0	0	0	
3	126	126	254	8	0	254	254	126	
4	0	0	0	255	255	0	0	0	
5	0	0	0	255	255	0	0	0	
6	0	0	0	255	255	0	0	0	
7	0	0	0	255	255	0	0	0	
8	0	0	0	255	255	0	0	0	
9	0	0	0	255	255	0	0	0	
10	0	0	0	255	255	0	0	0	
11	0	0	0	255	255	0	0	0	

**Single Short Address**

**Short Address Range**  **To**

There are also dialogs for

- Executing DALI commands
- Reading current states
- Reading and configuring group allocations
- Reading and configuring assignments within scenes
- Reading and writing the configuration of the following device types of DALI control gears:
  - 1 (Emergency lighting devices, Part 202) can be read out manually.
  - 4 (Regulation of supply voltage of incandescent lamps, Part 205) can be read out manually.
  - 6 (LED modules, part 207) can be read out manually.
  - 7 (Switching functions, Part 208) can be read out manually.
- Reading and writing the configuration of the following device types of DALI control devices:
  - 0 (Generic inputs, part 103) can be read out manually.
  - 1 (Push button, part 301) can be read out manually and event-driven.
  - 2 (Absolute value, part 302) can be read out manually.
  - 3 (Occupancy sensor, part 303) can be read out manually and event-driven.
  - 4 (Light sensor, part 304) can be read out manually.

For this purpose, one page is available for reading the configuration and one for writing.

- Adjustment of the fading settings
- Reading and, if the vendor wishes, configuration of the memory banks.
- Memory bank 1 for the control gears (part 102) contains the extension (device type 50).
- The device type 51 (energy report) is available by querying the memory banks 202, 203 and 204.



- The device type 52 (diagnostics and maintenance) is available by querying the memory banks 205, 206 and 207.
- Configuration of the KL6821/EL6821 (switching the power supply unit on/off, behavior on triggering of the K-bus watchdog, behavior in case of rising or falling edges on the two digital inputs of the terminal, enablement of the process image after the inputs were actuated, deactivation of the blocking of the process image when using the digital inputs of the KL6821/EL6821).

## Requirements

Required PLC library	DALI PLC Commissioning Tool
Tc3_DALI from v3.13.0.0	Tc3_DALI_PLC_Commissioning_Tool v3.8.0.0

### 5.1.4 Event Logger

The Tc3\_DALI library supports the TwinCAT Event Logger for the output of messages. This means that errors are displayed to the developer during runtime without having to explicitly extend the program for this purpose.

The output of the messages can be influenced by a variable. It is located in the Tc3\_DALI library as a global variable:

```
eEventTraceLevel : TcEventSeverity := TcEventSeverity.Critical;
```

Each message is assigned to a level that indicates how serious it is. The levels *Info*, *Warning*, *Error* or *Critical* are available. The global variable *eEventTraceLevel* defines the level from which a message is displayed in the message window.

By default, the variable is initialized so that only messages that correspond to the *Critical* level are output. Since most messages of the Tc3\_DALI library are assigned to the *Error* level, these messages are not displayed. The *Critical* level is rated more serious than the *Error* level.



The output of messages via the TwinCAT Event Logger may only be active for a limited time, e.g. during troubleshooting or commissioning.

## PLC

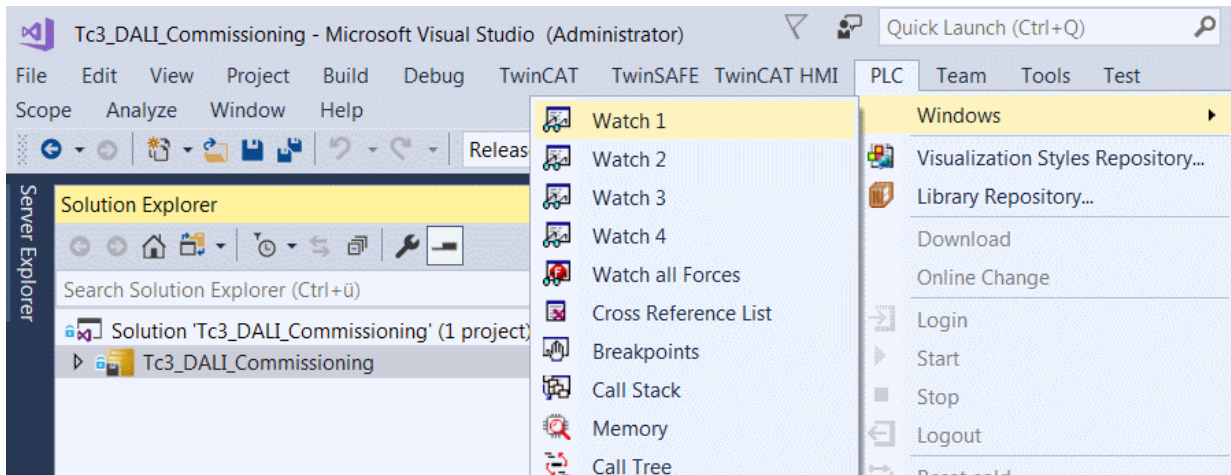
The following instruction, which you can insert into your PLC program, outputs all messages of all function blocks from the Tc3\_DALI library. This is particularly helpful during development or commissioning.

```
Tc3_DALI.GVL.eEventTraceLevel := TcEventSeverity.Verbose;
```

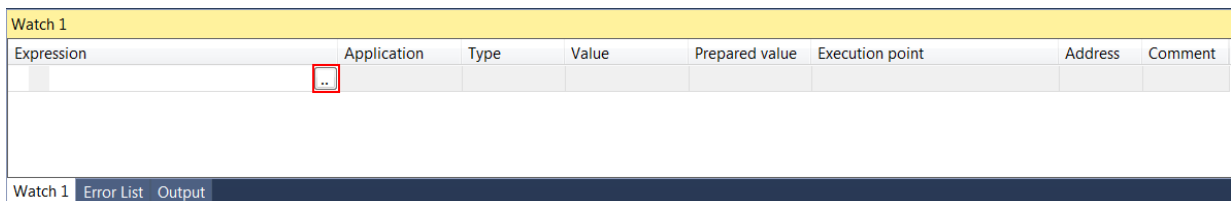
## Watch List

In addition to the extension of the PLC program, you can create a watch list in which the global variable *eEventTraceLevel* can be changed.

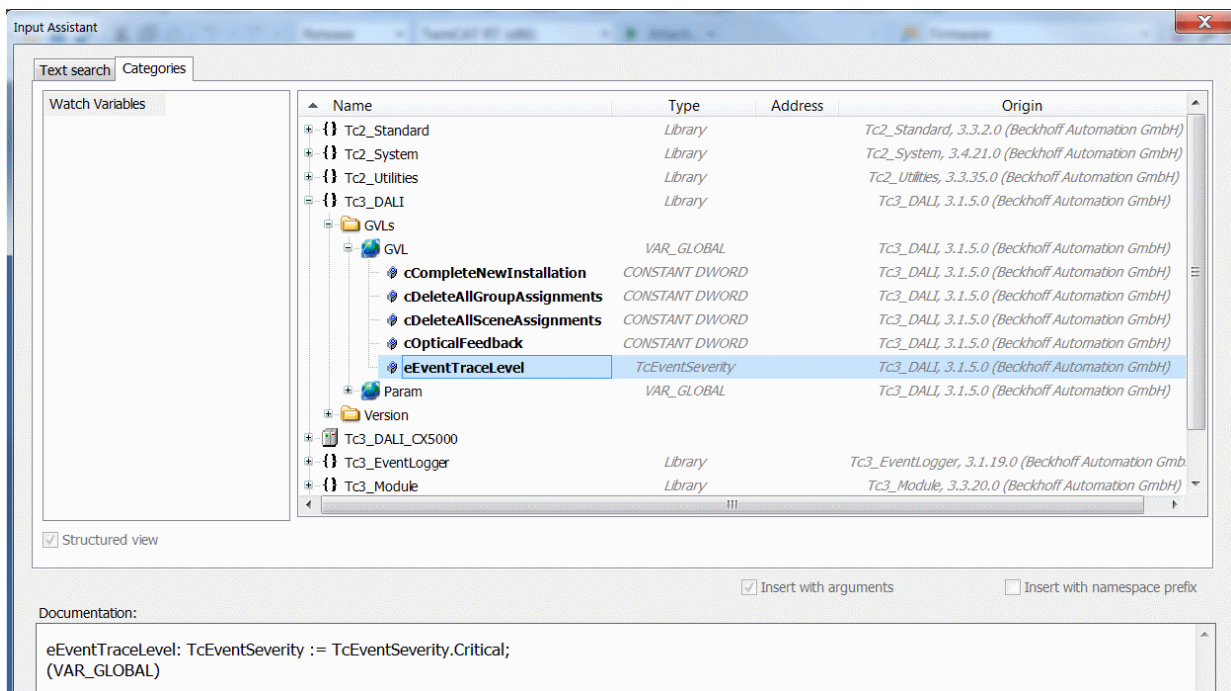
1. Create the watch list.



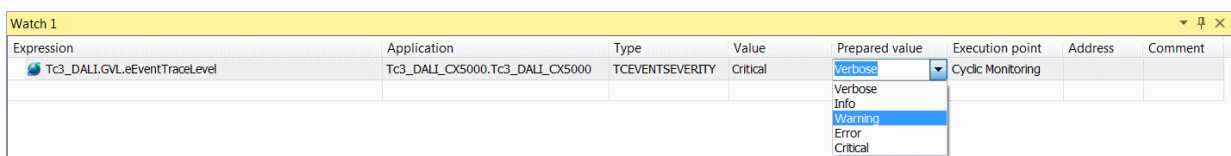
2. Select the global variable *eEventTraceLevel*.



The default setting of the variable can be seen in the **documentation** area of the input assistant.

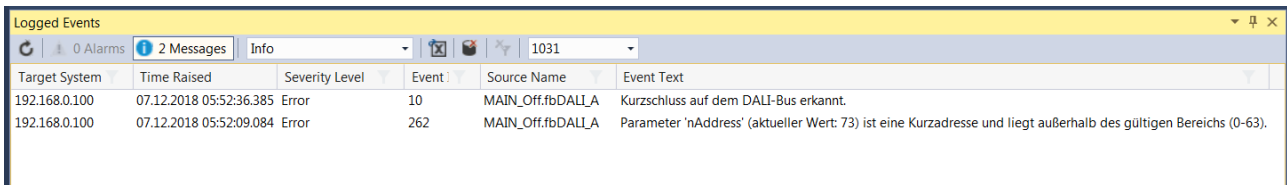


⇒ After adding the variable to the watch list you can change its value via a drop-down menu.

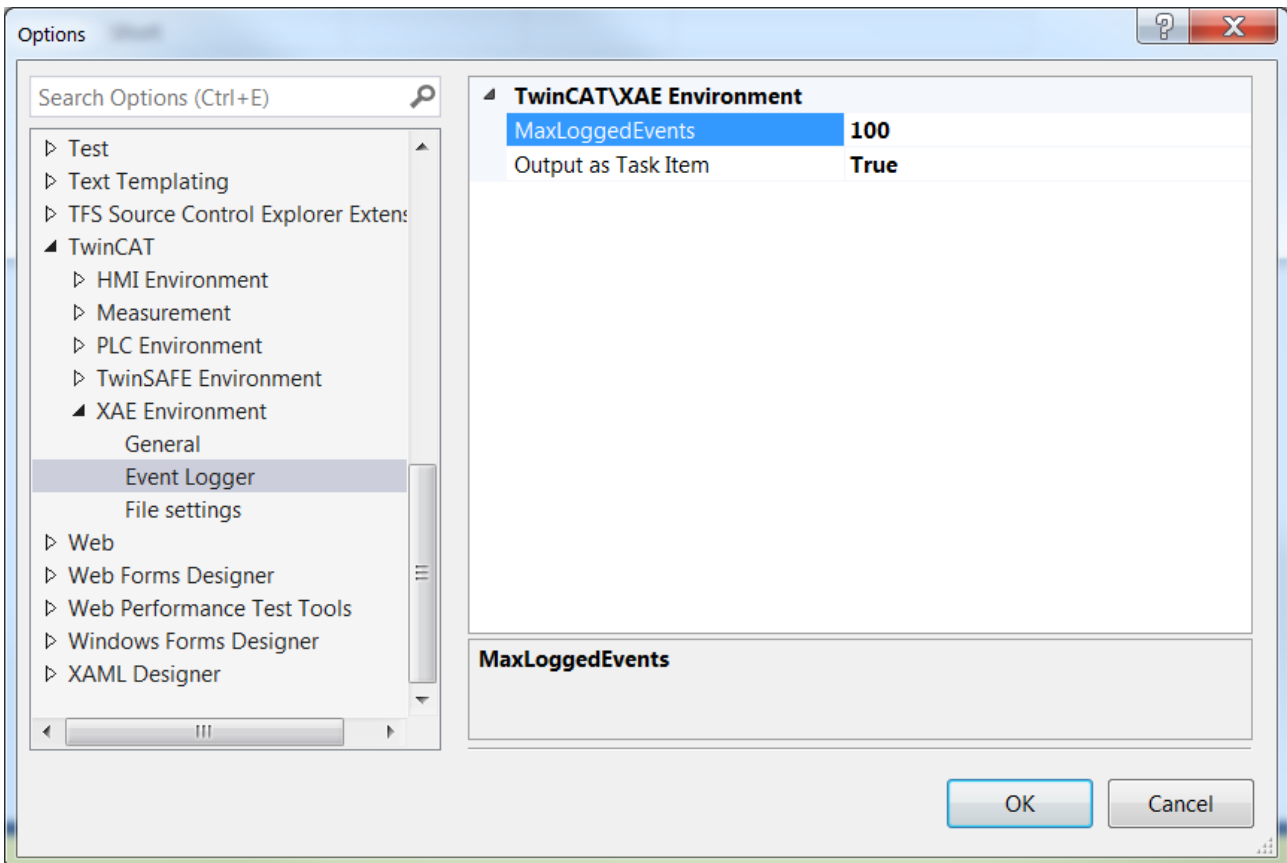


**Displaying the messages**

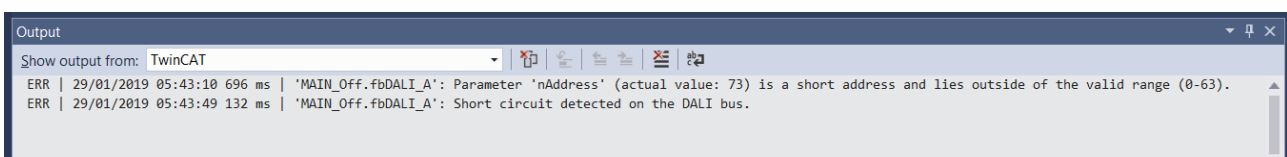
The messages are displayed in the TwinCAT **Logged Events** window (see chapter [Event Logger](#)).



All messages are stored in a ring buffer in the TwinCAT controller. The size of the ring buffer can be adapted in the TwinCAT options. This has the advantage that messages are saved even if the TwinCAT project is not open in Visual Studio.



In addition, the messages can also be displayed in the **Output** window.



Make sure that **Show output from** is set to **TwinCAT**.

### 5.1.5 Error evaluation

The interface pointer *ipResultMessage* of type *I\_TcMessage* provides information about a current event (runtime message [▶ 873]). The most important methods and properties are described below:

#### Methods

Name	Description
EqualsToEventEntryEx	Compares the event definition of the event with another event definition.
RequestEventText	Returns the text for the event.

## Properties

Name	Type	Description
eSeverity	TcEventSeverity	Returns the severity.
EventClass	GUID	Returns the unique ID of the event class.
nEventId	UDINT	Returns the ID of the event.
stEventEntry	TcEventEntry	Returns the event definition.

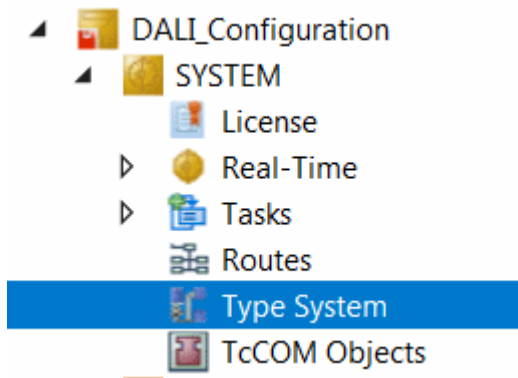
For details about the interface `I_TcMessage`, see the documentation for the `Tc3_EventLogger` library.

Each event is unambiguously described by the structure `TcEventEntry`. It contains three structure elements:

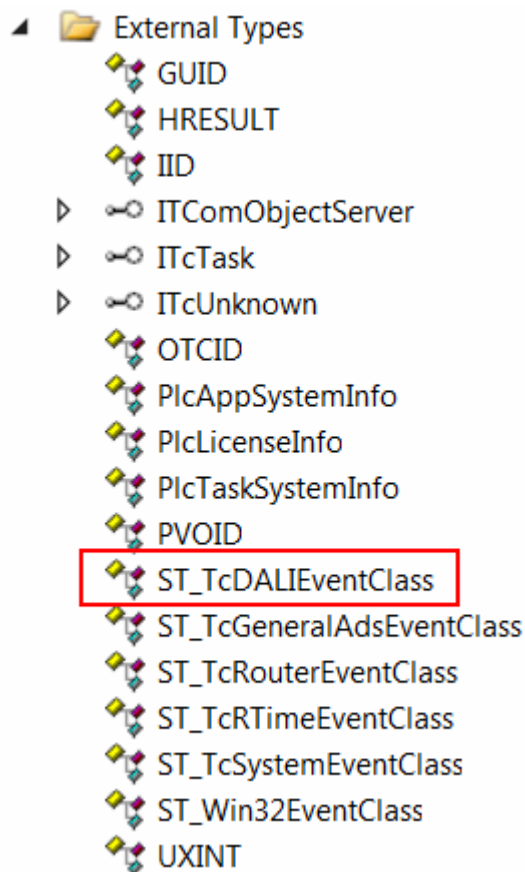
- Event-Class
- Event-ID
- Event-Severity

Several events are merged into one event class. For example, the event class for DALI and the event class for EnOcean can contain the event ID 100. The complete description of an event always consists of the event class and the event ID. This avoids ambiguities due to the same IDs within different libraries.

In order to make the events accessible to the PLC, TwinCAT automatically creates a structure for each event class in the type system.



This happens for all PLC libraries that are referenced in the project. The structure `ST_DALIEventClass` is added using the `Tc3_DALI` library.



This structure contains all events of the event class for the Tc3\_DALI library. Each element of the structure is mapped by a variable of type *TcEventEntry*. The name of the element indicates the meaning of the event.

```

TYPE ST_TcDALIEventClass:
  STRUCT
    NoResponseFromDALITerminal: TcEventEntry := ( ...
    NoResponseFromDALIDevice: TcEventEntry := ( ...
    CommandBufferOverflow: TcEventEntry := ( ...
    [...]
  END_STRUCT
END_TYPE

```

All event classes are stored in the global variable list *TC\_Events*, which is also generated automatically by TwinCAT.

```

VAR_GLOBAL CONSTANT
  [...]
  TcRTimeEventClass: ST_TcRTimeEventClass;
  Win32EventClass: ST_Win32EventClass;
  TcDALIEventClass: ST_TcDALIEventClass;
END_VAR

```

Thus all messages of all event classes are available in the PLC program.

## FAQ

### How can I query whether an FB returns an error?

In all libraries, the event ID for error-free execution is set to 0. Since the event ID not only outputs errors, but also notes and warnings, most function blocks have the output *bError*. This output is TRUE when an error occurs.

```

IF (fbDALI.bError) THEN
  ...
END_IF

```

This output remains FALSE in the event of warnings or notes. Therefore it may be the case that the event ID (*fbDALI.ipResultMessage.nEventID*) is not 0, but *bError* remains FALSE.

## How can I query whether a FB returns a specific event?

The method

```
ipResultMessage.EqualsToEventEntryEx(stOther TcEventEntry)
```

compares whether the event matches the transferred event *stOther*. Since a variable of type *TcEventEntry* automatically exists for each event, the query for a certain event can be carried out as follows.

```
IF (fbDALI.ipResultMessage.EqualsToEventEntryEx (
  TC_EVENTS.TcDALIEventClass.NoResponseFromTheDALIDevice)) THEN
...
END_IF
```

Alternatively, the Event ID can also be queried directly.

```
IF (fbDALI.ipResultMessage.nEventId = 2) THEN
...
END_IF
```

A structure of the data type *TcEventEntry* exists for each event. This is located inside the structure *TcDALIEventClass* (data type *ST\_TcDALIEventClass*). The meaning of the message is recognizable from the names of the individual events. The use of this structure can thus improve the readability of the PLC program:

```
IF (fbDALI.ipResultMessage.nEventId =
  TC_EVENTS.TcDALIEventClass.NoResponseFromTheDALIDevice.nEventId) then
...
END_IF
```

For a list of all events that represent an error, see [Runtime messages \[▶ 873\]](#).

## How can I query the text of an event?

The text that is output in the *TwinCAT Logged Events* window can also be queried in the PLC program using the method

```
ipResultMessage.RequestEventText(nLangId DINT, sResult REFERENCE TO STRING, nResultSize UDINT)
.
```

If an event is present, the following sample returns the event text:

```
sTxt      : STRING(255);
sEventText : STRING(255);
IF (fbDALI.ipResultMessage.RequestEventText(1031, sTxt, SIZEOF(sTxt))) THEN
  UTF8_TO_STRING(ADR(sEventText), ADR(sTxt), SIZEOF(sTxt));
END_IF
```

The first parameter specifies the language in which the text is to be read. The *Tc3\_DALI* library contains all texts in English (1033) and German (1031).

## 5.2 Use of properties

Properties have the advantage that they enable read and write access. Thus, parameters can be passed to a function block via properties, i.e. states of the function block can also be output.

The *Tc3\_DALI* uses properties mainly to pass parameters to function blocks. Each property has a documented initial value. Thus, it is not necessary to assign a value to each property.

### Assignment at runtime

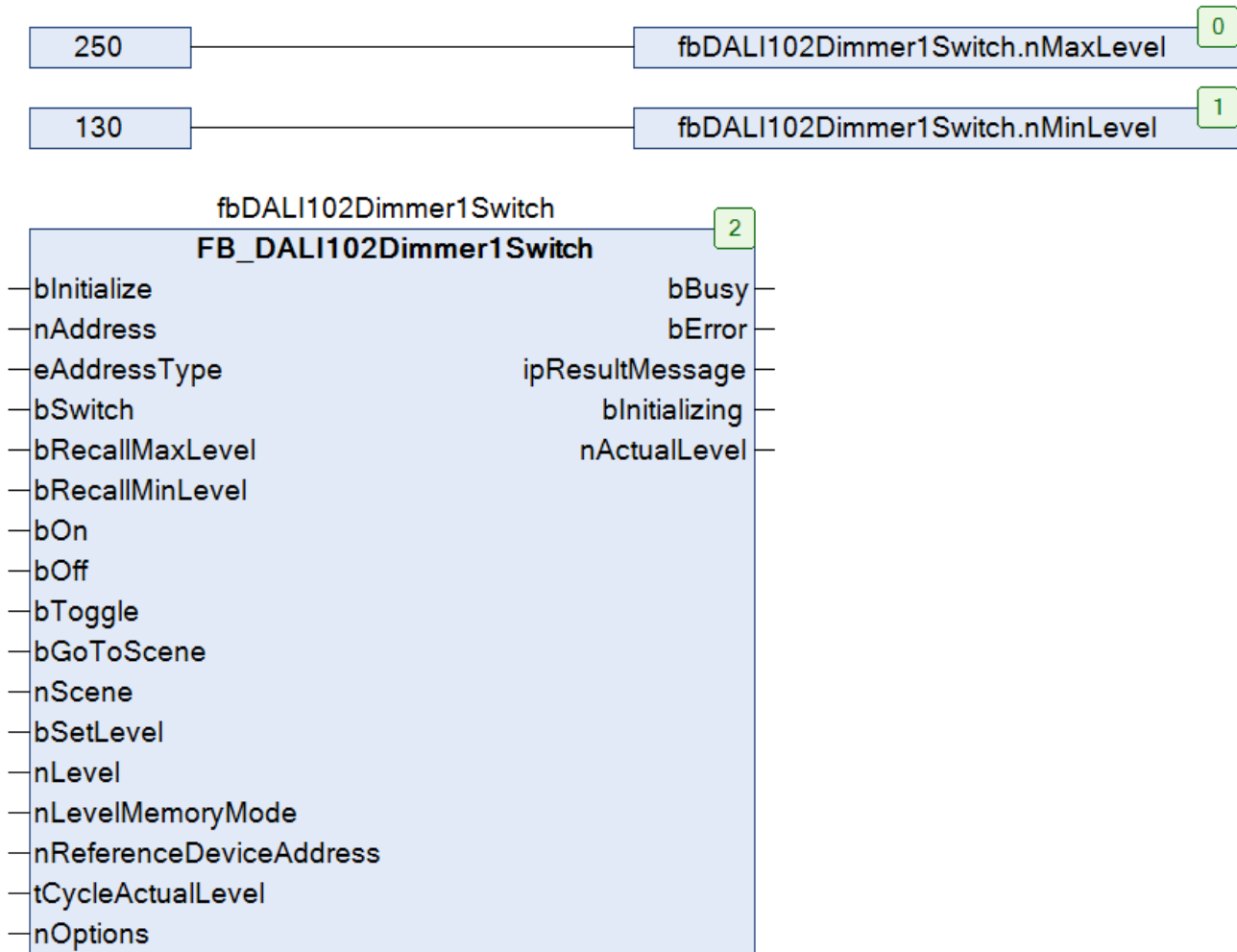
The following sample shows how values are passed to properties at runtime in ST and CFC.

```
fbDALI102Dimmer1Switch : FB_DALI102Dimmer1Switch(Communication.fbKL6821Communication);
```

In ST, the assignment of properties is done separately from the instance call.

```
fbDALI102Dimmer1Switch.nMaxLevel := 250;
fbDALI102Dimmer1Switch.nMinLevel := 130;
fbDALI102Dimmer1Switch();
```

The assignment of the properties is done in CFC by input and output elements and also independent of the instance call.



**Assignment during declaration**

Properties can also be given a value directly when a function block is declared.

```
fbDALI102Dimmer1Switch : FB_DALI102Dimmer1Switch(Communication.fbKL6821Communication) :=
    (nMaxLevel := 250, nMinLevel := 130);
```

The values are assigned to the properties before the first PLC cycle.

**Use of arrays**

If an array is declared by a function block, the properties for each element of the array can be given different values.

```
aDALI102Dimmer1Switch : ARRAY [1..3] OF
FB_DALI102Dimmer1Switch(Communication.fbKL6821Communication) :=
    [(nMaxLevel := 250, nMinLevel := 130), // Element 1
     (nMaxLevel := 240, nMinLevel := 130), // Element 2
     (nMaxLevel := 254, nMinLevel := 125)]; // Element 3
```

In the following sample, the two properties `nMaxLevel` and `nMinLevel` of element 1 and 2 are given individual values. The properties of element 3 remain unchanged and retain their initial values.

```
aDALI102Dimmer1Switch : ARRAY [1..3] OF
FB_DALI102Dimmer1Switch(Communication.fbKL6821Communication) :=
    [(nMaxLevel := 250, nMinLevel := 130), // Element 1
     (nMaxLevel := 240, nMinLevel := 130)]; // Element 2
```

The short form for multiple initialization can also be used.



```
aDALI102Dimmer1Switch : ARRAY [1..5] OF
FB_DALI102Dimmer1Switch(Communication.fbKL6821Communication) :=
    [2((nMaxLevel := 250, nMinLevel := 130)), // Element 1 and 2
     (nMaxLevel := 254, nMinLevel := 125), // Element 3
     2((nMaxLevel := 240, nMinLevel := 140))]; // Element 4 and 5
```

## 5.3 Transfer of the reference to the communication block

The individual function blocks of the Tc3\_DALI send and receive the DALI data packets via the respective DALI terminal (KL6811, KL6821 or EL6821). The function blocks access the DALI terminal via a special communication block ([FB\\_KL6811Communication](#) [► 583], [FB\\_KL6821Communication](#) [► 586] or [FB\\_EL6821Communication](#) [► 578]). Each instance of a communication block is assigned to exactly one DALI terminal.

The communication blocks ensure that the processing of the individual DALI commands is carried out correctly and that the received events are assigned to the correct function blocks.

Each function block requires a reference to a communication block. This assignment is implemented in the Tc3\_DALI with an interface pointer. There are two ways to pass the assignment of the communication block to a function block.

### Transfer during declaration

When declaring the function block, the complete path of the instance of the communication block ([FB\\_KL6811Communication](#), [FB\\_KL6821Communication](#) or [FB\\_EL6821Communication](#)) is specified in round brackets.

```
fbDALI102Off : FB_DALI102Off(Communication.fbKL6821Communication);
```

The complete path consists of the POU name in which the instance of the communication block was created ([Communication](#)) and the name of the instance ([fbKL6821Communication](#)), separated by a dot.

If no communication block is to be passed in the declaration, a zero reference can be passed by a '0'.

```
fbDALI102Off : FB_DALI102Off(0);
```

This can be useful if an assignment is to be made only at program runtime (see below).

If a module does not have a valid interface pointer to a communication block, a [runtime message](#) [► 873] is output.



If the interface pointer is changed in the declaration, this is not taken into account during an Online Change. In this case, the PLC program must be completely recompiled and loaded onto the target system.

### Use of arrays

Each instance can be assigned to an individual communication block. The instances of the communication blocks can also map different types of DALI terminals (KL6811, KL6821 or EL6821).

```
aDALI102Off : ARRAY[1..3] OF FB_DALI102Off[(Communication.fbKL6821Communication),
                                           (Communication.fbEL6821Communication),
                                           (Communication.fbKL6811Communication)];
```

In the following sample, all instances are assigned to the same communication block:

```
aDALI102Off : ARRAY[1..3] OF FB_DALI102Off(Communication.fbKL6821Communication);
```

### Transfer by property

So that the reference to the communication block can also be changed at runtime, all function blocks have the property *ipDALICommunication*. This property can also be used to pass the interface pointer.

```
fbDALI102Off.ipDALICommunication := Communication.fbKL6821Communication01;
```

The assignment does not have to take place in every PLC cycle. It is sufficient if this is done once.



```

IF (bSwitchToLinie01) THEN
    fbDALI1020ff.ipDALICommunication := Communication.fbKL6821Communication01;
    bSwitchToLinie01 := FALSE;
END_IF
IF (bSwitchToLinie02) THEN
    fbDALI1020ff.ipDALICommunication := Communication.fbKL6821Communication02;
    bSwitchToLinie02 := FALSE;
END_IF

```

For additional information on properties, see the chapter [Use op properties \[► 890\]](#).

## 5.4 Tc2\_DALI and Tc3\_DALI in the same project

The Tc2\_DALI library can be used for porting existing TwinCAT 2 projects to TwinCAT 3. At source code level, the TwinCAT 2 DALI library is compatible with the TwinCAT 3 Tc2\_DALI library, which significantly simplifies the transfer of TwinCAT 2 projects to TwinCAT 3.

In contrast, the TwinCAT 3 library Tc3\_DALI is a completely new development, which has been optimized for TwinCAT 3 and adapted to DALI-2. In this way it was possible to improve the performance and to simplify handling of the function blocks further.



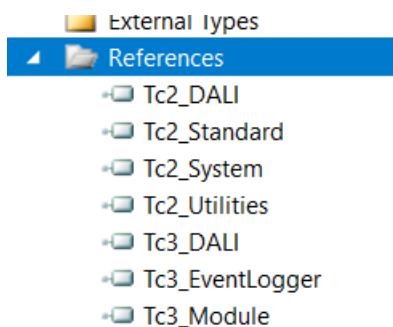
For new projects the Tc3\_DALI library should therefore be used if possible.

This can lead to a situation where the Tc2\_DALI library is used in a TwinCAT 3 project, this project is subsequently extended and the Tc3\_DALI library is to be used for this purpose.

The following section illustrates how a KL6821 with the Tc2\_DALI library and a further KL6821 with the Tc3\_DALI library are operated in a TwinCAT 3 project.

The function blocks of the Tc2\_DALI and Tc3\_DALI libraries cannot be mixed on the same Bus Terminal. Each library must have exclusive access to a KL6821 or KL6811.

Add both libraries to the TwinCAT 3 project:



The necessary variables must be created for each terminal and linked with the KL6821. In this example, the global variable list has the name GVL.

```

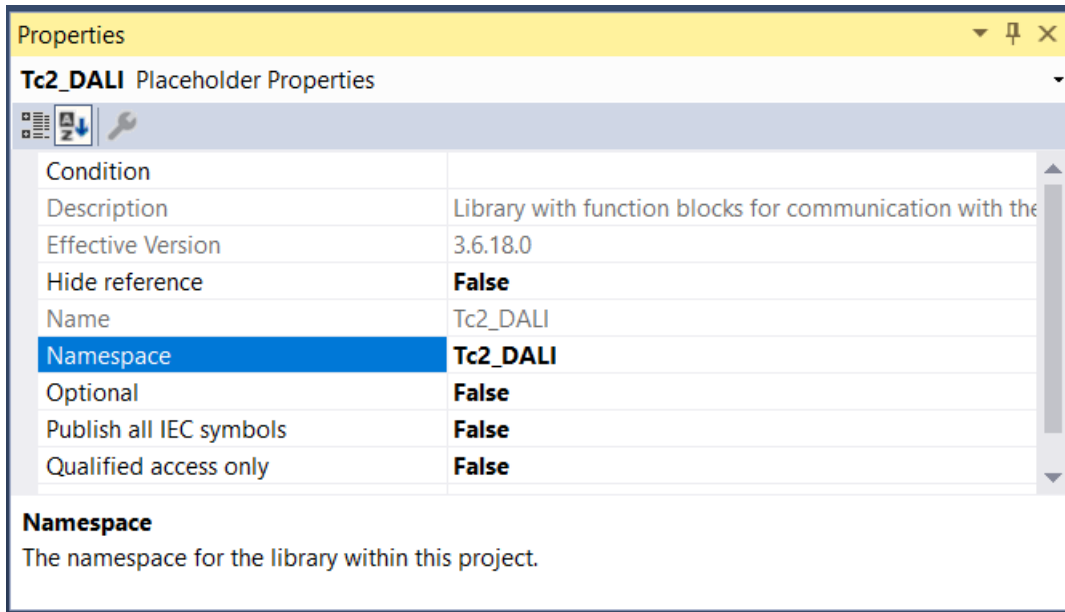
VAR_GLOBAL
    // Tc2_DALI
    stKL6821InDataTc2 AT %I* : Tc2_DALI.ST_KL6821InData;
    stKL6821OutDataTc2 AT %Q* : Tc2_DALI.ST_KL6821OutData;
    stCommandBufferKL6821 : Tc2_DALI.ST_DALIV2CommandBuffer;

    // Tc3_DALI
    stKL6821InDataTc3 AT %I* : Tc3_DALI.ST_KL6821InData;
    stKL6821OutDataTc3 AT %Q* : Tc3_DALI.ST_KL6821OutData;
END_VAR

```

Since the structures for linking the Bus Terminal have the same name in both libraries, the namespace must be prefixed. Although the structures have the same name in both libraries, they have different data types within the TwinCAT 3 type system. The namespace ensures that the appropriate data type from the respective library is used. For better readability, the namespace was also specified for the structure for the command buffer (*ST\_DALIV2CommandBuffer*) from the Tc2\_DALI library. Strictly speaking this would not be necessary at this point, because the identifier *ST\_DALIV2CommandBuffer* only occurs in the Tc2\_DALI library.

The namespace of a library is displayed in the **Properties** window.



By convention, the namespace of the Beckhoff library is the same as the name of the library.

Next, the function block that is called by the communication task is created. In this function block, an instance of FB\_KL6821Communication is called for each terminal, among other things. The namespace must also be specified here, since each library contains a function block with the name FB\_KL6821Communication.

```
PROGRAM Communication
VAR
  // Tc2_DALI
  fbKL6821CommunicationTc2 : Tc2_DALI.FB_KL6821Communication;

  // Tc3_DALI
  fbKL6821CommunicationTc3 : Tc3_DALI.FB_KL6821Communication;
END_VAR

// Tc2_DALI
fbKL6821CommunicationTc2(stInData := GVL.stKL6821InDataTc2,
stOutData := GVL.stKL6821OutDataTc2,
stCommandBuffer := GVL.stCommandBufferKL6821);

// Tc3_DALI
fbKL6821CommunicationTc3(stInData := GVL.stKL6821InDataTc3,
stOutData := GVL.stKL6821OutDataTc3);
```

When calling the respective function blocks for sending the DALI commands or for receiving the events, in most cases it is not necessary to specify the namespace. In the Tc2\_DALI library all elements have the prefix *DALIV2* while in the Tc3\_DALI the prefix *DALI* is used.

Here you can download the sample project: [https://infosys.beckhoff.com/content/1033/TcPlcLib\\_Tc3\\_DALI/Resources/9843858059/.zip](https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc3_DALI/Resources/9843858059/.zip)

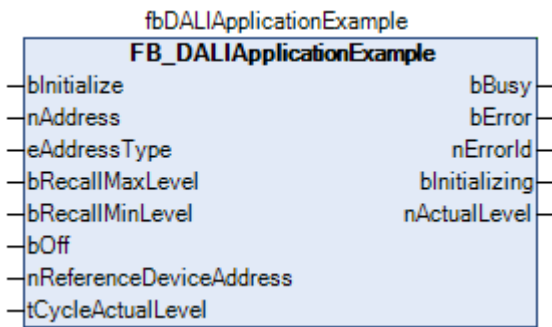
## 5.5 Creating an application function block

The following section illustrates how to create an application function block based on the Tc3\_DALI library. The example describes a simple function block that controls control gears via short address, group address or broadcast.

Here you can find the [https://infosys.beckhoff.com/content/1033/TcPlcLib\\_Tc3\\_DALI/Resources/9869845131/.zip](https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc3_DALI/Resources/9869845131/.zip).



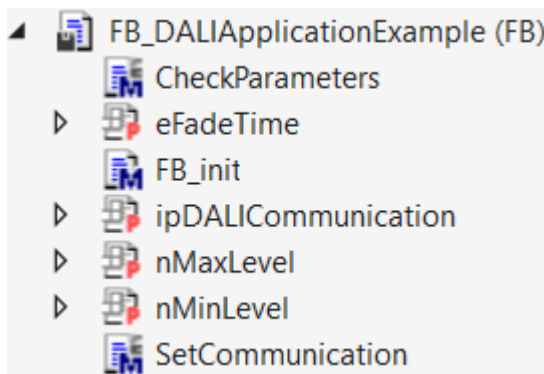
The TwinCAT project is available for download as \*.zip file. This must first be unpacked locally so that the archive (\*.tnzip file) is available for import into the TwinCAT project.



A positive edge at one of the three inputs sends the DALI command RECALL MAX LEVEL, RECALL MIN LEVEL or OFF.

**Parameterization of the DALI control gears**

The function block can initialize the variables *maxLevel*, *minLevel* and *fadeTime* of DALI control gears. Corresponding properties are available for this purpose. Their values are written to the DALI control gears when a positive edge is detected at *bInitialize*. During this time the output *bInitializing* is TRUE.



**DALI short address for reference device (*nReferenceDeviceAddress*)**

The parameter *nReferenceDeviceAddress* is used for reading the output value of a reference DALI control gear cyclically in the background (*tCycleActualLevel* > 0 s). The output value is available at the output *nActualLevel*.

No reference DALI control gear is required if a single DALI control gear is to be controlled with the function block (*eAddressType* = *E\_DALIAddressType.Short*). In this case, the output value of the individual control gear is determined via the individual address *nAddress*.

If a DALI group is addressed with the function block (*eAddressType* = *E\_DALIAddressType.Group*), *nReferenceDeviceAddress* must always contain the short address of a DALI control gear, which must be located in the addressed DALI group.

Even with a broadcast (*eAddressType* = *E\_DALIAddressType.Broadcast*), a DALI control gear with a corresponding short address must be located on the DALI line.

**Transfer of the reference to the communication block**

When declaring the function block, the complete path of the communication block ([FB\\_KL6811Communication](#) [[▶ 583](#)], [FB\\_KL6821Communication](#) [[▶ 586](#)] or [FB\\_EL6821Communication](#) [[▶ 578](#)],...) is specified in round brackets.

```
fbDALIApplicationExample : FB_DALIApplicationExample(Communication.fbKL6821Communication);
```

This is used to specify via which DALI terminal the function block is to communicate.

In *FB\_DALIApplicationExample* the parameter is passed by the method *FB\_init*. The method *FB\_init* is called automatically by the TwinCAT 3 runtime environment, once before the PLC program is started.

The parameter is of type *I\_DALICommunication* and is contained in the Tc3\_DALI library. All communication blocks (FB\_KL6811Communication [► 583], FB\_KL6821Communication [► 586] or FB\_EL6821Communication [► 578], ...) have implemented this interface. All DALI command function blocks (e.g. FB\_DALI102Off) communicate with the DALI communication block via the interface pointer.

In the method *FB\_init* the parameter is assigned to the variable *\_ipDALICommunication* of the function block.

To ensure that the reference can be changed at runtime, the interface pointer is also passed to the device function block via the property *ipDALICommunication*. Internally, the property is stored in the variable *\_ipDALICommunication*.

In the device function block it must be ensured that this interface pointer is passed on to all DALI command function blocks. The internal method *SetCommunication* is available for this purpose. In this method the interface pointer is passed to all DALI command function blocks located in the function block. The property *ipDALICommunication* is used in the command function blocks for this purpose. This property is used to reassign the reference to the DALI communication block at runtime.

### Checking the parameters

All parameters are checked by the internal method *CheckParameters*. In the event of an error the method returns the error Id, or 0 if there is no error. The Ids of the [runtime messages \[► 873\]](#) can be used here:

```
IF (THIS^.ipDALICommunication = 0) THEN
  CheckParameters := TC_EVENTS.TcDALIEventClass.TheInterfaceToTheCommunicationBufferIsNotInitialized
  .nEventId;
END_IF
```

If required, you can also use your own error Ids, starting at 1000:

```
IF (<check your parameter>) THEN
  CheckParameters := 1000;
END_IF
```

Similar to *SetCommunication*, *CheckParameters* is also called in the first PLC cycle. This call also occurs if a positive edge was detected at the inputs *bInitialize*, *bRecallMaxLevel*, *bRecallMinLevel* or *bOff*.

Alternatively, you could call the method every time the value of a parameter has changed.

### Structure of the function block

In the upper part of the function block the positive edges of the inputs *bInitialize*, *bRecallMaxLevel*, *bRecallMinLevel* or *bOff* are queried.

The middle part contains a step sequence in which the individual actions are carried out. If a positive edge is detected at one of the inputs, the corresponding step of the step sequence is triggered. The DALI command QUERY ACTUAL LEVEL is sent once the DALI commands RECALL MAX LEVEL (step 200) or RECALL MIN LEVEL (step 300) have been executed. This reads the new output value from the DALI reference device and outputs it at the output *nActualLevel*. The variables *maxLevel*, *minLevel* and *fadeTime* are written from step 10000.

In the lower part, the output value is read cyclically from the DALI reference device and output at the output *nActualLevel*.

### Comments

The step sequence can only be set to a new action if no action is currently being executed (*nStep* = 0).

The number of DALI commands should be minimized. For example, after the DALI commands RECALL MAX LEVEL or RECALL MIN LEVEL, the current output value is only read out if during the next 1000 ms the output value is not already read out by cyclic querying in the lower part of the function block:

```
IF ((tonCycleActualLevel.PT - tonCycleActualLevel.ET) > T#1S) OR (tCycleActualLevel = T#0S) THEN
  nStep := 9000;
ELSE
  bBusy := FALSE;
  nStep := 0;
END_IF
```

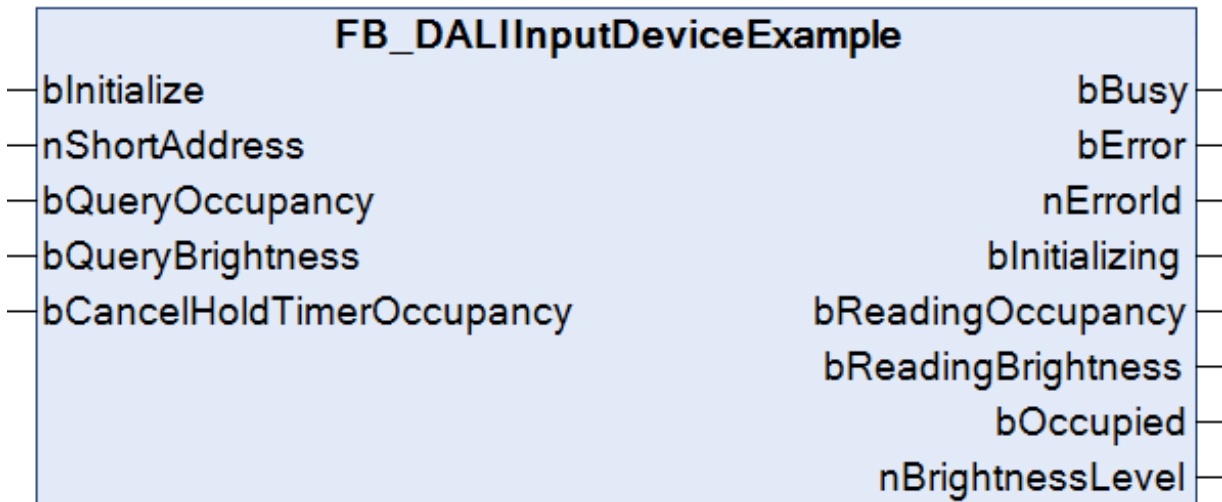
## 5.6 Creating a device function block

The following section illustrates how to create a device function block based on the Tc3\_DALI library. The sample describes a simple function block that maps a DALI sensor (input device) with occupancy sensor (instance 0) and light sensor (instance 1).

Here you can find the [https://infosys.beckhoff.com/content/1033/TcPlcLib\\_Tc3\\_DALI/Resources/11326365707/.zip](https://infosys.beckhoff.com/content/1033/TcPlcLib_Tc3_DALI/Resources/11326365707/.zip).



The TwinCAT project is available for download as \*.zip file. This must first be unpacked locally so that the archive (\*.tnzip file) is available for import into the TwinCAT project.

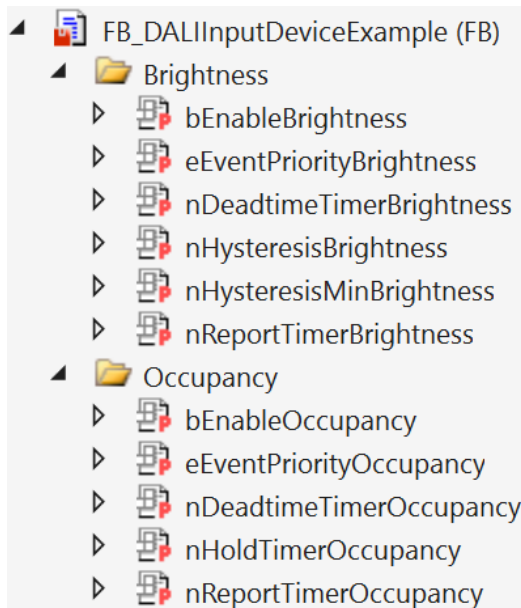


The current measured value is read directly from the respective instance via the inputs *bQueryOccupancy* and *bQueryBrightness* and output at the output *bOccupied* or *nBrightnessLevel*. During the reading process, the output *bReadingOccupancy* or *bReadingBrightness* = TRUE.

A positive edge at input *bCancelHoldTimerOccupancy* causes the DALI command CANCEL HOLD TIMER to be sent to instance 0 (occupancy sensor). This terminates the hold timer prematurely and, if there is no motion, sets the output *bOccupied* to FALSE.

### Parameterization of the DALI control gears

The function block can initialize all relevant variables from the occupancy sensor (instance 0) and from the light sensor (instance 1). Corresponding properties are available for this purpose. Their values are written to the DALI control device when a positive edge is detected at *bInitialize*. During this time the output *bInitializing* is TRUE.



### Transfer of the reference to the communication block

When declaring the function block, the complete path of the communication block (FB\_KL6811Communication [► 583], FB\_KL6821Communication [► 586] or FB\_EL6821Communication [► 578],...) is specified in round brackets.

```
fbDALIApplicationExample : FB_DALIApplicationExample(Communication.fbKL6821Communication);
```

This is used to specify via which DALI terminal the function block is to communicate.

In FB\_DALIApplicationExample the parameter is passed by the method *FB\_init*. The method *FB\_init* is called automatically by the TwinCAT 3 runtime environment, once before the PLC program is started.

The parameter is of type *I\_DALICommunication* and is contained in the Tc3\_DALI library. All communication blocks (FB\_KL6811Communication [► 583], FB\_KL6821Communication [► 586] or FB\_EL6821Communication [► 578], ...) have implemented this interface. All DALI command function blocks (e.g. FB\_DALI102Off) communicate with the DALI communication block via the interface pointer.

In the method *FB\_init* the parameter is assigned to the variable *\_ipDALICommunication* of the function block.

To ensure that the reference can be changed at runtime, the interface pointer is also passed to the device function block via the property *ipDALICommunication*. Internally, the property is stored in the variable *\_ipDALICommunication*.

In the device function block it must be ensured that this interface pointer is passed on to all DALI command function blocks. The internal method *SetCommunication* is available for this purpose. In this method the interface pointer is passed to all DALI command function blocks located in the function block. The property *ipDALICommunication* is used in the command function blocks for this purpose. This property is used to reassign the reference to the DALI communication block at runtime.

### Checking the parameters

All parameters are checked by the internal method *CheckParameters*. In the event of an error the method returns the error Id, or 0 if there is no error. The Ids of the [runtime messages \[► 873\]](#) can be used here:

```
IF (THIS^.ipDALICommunication = 0) THEN
  CheckParameters :=
  TC_EVENTS.TcDALEventClass.TheInterfaceToTheCommunicationBufferIsNotInitialized.nEventId;
END_IF
```

If required, you can also use your own error Ids, starting at 1000:

```
IF (<check your parameter>) THEN
  CheckParameters := 1000;
END_IF
```

*CheckParameters* is called in the first PLC cycle and on a positive edge at input *bInitialize*.

In addition, the method *CheckParameters* is called as soon as a parameter has changed. The check for parameter change is done in the method *ParameterChanged*. If this returns TRUE, a parameter has been changed. Then the method *CheckParameters* is used to check whether all parameters contain correct values. If this is not the case, the function block is terminated.

```
IF (THIS^.ParameterChanged()) THEN
  nErrorId := THIS^.CheckParameters();
  bError := (nErrorId > 0);
  IF (bError) THEN
    bParameterInvalid := TRUE;
    RETURN;
  ELSE
    bParameterInvalid := FALSE;
  END_IF
END_IF
IF (bParameterInvalid) THEN
  RETURN;
END_IF
```

### Structure of the function block

In the upper part of the function block, the parameters (*CheckParameters*) are checked when the function block is started for the first time and the interface pointer from the communication block (*\_ipDALICommunication*) is transferred to the internal DALI function blocks (*SetCommunication*).

In the middle part there is a step sequence with which the DALI control device is initialized. The function blocks FB\_DALI103ControlDevice, FB\_DALI303OccupancySensor and FB\_DALI304LightSensor are used for this purpose. At the end of the initialization, the current measured value is read from both instances. Initialization may only be started if no measured value is currently being read out (*bReadingOccupancy* and *bReadingBrightness* are FALSE).

In the lower part the notifications are evaluated and the reading of the measured values is executed by the inputs *bQueryOccupancy* and *bQueryBrightness*. The function blocks FB\_DALI103ControlDevice, FB\_DALI303OccupancySensor and FB\_DALI304LightSensor are also used for this purpose. This area is only processed if initialization is not active.





More Information:  
[www.beckhoff.com/te1000](http://www.beckhoff.com/te1000)

Beckhoff Automation GmbH & Co. KG  
Hülshorstweg 20  
33415 Verl  
Germany  
Phone: +49 5246 9630  
[info@beckhoff.com](mailto:info@beckhoff.com)  
[www.beckhoff.com](http://www.beckhoff.com)

