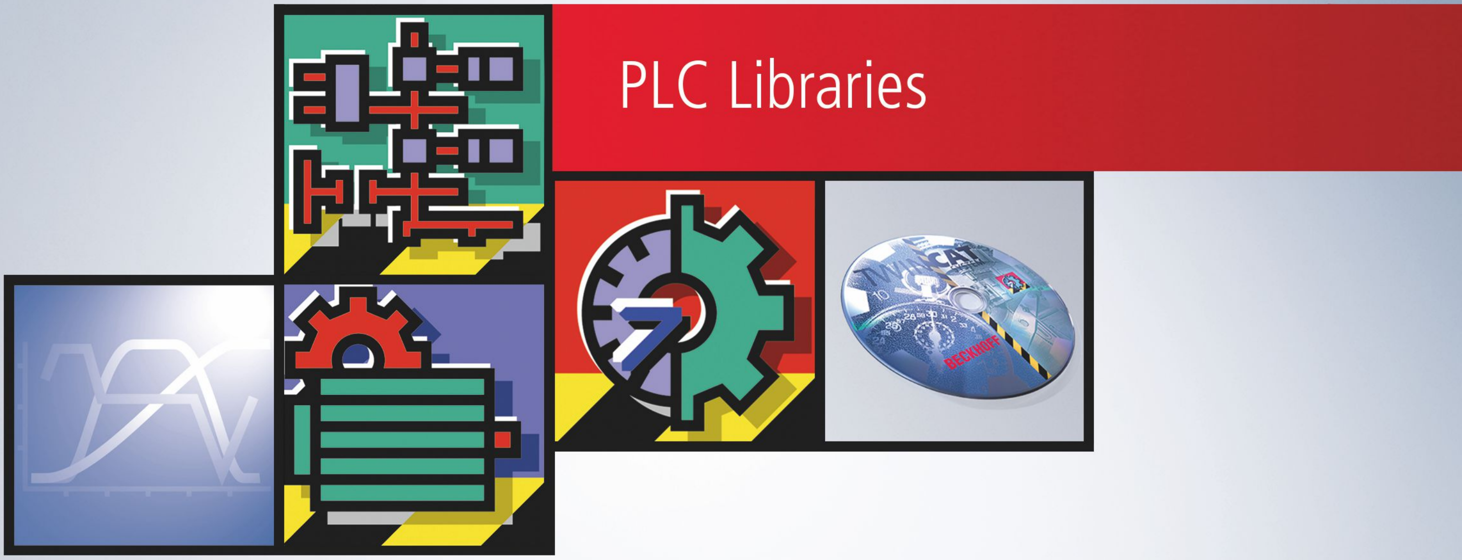
BECKHOFF New Automation Technology

Manual | EN

TX1200

TwinCAT 2 | PLC Library: TcDALIV2



2022-11-30 | Version: 1.1

Table of contents

1 Foreword

1.1 Notes on the documentation

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning the components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

The responsible staff must ensure that the application or use of the products described satisfy all the 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 prior announcement.
No claims for the modification of 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 trademarks of and licensed by Beckhoff Automation GmbH.
Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
with corresponding applications or registrations in various other countries.

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

Copyright

1.2 Safety instructions

Safety regulations

Please note the following safety instructions and explanations!
Product-specific safety instructions can be found on following pages or in the areas mounting, wiring, commissioning etc.

Exclusion of liability

All the components are supplied in particular hardware and software configurations 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 engineering who are familiar with the applicable national standards.

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.

NOTE

Damage to the environment or devices

Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.

ℹ **Tip or pointer**

This symbol indicates information that contributes to better understanding.

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 Target groups

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

- TwinCAT PLC Control
- TwinCAT System Manager
- PC and network knowledge
- Structure and properties of the Beckhoff Embedded PC and its Bus Terminal system
- Technology of DALI devices
- 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.

3 DALI

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

The KL6811 (DALI/DSI master) and KL6821 (DALI2 master) Bus Terminals are integrated into the Bus Terminal system as normal Bus Terminals 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.

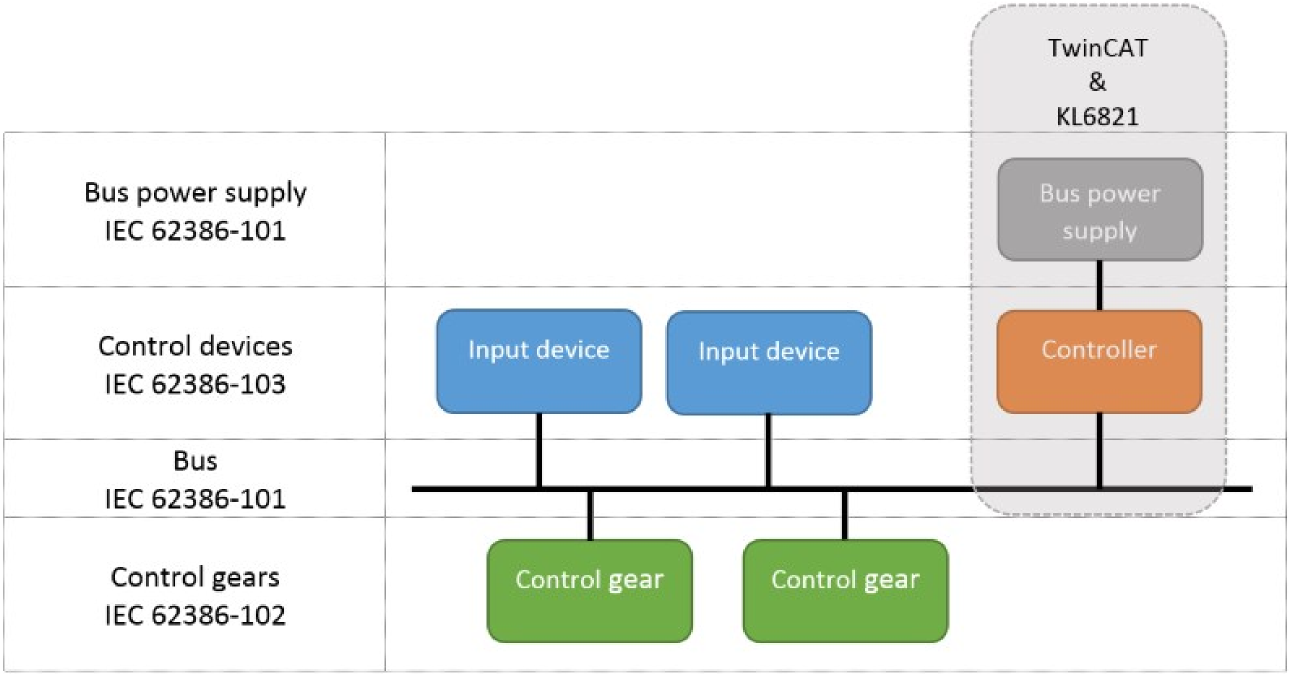
IEC 62386

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 ballasts (lamps) were considered, from the second revision onwards control units (sensors) are also included. These are described in the respective section of IEC 62386:

IEC 62386-101	general system properties such as cabling, feed-in and telegram structure.
IEC 62386-102	General properties of the ballasts (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 62386-103	General properties of the control units (control devices).
	IEC 62386-301: Push buttons 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.



In each DALI line up to 64 control gears and up to 64 input devices can be connected. The KL6821 represents the DALI controller. One such device exists for each DALI line. Any number of DALI lines (KL6821) can be operated with a single TwinCAT controller.

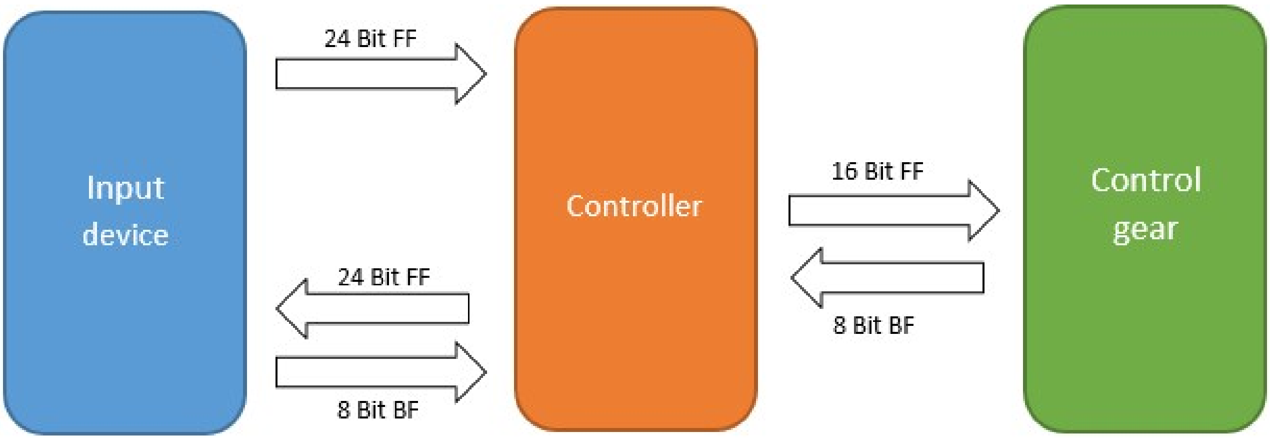
Communication

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

- 16-bit query, configuration and control telegram.

- 24-bit query, configuration and control telegram.

- 24-bit event telegram.



BF: backward frame
FF: forward frame

16-bit telegrams

16-bit telegrams 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 response. DALI control gears only send an 8-bit telegram when requested.

In the DALI library these commands are provided in the form of PLC function blocks with the prefix *FB_DALIV2*, e.g. FB_DALIV2QueryActualLevel().

24-bit telegrams

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

In the DALI library these commands are provided in the form of PLC function blocks with the prefix *FB_DALIV2x*, e.g. FB_DALIV2xQueryOperatingMode().

24-bit events

DALI input devices can send events. They are always evaluated by the DALI controller and have a length of 24 bits.

Individual events can be filtered out with the function block FB_DALIV2xGetEventData() [▶ 94] for further processing.

Further information on DALI can be found on the website of the DALI Activity Group (http://www.dali-ag.org) or the Digital Illumination Interface Alliance (https://www.digitalilluminationinterface.org) and in the IEC 62386 standard.

The present version of the DALI library replaces the previous version of the DALI library. You will find the older DALI library, with instructions, under https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019285387/.zip. You should only use the current version for newer projects, as the old version will no longer be further developed.

The KL6811 only supports the first revision of the DALI standard. It is not possible to operate control units (sensors) with the KL6811. The KL6821 is backward compatible with the KL6811, but it does not support DSI.

4 Integration into TwinCAT

4.1 KL6811 Integration into TwinCAT (CX9020)

This example explains how to write a simple PLC program for DALI in TwinCAT and how to link it with the hardware. The task is to control an individual dimmable lamp via a button.

https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019286795/.zip https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019286795/.zip

Hardware

Setting up the components

The following hardware is required:

- 1x Embedded PC CX9020
- 1x digital 2-channel input terminal KL1002 (for the dimming and reset functions)
- 1 DALI terminal KL6811
- 1x end terminal KL9010

Set up the hardware and the DALI components as described in the associated documentation.

This example assumes that a Dim button was connected to the first KL1002 input and a Reset button to the second, and that a dimmable lamp is connected to DALI address 0. Set the initial fade rate of the ballast to 7 to achieve suitable dimming.

Software

Creation of the PLC program

Create a new PLC project for PC-based systems (ARM) and add the *TcDALIV2.lib* library.

Next, generate the following global variables:

```
VAR_GLOBAL
    bSwitch       AT %I*      : BOOL;
    bReset        AT %I*      : BOOL;
    stDALIInData      AT %I* : ST_DALIV2InData;
    stDALIOutData     AT %Q* : ST_DALIV2OutData;
    stCommandBuffer          : ST_DALIV2CommandBuffer;
END_VAR
```

bSwitch: Input variable for the Dim button.

bReset: Input variable for the Reset button.

stDALIInData: Input variable [▸ 384] for the DALI terminal.

stDALIOutData: Output variable [▸ 384] for the DALI terminal.

stCommandBuffer : Required for the communication with DALI.

Then create a program (CFC) for background communication with DALI. The FB_DALIV2Communication() [▸ 93] block is called in this program. Make sure to link the communication block with *stDALIInData*, *stDALIOutData* and *stCommandBuffer*.

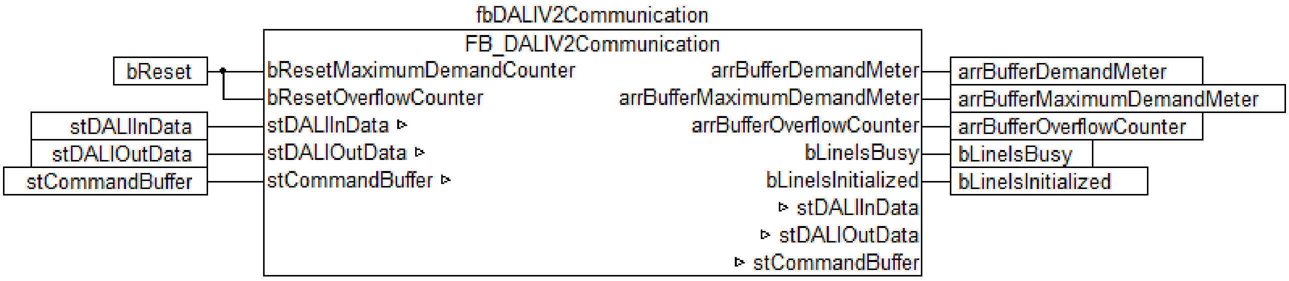
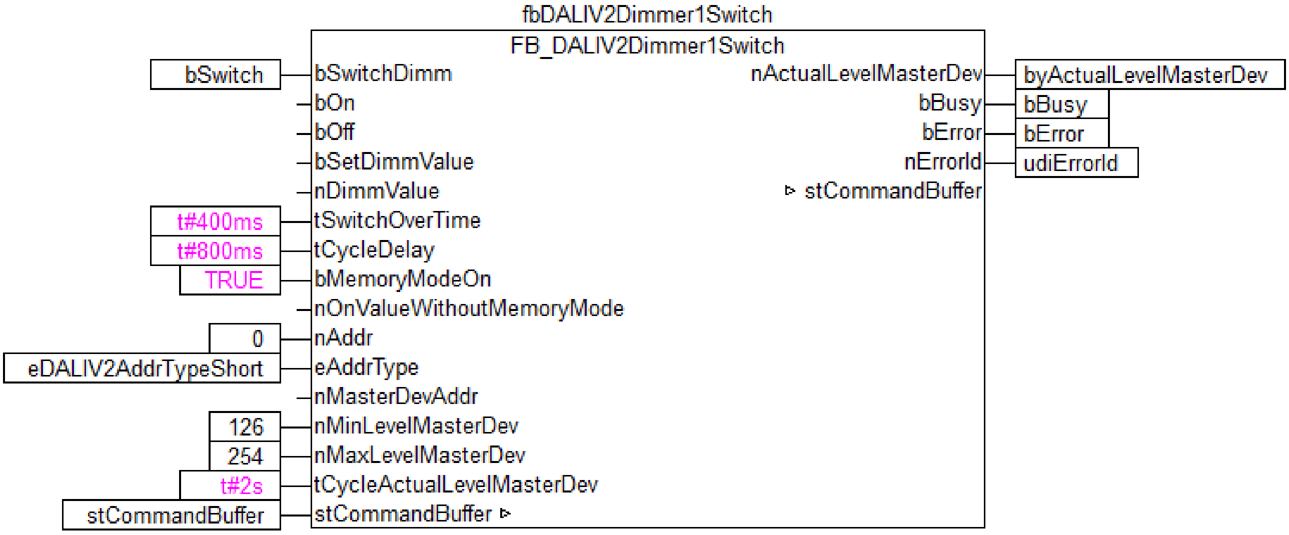


Fig. 1: Sample-PC-Comm.gif

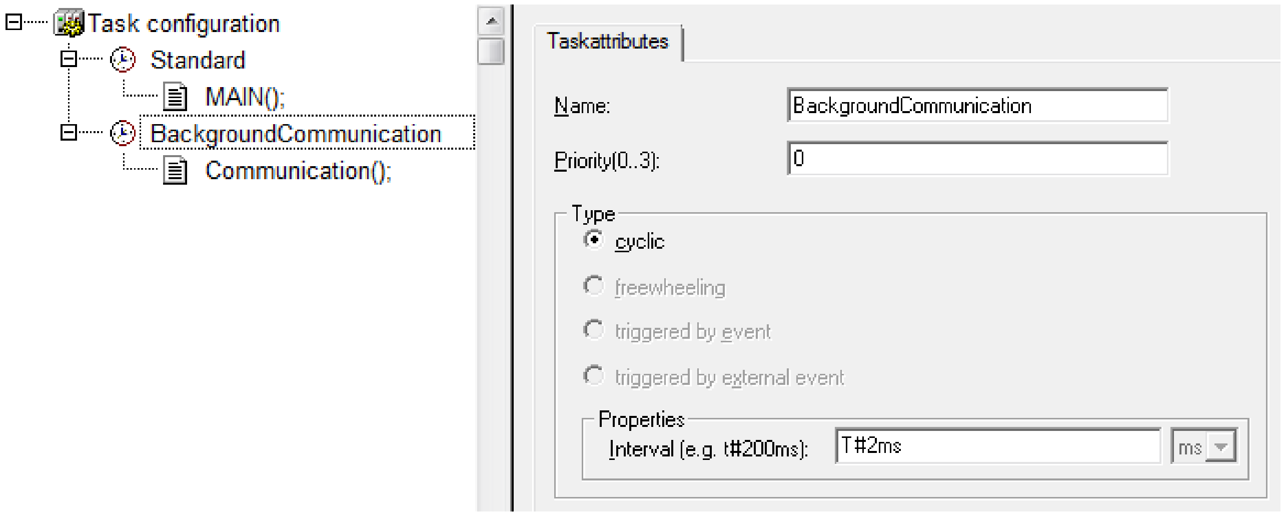
Create a MAIN program (CFC) in which the block FB_DALIV2Dimmer1Switch() [▸ 55] is called up. Connect the input *bSwitchDimm* of the dimmer block with the global variable *bSwitch* and *stCommandBuffer* with the global variable *stCommandBuffer*.



Parameter

Make sure that the specified parameters nMinlevelMasterDevice and nMaxLevelMasterDevice match the minimum and maximum values of the device, in order to avoid malfunction.

Under Task Configuration create a new task for the background communication. Add the communication program to this task. Assign a higher priority (lower number) and shorter interval time to this task than for the standard task. More detailed information can be found in the FB_DALIV2Communication() [▸ 93] block description.



Name: BackgroundCommunication

Priority(0..3): 0

Type
- ● cyclic
- ○ freewheeling
- ○ triggered by event
- ○ triggered by external event

Properties

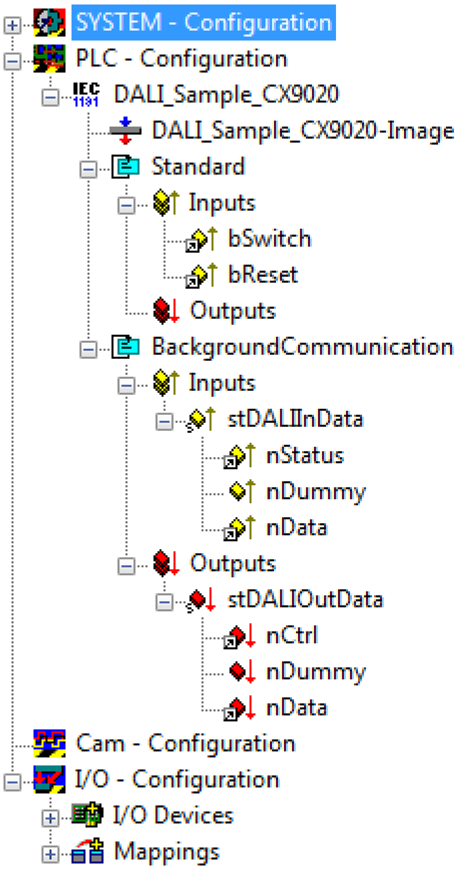
Interval (e.g. t#200ms): T#2ms ms

Load the project to the CX as the boot project and save it.

Configuration in the System Manager

Create a new TwinCAT System Manager project, select the CX as the target system, and search for the associated hardware.

Add the PLC program created above under PLC configuration. The two tasks are listed when the PLC project is expanded in the tree view. Expand the tasks – all global input and output variables should be allocated to the standard task. However, since the variables *stDALIInData* and *stDALIOutData* are to be processed faster, move them to the background communication task via drag & drop.



```
SYSTEM - Configuration
PLC - Configuration
    DALI_Sample_CX9020
        DALI_Sample_CX9020-Image
        Standard
            Inputs
                bSwitch
                bReset
            Outputs
        BackgroundCommunication
            Inputs
                stDALIInData
                    nStatus
                    nDummy
                    nData
            Outputs
                stDALIOutData
                    nCtrl
                    nDummy
                    nData
Cam - Configuration
I/O - Configuration
    I/O Devices
    Mappings
```

Now link the global variables of the PLC program with the Bus Terminal inputs and outputs, create the allocations, and activate the configuration. Then start the device in run mode.
Your CX is now ready for use.

The lamp can now be controlled by pressing or holding the dimmer button. Use the Reset button to reset the entries in *arrBufferMaximumDemandMeter* and *arrBufferOverflowCounter*.

4.2 KL6811 Integration into TwinCAT (BC9191)

This example explains how to write a simple PLC program for DALI in TwinCAT and how to link it with the hardware. The task is to control an individual dimmable lamp via a button.

https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019288203/.zip https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019288203/.zip

Hardware

Setting up the components

The following hardware is required:

- 1x Bus Terminal Controller BC9191

- 1x potential feed terminal 24V DC e.g. KL9190
- 1x digital 2-channel input terminal KL1002 (for the dimming and reset functions)
- 1x DALI terminal KL6811
- 1x end terminal KL9010

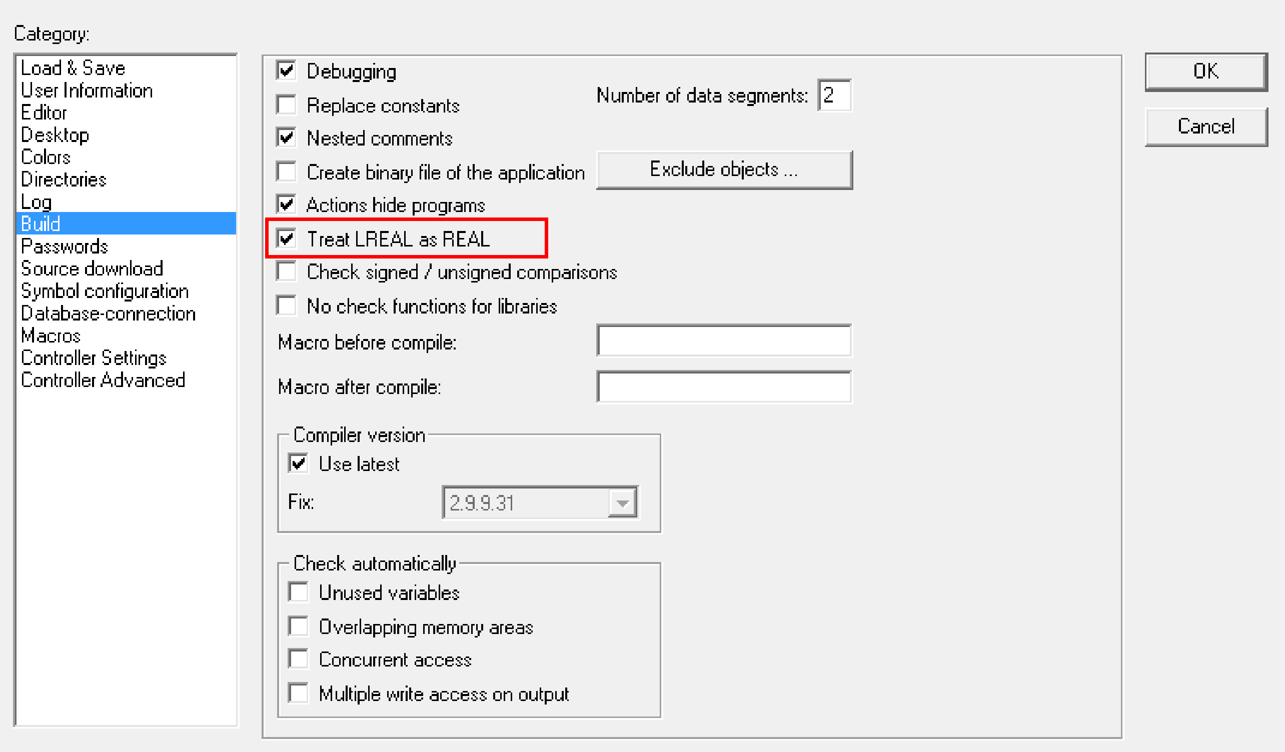
Set up the hardware and the DALI components as described in the associated documentation.

This example assumes that a Dim button was connected to the first KL1002 input and a Reset button to the second, and that a dimmable lamp is connected to DALI address 0. Set the initial fade rate of the ballast to 7 to achieve suitable dimming.

Software

Creation of the PLC program

Create a new PLC project for BC-based systems (BCxx50 via AMS) and add the libraries *TcDALIV2.lbx* and *TcSystemBCxx50.lbx*. Then navigate to *Project→Options*... *→Build* and select *TreatLREAL as REAL*.



Category:

- Load & Save
- User Information
- Editor
- Desktop
- Colors
- Directories
- Log
- Build
- Passwords
- Source download
- Symbol configuration
- Database-connection
- Macros
- Controller Settings
- Controller Advanced

☑ Debugging

Number of data segments: 2

☐ Replace constants

☑ Nested comments

☐ Create binary file of the application

☑ Actions hide programs

☑ Treat LREAL as REAL

☐ Check signed / unsigned comparisons

☐ No check functions for libraries

Macro before compile:

Macro after compile:

Compiler version

☑ Use latest

Fix: 2.9.9.31

Check automatically

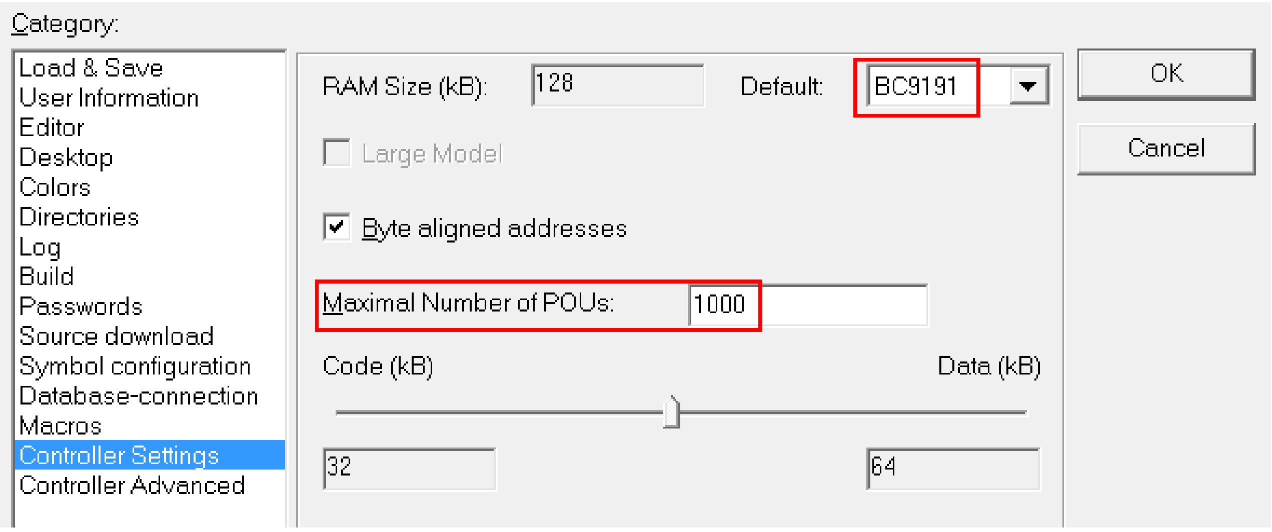
☐ Unused variables

☐ Overlapping memory areas

☐ Concurrent access

☐ Multiple write access on output

Under *Controller Settings*, select *BC9191* and set the *Maximal Number of POUs* to 1000.



Category:

- Load & Save
- User Information
- Editor
- Desktop
- Colors
- Directories
- Log
- Build
- Passwords
- Source download
- Symbol configuration
- Database-connection
- Macros
- Controller Settings
- Controller Advanced

RAM Size (kB): 128

Default: BC9191

□ Large Model

☑ Byte aligned addresses

Maximal Number of POUs: 1000

Code (kB): 32

Data (kB): 64

Next, generate the following global variables:

```
VAR_GLOBAL
    bSwitch      AT %I*      : BOOL;
    bReset       AT %I*      : BOOL;
    stDALIInData     AT %I* : ST_DALIV2InData;
    stDALIOutData    AT %Q* : ST_DALIV2OutData;
    stCommandBuffer         : ST_DALIV2CommandBuffer;
END_VAR
```

bSwitch: Input variable for the Dim button.

bReset: Input variable for the Reset button.

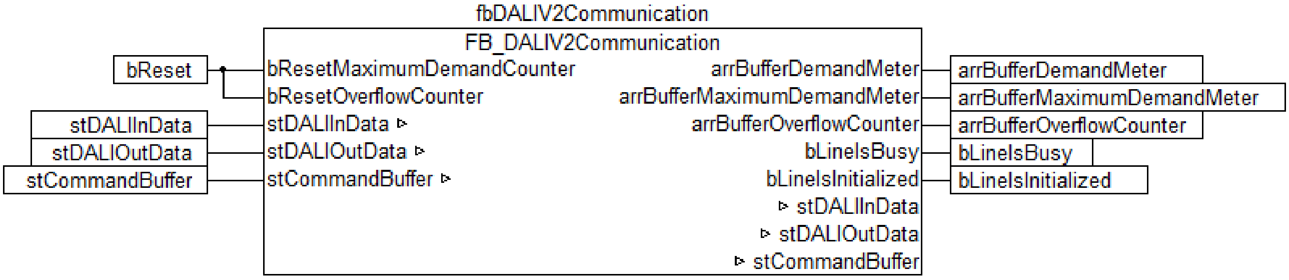
stDALIInData: Input variable [⏵ 384] for the DALI terminal.

stDALIOutData: Output variable [⏵ 384] for the DALI terminal.

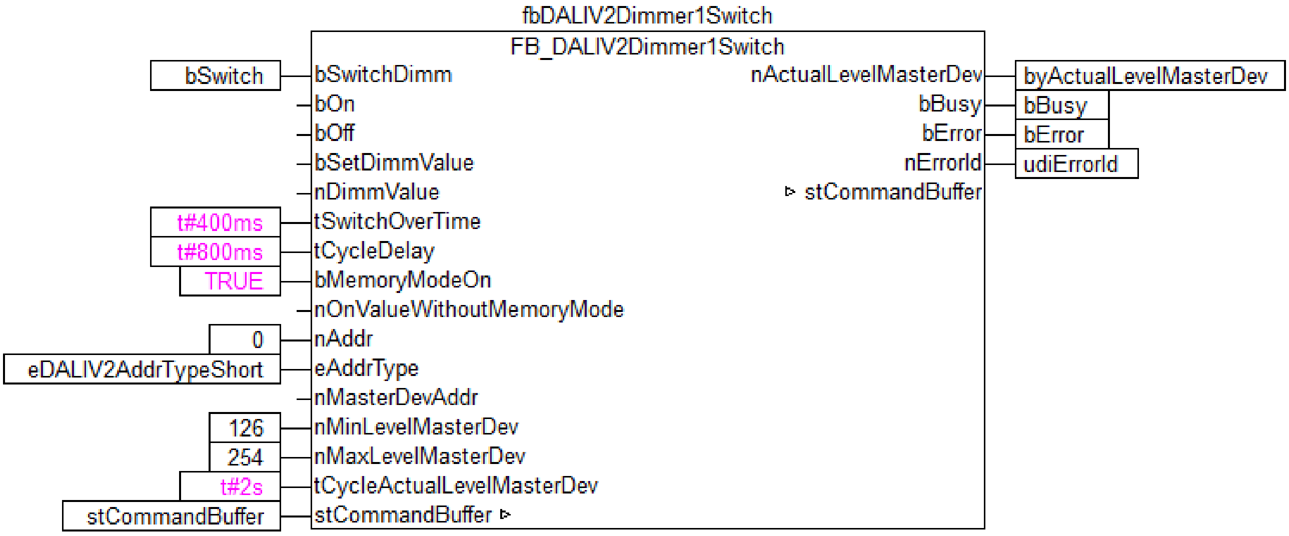
stCommandBuffer : Required for the communication with DALI.

Since BC devices can only process one task, communication with DALI cannot run separately.

Therefore, create a MAIN program (CFC) in which the FB_DALIV2Communication() [⏵ 93] and FB_DALIV2Dimmer1Switch() [⏵ 55] function blocks are called. Make sure to link the communication block with *stDALIInData*, *stDALIOutData* and *stCommandBuffer*.



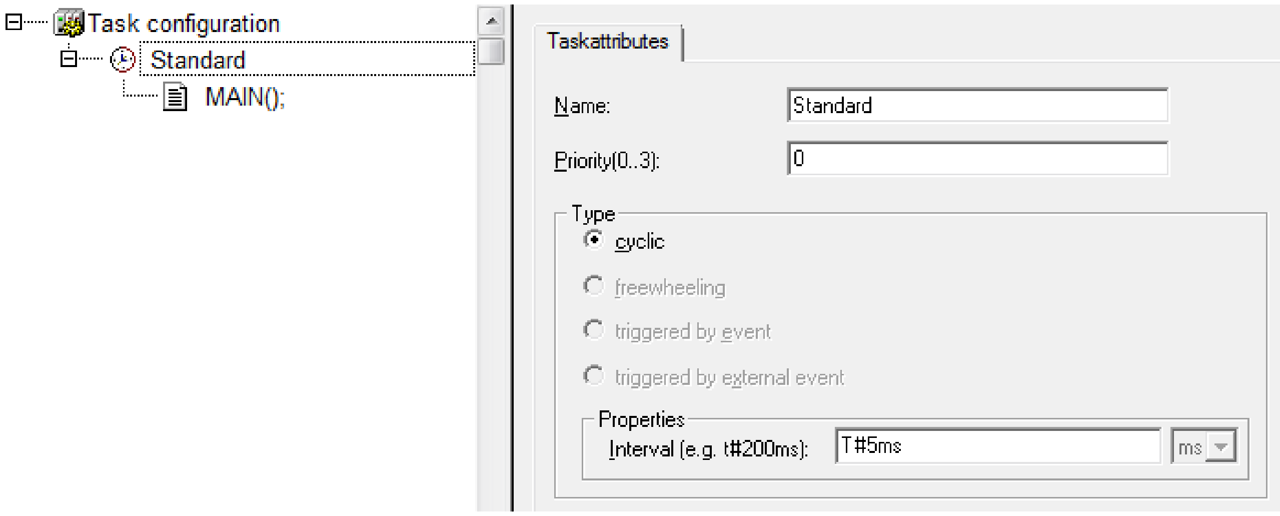
Connect the input *bSwitchDimm* of the dimmer block with the global variable *bSwitch* and *stCommandBuffer* with the global variable *stCommandBuffer*.



Parameter

Make sure that the specified parameters nMinlevelMasterDevice and nMaxLevelMasterDevice match the minimum and maximum values of the device, in order to avoid malfunction.

Go to the task configuration and give the task a lower interval time. More detailed information can be found in the FB_DALIV2Communication() [▸ 93] block description.



Name: Standard

Priority(0..3): 0

Type

● cyclic

○ freewheeling

○ triggered by event

○ triggered by external event

Properties

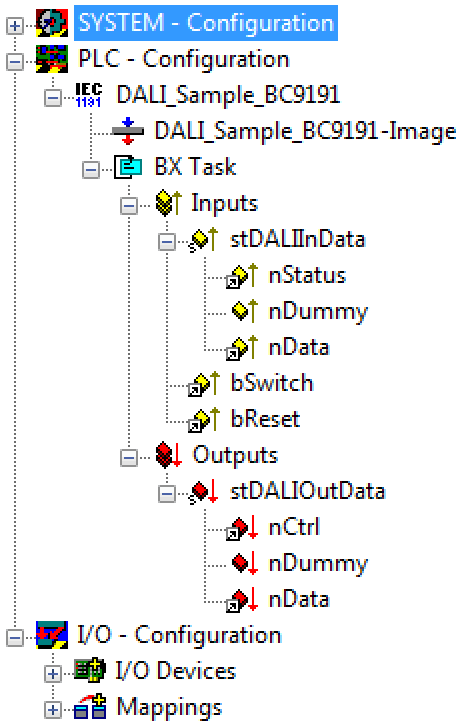
Interval (e.g. t#200ms): T#5ms ms

Now load the project as a boot project to the BC and save it.

Configuration in the System Manager

Create a new TwinCAT System Manager project, select the BC as the target system, and search for the associated hardware.

Add the PLC program created above under PLC configuration.



```
SYSTEM - Configuration
PLC - Configuration
    DALI_Sample_BC9191
        DALI_Sample_BC9191-Image
        BX Task
            Inputs
                stDALIInData
                    nStatus
                    nDummy
                    nData
                bSwitch
                bReset
            Outputs
                stDALIOutData
                    nCtrl
                    nDummy
                    nData
I/O - Configuration
    I/O Devices
    Mappings
```

Now link the global variables of the PLC program with the Bus Terminal inputs and outputs, create the allocations, and activate the configuration. Then start the device in run mode.
Your BC is now ready for use.

The lamp can now be controlled by pressing or holding the dimmer button. Use the Reset button to reset the entries in *arrBufferMaximumDemandMeter* and *arrBufferOverflowCounter*.

4.3 KL6821 Integration into TwinCAT (CX9020)

This example explains how to write a simple PLC program for DALI in TwinCAT and how to link it with the hardware. The task is to control an individual dimmable lamp via a button.

https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019289611/.zip https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019289611/.zip

Hardware

Setting up the components

The following hardware is required:

- 1x Embedded PC CX9020
- 1x digital 4-channel input terminal KL1104 (for the dimming and reset functions)
- 1x DALI terminal KL6821
- 1x end terminal KL9010

Set up the hardware and the DALI components as described in the associated documentation.

This example assumes that a Dim button was connected to the first KL1002 input and a Reset button to the second, and that a dimmable lamp is connected to DALI address 0. Set the initial fade rate of the ballast to 7 to achieve suitable dimming.

Software

Creation of the PLC program

Create a new PLC project for PC-based systems (ARM) and add the *TcDALIV2.lib* library.

Next, generate the following global variables:

```
VAR_GLOBAL
    bSwitch       AT %I*      : BOOL;
    bReset        AT %I*      : BOOL;
    stKL6821InData    AT %I* : ST_KL6821InData;
    stKL6821OutData   AT %Q* : ST_KL6821OutData;
    stCommandBuffer          : ST_DALIV2CommandBuffer;
END_VAR
```

bSwitch: Input variable for the Dim button.

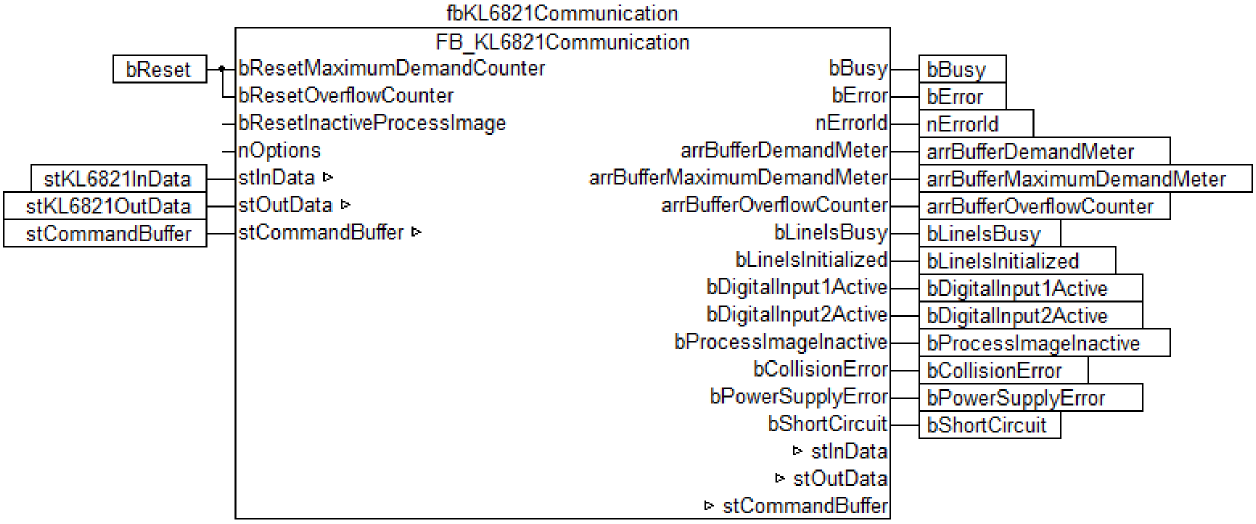
bReset: Input variable for the Reset button.

stDALIInData: Input variable [▸ 385] for the DALI terminal.

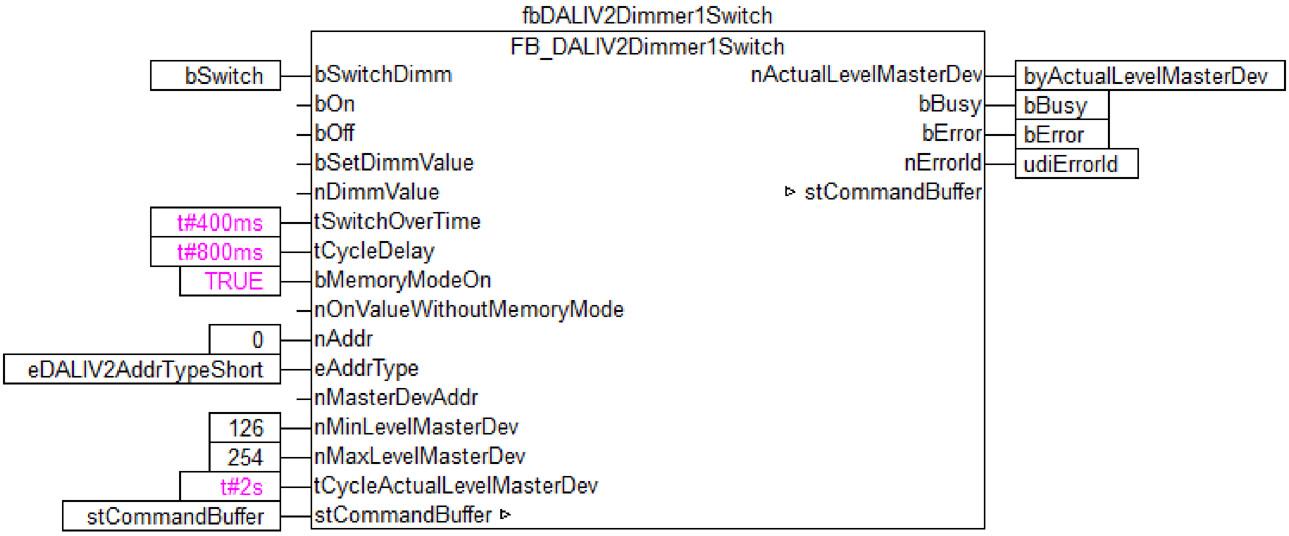
stDALIOutData: Output variable [▸ 385] for the DALI terminal.

stCommandBuffer : Required for the communication with DALI.

Then create a program (CFC) for background communication with DALI. The FB_KL6821Communication() [▸ 101] block is called in this program. Make sure to link the communication block with *stDALIInData*, *stDALIOutData* and *stCommandBuffer*.



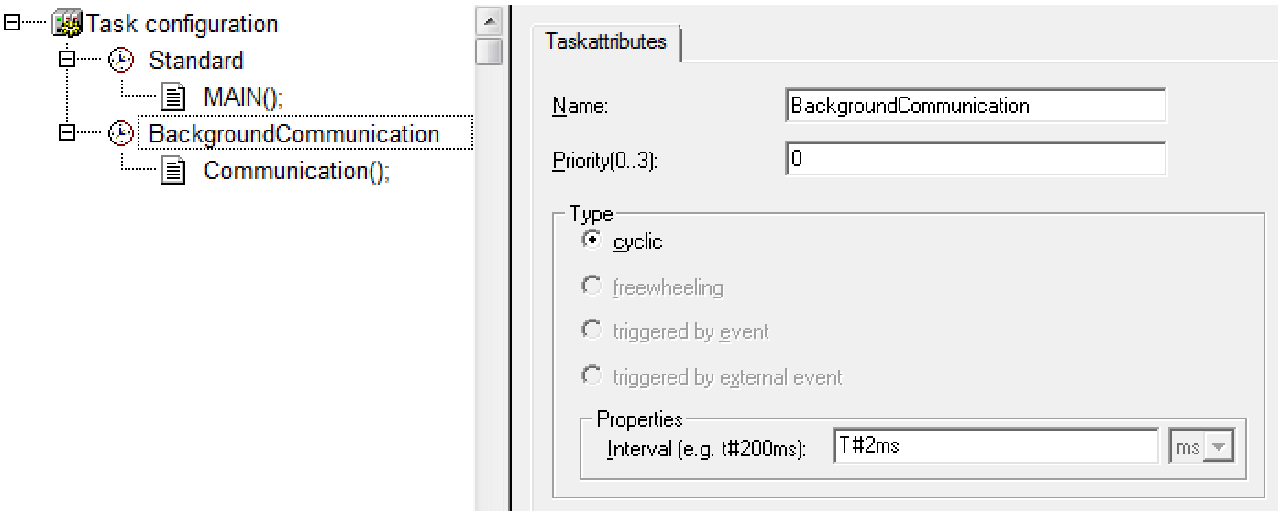
Create a MAIN program (CFC) in which the block FB_DALIV2Dimmer1Switch() [▸ 55] is called up. Connect the input *bSwitchDimm* of the dimmer block with the global variable *bSwitch* and *stCommandBuffer* with the global variable *stCommandBuffer*.



Parameter

Make sure that the specified parameters nMinlevelMasterDevice and nMaxLevelMasterDevice match the minimum and maximum values of the device, in order to avoid malfunction.

Under Task Configuration create a new task for the background communication. Add the communication program to this task. Assign a higher priority (lower number) and shorter interval time to this task than for the standard task. More detailed information can be found in the FB_DALIV2Communication() [▸ 93] block description.



Name: BackgroundCommunication

Priority(0..3): 0

Type
- ● cyclic
- ○ freewheeling
- ○ triggered by event
- ○ triggered by external event

Properties

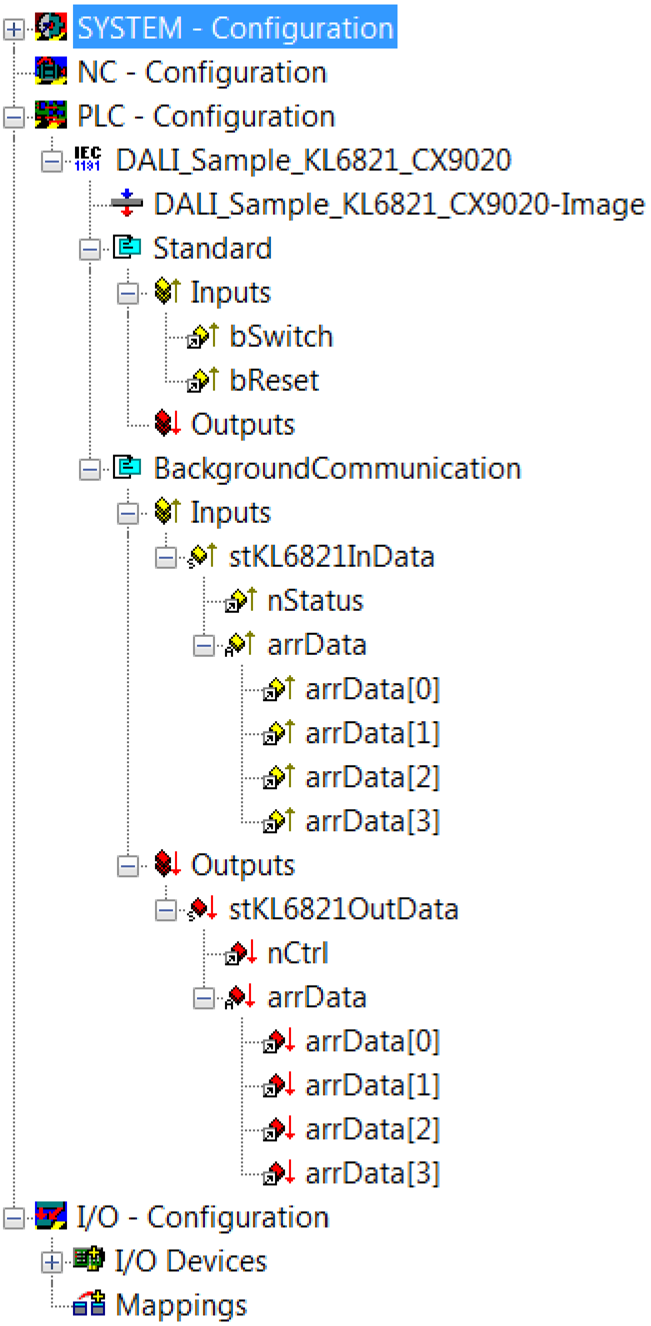
Interval (e.g. t#200ms): T#2ms ms

Load the project to the CX as the boot project and save it.

Configuration in the System Manager

Create a new TwinCAT System Manager project, select the CX as the target system, and search for the associated hardware.

Add the PLC program created above under PLC configuration. The two tasks are listed when the PLC project is expanded in the tree view. Expand the tasks – all global input and output variables should be allocated to the standard task. However, since the variables *stDALIInData* and *stDALIOutData* are to be processed faster, move them to the background communication task via drag & drop.



```
SYSTEM - Configuration
NC - Configuration
PLC - Configuration
    DALI_Sample_KL6821_CX9020
        DALI_Sample_KL6821_CX9020-Image
        Standard
            Inputs
                bSwitch
                bReset
            Outputs
        BackgroundCommunication
            Inputs
                stKL6821InData
                    nStatus
                    arrData
                        arrData[0]
                        arrData[1]
                        arrData[2]
                        arrData[3]
            Outputs
                stKL6821OutData
                    nCtrl
                    arrData
                        arrData[0]
                        arrData[1]
                        arrData[2]
                        arrData[3]
I/O - Configuration
    I/O Devices
    Mappings
```

Now link the global variables of the PLC program with the Bus Terminal inputs and outputs, create the allocations, and activate the configuration. Then start the device in run mode.
Your CX is now ready for use.

The lamp can now be controlled by pressing or holding the dimmer button. Use the Reset button to reset the entries in *arrBufferMaximumDemandMeter* and *arrBufferOverflowCounter*.

4.4 KL6821 Integration into TwinCAT (BC9191)

This example explains how to write a simple PLC program for DALI in TwinCAT and how to link it with the hardware. The task is to control an individual dimmable lamp via a button.

https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019291019/.zip https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019291019/.zip

Hardware

Setting up the components

The following hardware is required:

- 1x Bus Terminal Controller BC9191
- 1x potential feed terminal 24V DC e.g. KL9190
- 1x digital 2-channel input terminal KL1104 (for the dimming and reset functions)
- 1x DALI terminal KL6821
- 1x end terminal KL9010

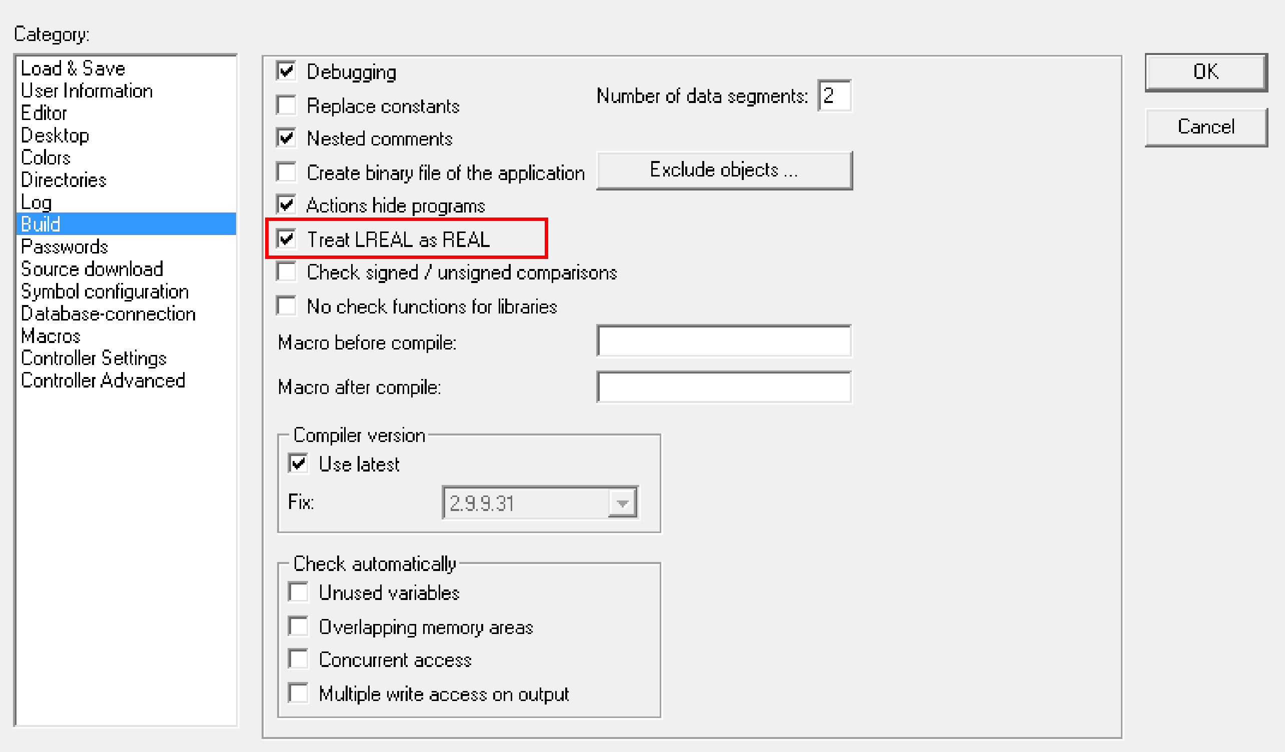
Set up the hardware and the DALI components as described in the associated documentation.

This example assumes that a Dim button was connected to the first KL1002 input and a Reset button to the second, and that a dimmable lamp is connected to DALI address 0. Set the initial fade rate of the ballast to 7 to achieve suitable dimming.

Software

Creation of the PLC program

Create a new PLC project for BC-based systems (BCxx50 via AMS) and add the libraries *TcDALIV2.lbx* and *TcSystemBCxx50.lbx*. Then navigate to *Project→Options*... →*Build* and select *TreatLREAL as REAL*.



Category:

- Load & Save
- User Information
- Editor
- Desktop
- Colors
- Directories
- Log
- Build
- Passwords
- Source download
- Symbol configuration
- Database-connection
- Macros
- Controller Settings
- Controller Advanced

☑ Debugging

☐ Replace constants

Number of data segments: 2

☑ Nested comments

☐ Create binary file of the application

☑ Actions hide programs

☑ Treat LREAL as REAL

☐ Check signed / unsigned comparisons

☐ No check functions for libraries

Macro before compile:

Macro after compile:

Compiler version

☑ Use latest

Fix: 2.9.9.31

Check automatically

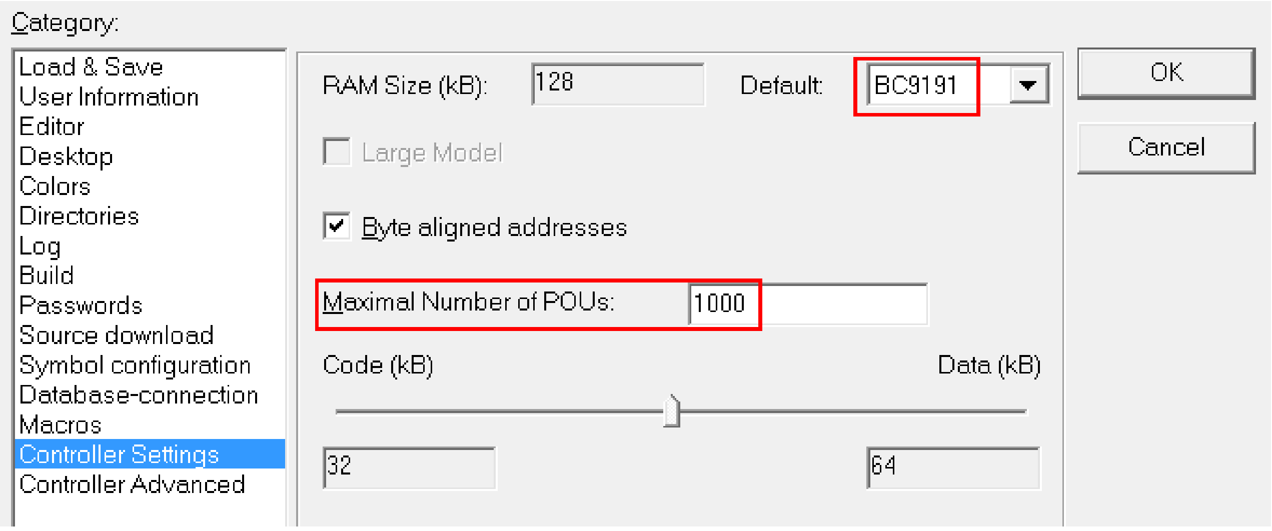
☐ Unused variables

☐ Overlapping memory areas

☐ Concurrent access

☐ Multiple write access on output

Under *Controller Settings*, select *BC9191* and set the *Maximal Number of POUs* to 1000.



Category:
- Load & Save
- User Information
- Editor
- Desktop
- Colors
- Directories
- Log
- Build
- Passwords
- Source download
- Symbol configuration
- Database-connection
- Macros
- Controller Settings
- Controller Advanced

RAM Size (kB): 128

Default: BC9191

□ Large Model

☑ Byte aligned addresses

Maximal Number of POUs: 1000

Code (kB): 32

Data (kB): 64

Next, generate the following global variables:

```
VAR_GLOBAL
    bSwitch        AT %I*      : BOOL;
    bReset         AT %I*      : BOOL;
    stKL6821InData    AT %I* : ST_KL6821InData;
    stKL6821OutData   AT %Q* : ST_KL6821OutData;
    stCommandBuffer          : ST_DALIV2CommandBuffer;
END_VAR
```

bSwitch: Input variable for the Dim button.

bReset: Input variable for the Reset button.

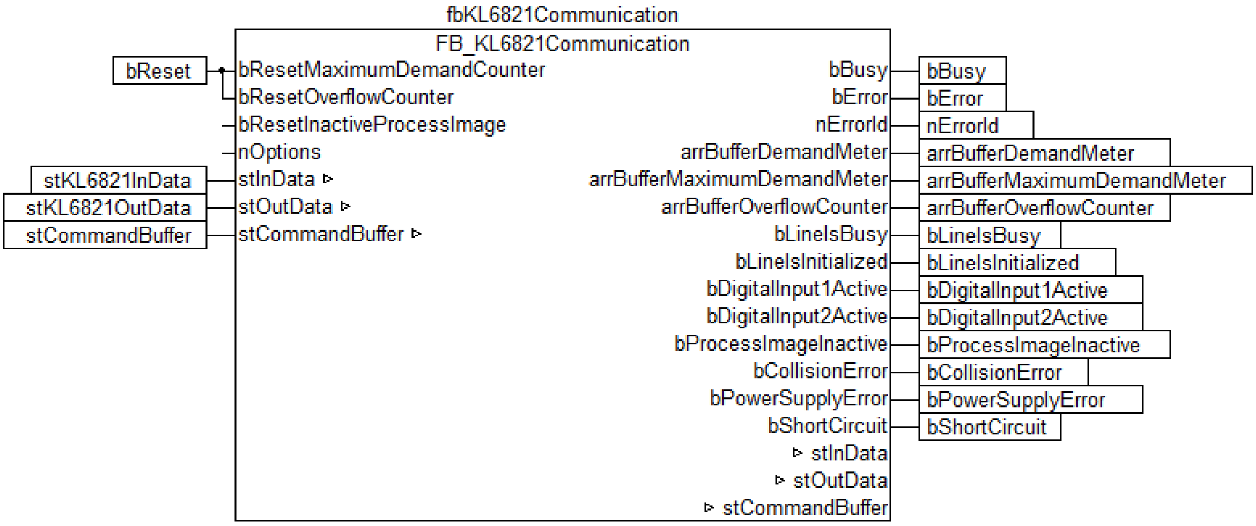
stDALIInData: Input variable [▶ 385] for the DALI terminal.

stDALIOutData: Output variable [▶ 385] for the DALI terminal.

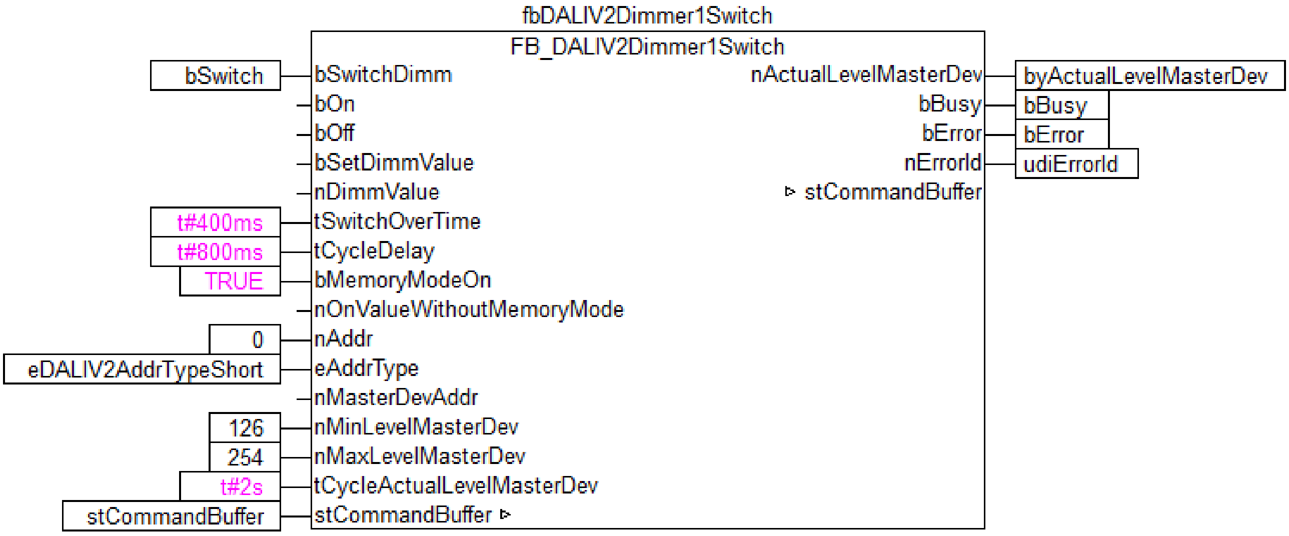
stCommandBuffer : Required for the communication with DALI.

Since BC devices can only process one task, communication with DALI cannot run separately.

Therefore, create a MAIN program (CFC) in which the FB_KL6821Communication() [▶ 101] and FB_DALIV2Dimmer1Switch() [▶ 55] function blocks are called. Make sure to link the communication block with *stDALIInData*, *stDALIOutData* and *stCommandBuffer*.



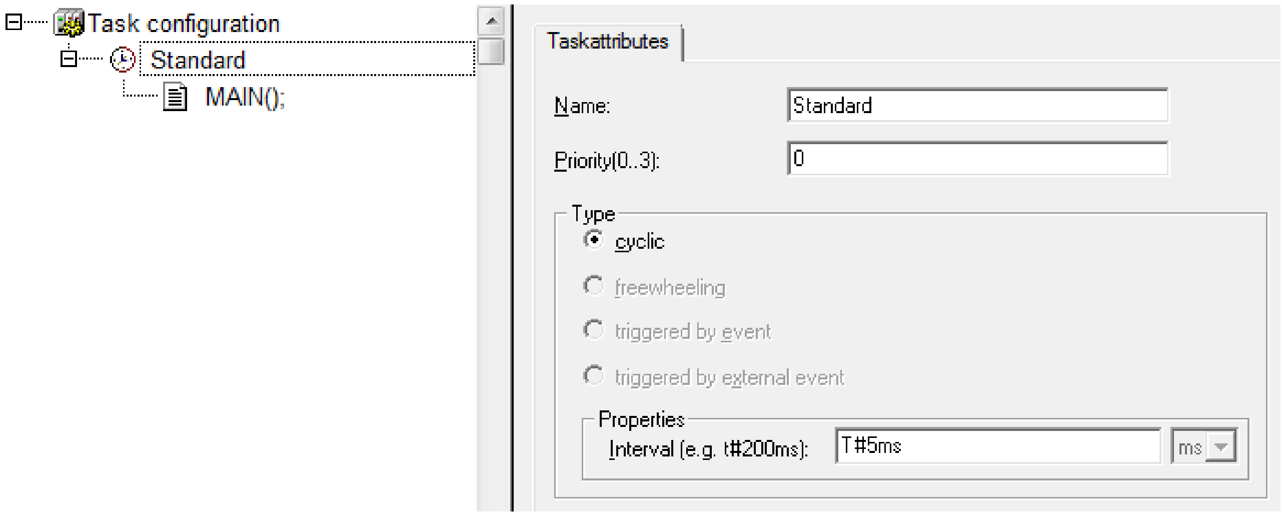
Connect the input *bSwitchDimm* of the dimmer block with the global variable *bSwitch* and *stCommandBuffer* with the global variable *stCommandBuffer*.



Parameter

Make sure that the specified parameters nMinlevelMasterDevice and nMaxLevelMasterDevice match the minimum and maximum values of the device, to avoid malfunction.

Go to the task configuration and give the task a lower interval time. More detailed information can be found in the FB_DALIV2Communication() [▶ 93] block description.


Task configuration
Standard
MAIN();

Taskattributes

Name: Standard

Priority(0..3): 0

Type

● cyclic

○ freewheeling

○ triggered by event

○ triggered by external event

Properties

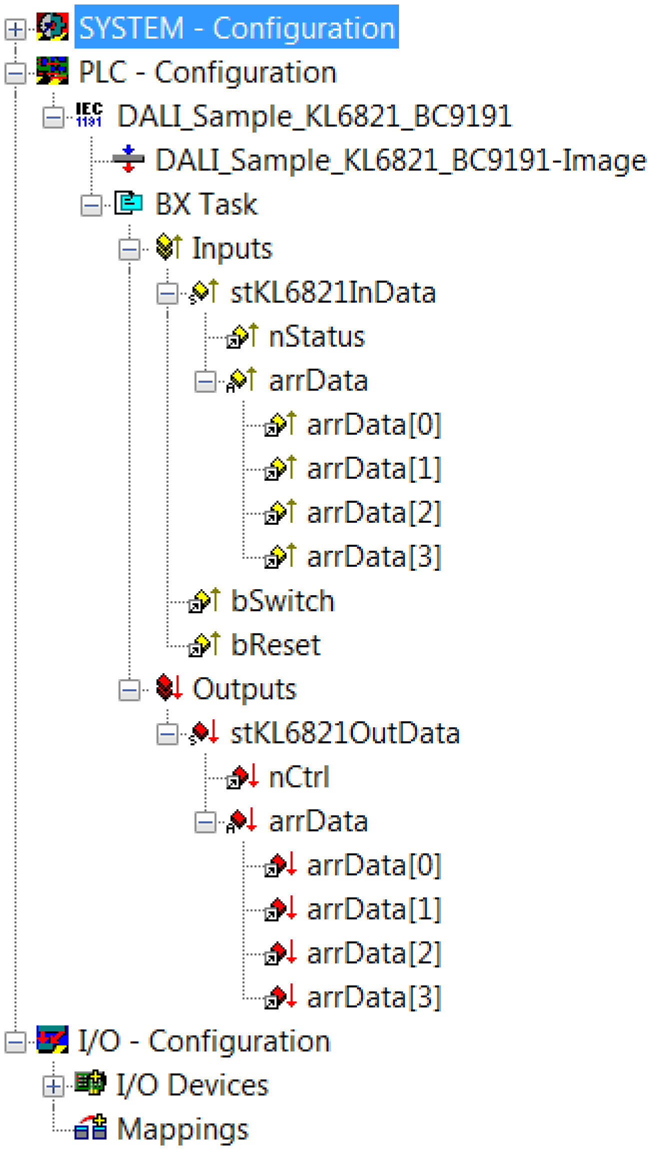
Interval (e.g. t#200ms): T#5ms ms

Now load the project as a boot project to the BC and save it.

Configuration in the System Manager

Create a new TwinCAT System Manager project, select the BC as the target system, and search for the associated hardware.

Add the PLC program created above under PLC configuration.



```
SYSTEM - Configuration
PLC - Configuration
    DALI_Sample_KL6821_BC9191
        DALI_Sample_KL6821_BC9191-Image
        BX Task
            Inputs
                stKL6821InData
                    nStatus
                    arrData
                        arrData[0]
                        arrData[1]
                        arrData[2]
                        arrData[3]
                bSwitch
                bReset
            Outputs
                stKL6821OutData
                    nCtrl
                    arrData
                        arrData[0]
                        arrData[1]
                        arrData[2]
                        arrData[3]
I/O - Configuration
    I/O Devices
    Mappings
```

Now link the global variables of the PLC program with the Bus Terminal inputs and outputs, create the allocations, and activate the configuration. Then start the device in run mode.
Your BC is now ready for use.

The lamp can now be controlled by pressing or holding the dimmer button. Use the Reset button to reset the entries in *arrBufferMaximumDemandMeter* and *arrBufferOverflowCounter*.

5 Programming

Installation

Beginning with TwinCAT 2.11 Build 2229 (R3 and x64 Engineering), the libraries "Tc-DALIV2.lib/.lb6/.lbx" will be installed automatically.

Hardware documentation in Beckhoff Information System: KL6811 - Master Terminal for DALI/DSI

Hardware documentation in Beckhoff Information System: KL6821 - Master Terminal for DALI2

Further libraries are required

For PC systems (x86) and Embedded-PCs (CXxxxx):

- Standard.lib
- TcBase.lib
- TcSystem.lib

For Bus Terminal Controller of BCxx00 series:

- Standard.lb6
- TcPlcUtilitiesBC.lb6
- PlcHelperBC.lb6
- PlcSystemBC.lb6

For Bus Terminal Controller of BCxx50, BCxx20 and BC9191 series:

- Standard.lbx
- TcBaseBCxx50.lbx
- TcSystemBCxx50.lbx

For Bus Terminal Controller of BXxx00 series:

- Standard.lbx
- TcBaseBX.lbx
- TcSystemBX.lbx

Memory usage

By linking the library PLC program memory is already consumed. Depending on the application program the remaining memory can not be sufficient.

5.1 POUs

5.1.1 Emergency lighting function blocks

The function blocks and variables for emergency lighting supply units with DALI interface are described below. All function blocks described below call 'application extended commands'. According to the DALI standard (IEC 62386) these commands are within the range 224 to 255. Because of the existence of a variety of application extended commands, it is necessary to use the FB_DALIV2EnableDeviceType() [▸ 350] function block to specify which type of control gears (emergency lighting, discharge lamps, ...) should react to the extension commands. A detailed description of the individual DALI commands and the variables for emergency lighting supply units can be found in Part 202 of the IEC 62386 standard.

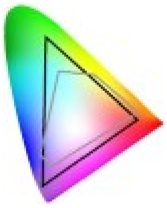
Name	Description
FB_DALIV2Inhibit [▶ 220]	Prevents the control gear from switching to emergency mode for 15 minutes.
FB_DALIV2QueryBatteryCharge [▶ 221]	The variable BATTERY CHARGE [▶ 370]
FB_DALIV2QueryDurationTestResult [▶ 222]	The variable DURATION TEST RESULT [▶ 370]
FB_DALIV2QueryEmergencyLevel [▶ 223]	The variable EMERGENCY LEVEL [▶ 368]
FB_DALIV2QueryEmergencyMaxLevel [▶ 224]	The variable EMERGENCY MAX LEVEL [▶ 368]
FB_DALIV2QueryEmergencyMinLevel [▶ 225]	The variable EMERGENCY MIN LEVEL [▶ 368]
FB_DALIV2QueryEmergencyMode [▶ 226]	The variable EMERGENCY MODE [▶ 370]
FB_DALIV2QueryEmergencyStatus [▶ 227]	The variable EMERGENCY STATUS [▶ 371]
FB_DALIV2QueryFailureStatus [▶ 228]	The variable FAILURE STATUS [▶ 371]
FB_DALIV2QueryFeatures [▶ 229]	The variable FEATURES [▶ 371] (performance characteristics) is read from the control gear.
FB_DALIV2QueryLampEmergencyTime [▶ 231]	The variable LAMP EMERGENCY TIME [▶ 370]
FB_DALIV2QueryLampTotalOperationTime [▶ 232]	The variable LAMP TOTAL OPERATION TIME [▶ 370]
FB_DALIV2QueryRatedDuration [▶ 233]	The variable RATED DURATION [▶ 370]
FB_DALIV2QueryTestTiming [▶ 234]	Depending on the contents of the DTR (Data Transfer Register), the variables FUNCTION TEST DELAY TIME [▶ 369], DURATION TEST DELAY TIME [▶ 369], FUNCTION TEST INTERVAL [▶ 369], DURATION TEST INTERVAL [▶ 369], TEST EXECUTION TIMEOUT [▶ 370], PROLONG TIME [▶ 368]
FB_DALIV2ReLightResetInhibit [▶ 235]	The control gear is switched back to emergency mode (in the absence of mains voltage).
FB_DALIV2ResetDurationTestDoneFlag [▶ 236]	The 'Duration test completed and result is valid' flag is reset.
FB_DALIV2ResetFunctionTestDoneFlag [▶ 237]	The 'Function test completed and result is valid' flag is reset.
FB_DALIV2ResetLampTime [▶ 238]	The variables LAMP EMERGENCY TIME [▶ 370] and LAMP TOTAL OPERATION TIME [▶ 370]
FB_DALIV2Rest [▶ 239]	Switches the lamp off when emergency mode is active.
FB_DALIV2StartDurationTest [▶ 240]	Starts the duration test.
FB_DALIV2StartFunctionTest [▶ 241]	Starts the function test.
FB_DALIV2StopTest [▶ 242]	Stops any type of function test or duration test.
FB_DALIV2StoreDTRAsDurationTestInterval [▶ 243]	Writes the value of the DTR (data transfer register) into the variable DURATION TEST INTERVAL [▶ 369]
FB_DALIV2StoreDTRAsEmergencyLevel [▶ 244]	Writes the value of the DTR (data transfer register) into the variable EMERGENCY LEVEL [▶ 368]
FB_DALIV2StoreDTRAsFunctionTestInterval [▶ 245]	Writes the value of the DTR (data transfer register) into the variable FUNCTION TEST INTERVAL [▶ 369]
FB_DALIV2StoreDTRAsProlongTime [▶ 246]	Writes the value of the DTR (data transfer register) into the variable PROLONG TIME [▶ 368]
FB_DALIV2StoreDTRAsTestDelayTimeHighByte [▶ 247]	Writes the value of the DTR (data transfer register) into the high-order byte of variable TEST DELAY TIME [▶ 368]

Name	Description
FB_DALIV2StoreDTRAsTestDelayTimeLowByte [▶ 248]	Writes the value of the DTR (data transfer register) into the low-order byte of the variable TEST DELAY TIME
FB_DALIV2StoreDTRAsTestExecutionTimeout [▶ 249]	Writes the value of the DTR (data transfer register) into the variable TEST EXECUTION TIMEOUT [▶ 370]

These commands belong to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *Tc-DALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

5.1.2 Colour/colour temperature control function blocks

The function blocks and variables for lamps for color/color temperature control with DALI interface are described below. All function blocks described below call 'application extended commands'. According to the DALI standard (IEC 62386) these commands are within the range 224 to 255. Because of the existence of a variety of application extended commands, it is necessary to use the FB_DALIV2EnableDeviceType() [▶ 350] function block to specify which type of control gears (emergency lighting, discharge lamps, ...) should react to the extension commands. A precise description of the individual DALI commands and the variables for lamps for color/color temperature control with DALI interface can be found in IEC 62386 part 209.



Name	Description
FB_DALIV2Activate [▶ 282]	Starts a new cross-fade. A running cross-fade will be ended beforehand.
FB_DALIV2AssignColourToLinkedChannel [▶ 284]	Linked output channels are assigned to the defined color.
FB_DALIV2ColourTemperatureTcStepCooler [▶ 285]	The value COLOUR TEMPERATURE Tc [▶ 374] is decremented by 1 Mirek.
FB_DALIV2ColourTemperatureTcStepWarmer [▶ 286]	The value COLOUR TEMPERATURE Tc [▶ 374] is incremented by 1 Mirek.
FB_DALIV2CopyReportToTemporary [▶ 287]	The color settings report is copied to the temporary color settings.
FB_DALIV2QueryAssignedColour [▶ 288]	Reads the assigned color of the specified output channel.
FB_DALIV2QueryColourStatus [▶ 290]	The variable COLOUR STATUS [▶ 378]is read from the control gear.
FB_DALIV2QueryColourTypeFeatures [▶ 291]	The color representations supported by the control gear are read out.
FB_DALIV2QueryColourValue [▶ 293]	The specified color value is read from the control gear.
FB_DALIV2QueryGearFeaturesStatus [▶ 296]	The variable GEAR FEATURES/STATUS [▶ 379] is read from the control gear.
FB_DALIV2QueryRGBWAFControl [▶ 297]	The variable RGBWAF CONTROL [▶ 379] is read from the control gear.
FB_DALIV2SetTemporaryColourTemperatureTc [▶ 298]	Saves the value in the variable TEMPORARY COLOUR TEMPERATURE Tc [▶ 374] of the control gear.
FB_DALIV2SetTemporaryPrimaryNDimlevel [▶ 300]	Saves the value in the variable TEMPORARY PRIMARY N DIMLEVEL [▶ 374] of the control gear.

Name	Description
FB_DALIV2SetTemporaryRGBDimlevel [▶ 301]	Saves the values in the variables TEMPORARY RED DIMLEVEL [▶ 374], TEMPORARY GREEN DIMLEVEL [▶ 374] and TEMPORARY BLUE DIMLEVEL [▶ 374] of the control gear.
FB_DALIV2SetTemporaryRGBWAFControl [▶ 302]	Saves the value in the variable TEMPORARY RGBWAF CONTROL [▶ 374] of the control gear.
FB_DALIV2SetTemporaryWAFDimlevel [▶ 304]	Saves the values in the variables TEMPORARY WHITE DIMLEVEL [▶ 374], TEMPORARY AMBER DIMLEVEL [▶ 374] and TEMPORARY FREECOLOUR DIMLEVEL [▶ 374] of the control gear.
FB_DALIV2SetTemporaryXCoordinate [▶ 305]	Saves the value in the variable TEMPORARY x-COORDINATE [▶ 374] of the control gear.
FB_DALIV2SetTemporaryYCoordinate [▶ 306]	Saves the value in the variable TEMPORARY y-COORDINATE [▶ 374] of the control gear.
FB_DALIV2StartAutoCalibration [▶ 307]	The calibration procedure is started in order to measure the x-coordinate, the y-coordinate and the TY value of all supported primary colors.
FB_DALIV2StoreColourTemperatureTcLimit [▶ 309]	Saves the value in the variable COLOUR TEMPERATURE Tc COOLEST [▶ 374], COLOUR TEMPERATURE Tc WARMEST [▶ 374], COLOUR TEMPERATURE Tc PHYSICAL COOLEST [▶ 374] or COLOUR TEMPERATURE Tc PHYSICAL WARMEST [▶ 374] of the control gear.
FB_DALIV2StoreGearFeaturesStatus [▶ 310]	Saves the value in the variable GEAR FEATURES/STATUS [▶ 379] of the control gear.
FB_DALIV2StoreTYPrimaryN [▶ 312]	Saves the value in the variable TY PRIMARY N [▶ 374] of the control gear.
FB_DALIV2StoreXyCoordinatePrimaryN [▶ 313]	Copies the value from the variables TEMPORARY x-COORDINATE [▶ 374]and TEMPORARY y-COORDINATE [▶ 374] to the variables x-COORDINATE PRIMARY N [▶ 374] and y-COORDINATE PRIMARY N [▶ 374].
FB_DALIV2XCoordinateStepDown [▶ 314]	The variable x-COORDINATE is reduced by 256 steps (256 / 65536) without cross-fading.
FB_DALIV2XCoordinateStepUp [▶ 315]	The variable x-COORDINATE is increased by 256 steps (256 / 65536) without cross-fading.
FB_DALIV2YCoordinateStepDown [▶ 316]	The variable y-COORDINATE is reduced by 256 steps (256 / 65536) without cross-fading.
FB_DALIV2YCoordinateStepUp [▶ 318]	The variable y-COORDINATE is increased by 256 steps (256 / 65536) without cross-fading.

These commands belong to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the Enable Device Type 8 command, which can be sent with the function block FB_DALIV2EnableDeviceType(). From version 2.6.0 of the libraries TcDALIV2 and TcDALIV2AppExtCmds, however, the Enable Device Type 8 command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

5.1.3 Discharge lamps function blocks

The function blocks and variables for discharge lamps with DALI interface are described below. All function blocks described below call 'application extended commands'. According to the DALI standard (IEC 62386) these commands are within the range 224 to 255. Because of the existence of a variety of application extended commands, it is necessary to use the FB_DALIV2EnableDeviceType() [▸ 350] function block to specify which type of control gears (emergency lighting, discharge lamps, ...) should react to the extension commands. A precise description of the individual DALI commands and the variables for discharge lamps with DALI interface will be found in Part 203 of the IEC 62386 standard.

Name	Description
FB_DALIV2QueryActualHIDFailure [▸ 250]	The variable ACTUAL HID FAILURE [▸ 372]
FB_DALIV2QueryHIDFeatures [▸ 251]	The variable HID FEATURES [▸ 373]
FB_DALIV2QueryHIDStatus [▸ 252]	The variable HID STATUS [▸ 372]
FB_DALIV2QueryStoredHIDFailure [▸ 253]	The variable STORED HID FAILURE [▸ 373]
FB_DALIV2QueryThermalLoad [▸ 254]	The variable THERMAL LOAD [▸ 373]
FB_DALIV2QueryThermalOverloadTime [▸ 255]	The 16-bit variable THERMAL OVERLOAD TIME [▸ 373]
FB_DALIV2QueryThermalOverloadTimeHB [▸ 256]	The high-order byte of the 16-bit variable THERMAL OVERLOAD TIME
FB_DALIV2QueryThermalOverloadTimeLB [▸ 257]	The low-order byte of the 16-bit variable THERMAL OVERLOAD TIME
FB_DALIV2ResetStoredHIDFailure [▸ 258]	The variable STORED HID FAILURE

These commands belong to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

5.1.4 LED modules - function blocks

The function blocks and variables for LED modules with DALI interface are described below. All function blocks described below call 'application extended commands'. According to the DALI standard (IEC 62386) these commands are within the range 224 to 255. Because of the existence of a variety of application extended commands, it is necessary to use the FB_DALIV2EnableDeviceType() [▸ 350] function block to specify which type of control gears (emergency lighting, discharge lamps, ...) should react to the extension commands. A detailed description of the individual DALI commands and the variables for LED modules with DALI interface can be found in standard IEC 62386 Part 207.

Name	Description
FB_DALIV2DisableCurrentProtector [▸ 259]	The command disables the current protection device of the control gear.
FB_DALIV2EnableCurrentProtector [▸ 260]	The command enables the current protection device of the control gear.
FB_DALIV2QueryCurrentProtectorActive [▸ 261]	The system queries whether the current protection device is active.
FB_DALIV2QueryCurrentProtectorEnabled [▸ 262]	The system queries whether the current protection device is enabled.
FB_DALIV2QueryDimmingCurve [▸ 263]	The dimming curve of the control gear is read out.
FB_DALIV2QueryFastFadeTime [▸ 264]	Queries the value of FAST FADE TIME.

Name	Description
FB_DALIV2QueryGearType [▶ 265]	Queries the value of GEAR TYPE.
FB_DALIV2QueryLedFailureStatus [▶ 266]	Queries the value of FAILURE STATUS.
FB_DALIV2QueryLedFeatures [▶ 267]	Queries the value of FEATURES.
FB_DALIV2QueryLoadDecrease [▶ 268]	The system queries whether a significant decrease in load (compared to the reference power of the system) has been detected.
FB_DALIV2QueryLoadIncrease [▶ 269]	The system queries whether a significant increase in load (compared to the reference power of the system) has been detected.
FB_DALIV2QueryMinFastFadeTime [▶ 270]	Queries the value of MIN FAST FADE TIME.
FB_DALIV2QueryOpenCircuit [▶ 271]	The system queries whether an idle mode has been detected.
FB_DALIV2QueryOperatingMode [▶ 272]	Queries the value of OPERATING MODE.
FB_DALIV2QueryPossibleOperatingModes [▶ 273]	Queries the value of POSSIBLE OPERATING MODE.
FB_DALIV2QueryReferenceMeasurementFailed [▶ 274]	The system queries whether a started reference measurement has failed.
FB_DALIV2QueryReferenceRunning [▶ 275]	The system queries whether a reference measurement of the system performance is active.
FB_DALIV2QueryShortCircuit [▶ 276]	The system queries whether a short circuit has been detected.
FB_DALIV2QueryThermalOverload [▶ 277]	The system queries whether there is a thermal overload with reduction of the luminous flux.
FB_DALIV2QueryThermalShutDown [▶ 278]	The system queries whether a thermal shutdown occurred.
FB_DALIV2ReferenceSystemPower [▶ 279]	The control gear measures and stores the performance level of the system, in order to detect load increase and decrease.
FB_DALIV2SelectDimmingCurve [▶ 280]	The dimming curve of the control gear is selected.
FB_DALIV2SetFastFadeTime [▶ 281]	Sets the FAST FADE TIME in the control gear.

These commands belong to the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

5.1.5 Push button - function blocks

A detailed description of the individual DALI commands and the variables can be found in standard IEC 62386 Part 301.

Name	Description
FB_DALIV2x301QueryDoubleTimer [▶ 320]	Queries the value of the DOUBLE TIMER.
FB_DALIV2x301QueryDoubleTimerMin [▶ 321]	Queries the minimum value of the DOUBLE TIMER.
FB_DALIV2x301QueryRepeatTimer [▶ 322]	Queries the value of the REPEAT TIMER.
FB_DALIV2x301QueryShortTimer [▶ 323]	Queries the value of the SHORT TIMER.
FB_DALIV2x301QueryShortTimerMin [▶ 324]	Queries the minimum value of the SHORT TIMER.
FB_DALIV2x301QueryStuckTimer [▶ 325]	Queries the value of the STUCK TIMER.
FB_DALIV2x301SetDoubleTimer [▶ 327]	Sets the value of the DOUBLE TIMER.

Name	Description
FB_DALIV2x301SetRepeatTimer [▸ 328]	Sets the value of the REPEAT TIMER.
FB_DALIV2x301SetShortTimer [▸ 329]	Sets the value of the SHORT TIMER.
FB_DALIV2x301SetStuckTimer [▸ 330]	Sets the value of the STUCK TIMER.

5.1.6 Occupancy sensor function blocks

A detailed description of the individual DALI commands and the variables can be found in standard IEC 62386 Part 303.

Name	Description
FB_DALIV2x303CancelHoldTimer [▸ 331]	Terminates the HOLD TIMER prematurely.
FB_DALIV2x303CatchMovement [▸ 332]	After calling this command, an event is only sent once if a movement is detected.
FB_DALIV2x303QueryCatching [▸ 333]	Queries whether the system is waiting for the detection of movement.
FB_DALIV2x303QueryDeadtimeTimer [▸ 334]	Queries the value of the DEADTIME TIMER.
FB_DALIV2x303QueryHoldTimer [▸ 335]	Queries the value of the HOLD TIMER.
FB_DALIV2x303QueryReportTimer [▸ 336]	Queries the value of the REPORT TIMER.
FB_DALIV2x303SetDeadtimeTimer [▸ 338]	Sets the value of the DEADTIME TIMER.
FB_DALIV2x303SetHoldTimer [▸ 339]	Sets the value of the HOLD TIMER.
FB_DALIV2x303SetReportTimer [▸ 340]	Sets the value of the REPORT TIMER.

5.1.7 Interior Automation functions

The Interior Automation company has defined its own DALI commands for PIR sensors. These commands extend beyond the possible DALI commands in accordance with the IEC 62386 standard. Please contact Interior Automation for a more detailed description of the commands.

Name	Description
FB_DALIV2IAPIR [▸ 352]	This function block cyclically reads the status of an IA PIR sensor and scales the measured brightness and detected presence based on the received value.
FB_DALIV2IAPIRPhysicalIndicatorOff	Deactivates the red LED.
FB_DALIV2IAPIRPhysicalIndicatorOn	Activates the red LED.
FB_DALIV2IAPIRQueryExtendedVersion	Reads the software version number.
FB_DALIV2IAPIRQueryFlags	Reads the properties.
FB_DALIV2IAPIRQuerySensitivity	Reads the sensor sensitivity.
FB_DALIV2IAPIRQueryTimeout	Reads the time-out.
FB_DALIV2IAPIRStartIdentification	Makes the green LED flash for 10 seconds.
FB_DALIV2IAPIRStoreDTRAsFlags	Saves the data in the Data Transfer Register (DTR) as properties.
FB_DALIV2IAPIRStoreDTRAsSensitivity	Saves the data in the Data Transfer Register (DTR) as sensor sensitivity.
FB_DALIV2IAPIRStoreDTRAsTimeout	Saves the data in the Data Transfer Register (DTR) as time-out.

These commands belong to the application extended commands for DALI devices. These only work if they are preceded by the *Enable Device Type 128* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 128* command is internally placed automatically before all application extended commands for DALI devices.

5.1.8 Brightness sensors function blocks

A detailed description of the individual DALI commands and the variables can be found in standard IEC 62386 Part 304

Name	Description
FB_DALIV2x304QueryDeadtimeTimer [▸ 341]	Queries the value of the DEADTIME TIMER.
FB_DALIV2x304QueryHysteresis [▸ 342]	Queries the hysteresis value.
FB_DALIV2x304QueryHysteresisMin [▸ 343]	Queries the value for the minimum possible hysteresis.
FB_DALIV2x304QueryReportTimer [▸ 344]	Queries the value of the REPORT TIMER.
FB_DALIV2x304SetDeadtimeTimer [▸ 345]	Sets the value of the DEADTIME TIMER.
FB_DALIV2x304SetHysteresis [▸ 346]	Sets the hysteresis value.
FB_DALIV2x304SetHysteresisMin [▸ 348]	Sets the value for the minimum possible hysteresis.
FB_DALIV2x304SetReportTimer [▸ 349]	Sets the value of the REPORT TIMER.

5.1.9 B.E.G. function blocks

The company B.E.G. has defined its own special DALI commands. These commands extend beyond the possible DALI commands in accordance with the IEC 62386 standard. However, these commands are only applicable to certain B.E.G. devices. Please contact B.E.G. for a more detailed description of the commands. B. E. G. also offers the following services DALI multi-sensors, which have been developed according to IEC 62386 part 103, 303 and 304. The devices of this product family are marked with "DALI-2-BMS" in the product name.

Name	Description
FB_DALIV2xBEGLuxomat [▸ 353]	This function block evaluates the measured brightness and presence of the B.E.G. DALI control unit. This function block can also be used to initialize the DALI control units.

5.1.10 Osram function blocks

Osram has defined its own special DALI commands. These commands extend beyond the possible DALI commands in accordance with the IEC 62386 standard. However, these commands are only applicable to certain Osram devices. Please contact Osram for a more detailed description of the commands.

Name	Description
FB_DALIV2xOsramProfSensorCoupler [▸ 356]	This function block evaluates the measured brightness and presence of the Osram DALI Professional Sensor Coupler. This function block can also be used to initialize the DALI control units.
FB_DALIV2xOsramProfPushbuttonCoupler [▸ 355]	This function block evaluates the status of the digital input of the Osram DALI Professional Pushbutton Coupler. This function block can also be used to initialize the DALI control units.
FB_DALIV2xOsramDisableSignalMode	Disables the *Input Signal Mode* for a channel.
FB_DALIV2xOsramEnableSignalMode	Enables the *Input Signal Mode* for a channel.

Name	Description
FB_DALIV2xOsramQueryConfigurationId	Reading the configuration for a channel.
FB_DALIV2xOsramQueryInputDeviceType	Reading the device type.
FB_DALIV2xOsramQueryInputValue	Reading the input value of a channel.
FB_DALIV2xOsramQueryResolution	Queries the resolution of the input values of the control unit.
FB_DALIV2xOsramStoreConfigurationId	Writes the configuration for a channel.
FB_DALIV2xOsramIdentifySelectedDevice	Starts the identification routine for the selected control unit (random address and search address are the same).
FB_DALIV2xOsramQueryChannelSize	Reads the number of channels supported by the control unit.
FB_DALIV2xOsramQueryConfigurationFeature	Reads the possible configuration values for a channel.
FB_DALIV2xOsramQueryDeviceError	Queries whether the control unit has detected an error.
FB_DALIV2xOsramQueryStatus	Reads the device status.
FB_DALIV2xOsramReadMemoryLocation	A byte is read from the memory of the control unit.

5.1.11 Philips functions

The Philips company has defined its own DALI commands for special discharge lamps. These commands extend beyond the possible DALI commands in accordance with the standard IEC 62386 Part 203. However, these commands are only usable for certain Philips control gears. Please contact Philips for a more detailed description of the commands.

Name	Description
FB_DALIV2PhilipsChangePAEC	Activates or deactivates the "Application Extended Command Set".
FB_DALIV2PhilipsQueryCtrlGearOperationTime	Reads the execution time of the control gear.
FB_DALIV2PhilipsQueryCtrlGearOvertempLevel	Reads the overtemperature threshold value of the control gear.
FB_DALIV2PhilipsQueryCtrlGearOvertempTime	Reads the overtemperature time of the control gear.
FB_DALIV2PhilipsQueryCtrlGearTemperature	Reads the temperature of the control gear.
FB_DALIV2PhilipsQueryFailureStatus	Reads the failure status.
FB_DALIV2PhilipsQueryHIDLampLevel	Reads the lamp status.
FB_DALIV2PhilipsQueryHIDMaxFadeDownRate	Reads the maximum fade down rate.
FB_DALIV2PhilipsQueryHIDMaxFadeUpRate	Reads the maximum fade up rate.
FB_DALIV2PhilipsQueryLampType	Reads the lamp type.
FB_DALIV2PhilipsQueryLampVoltage	Reads the lamp voltage.
FB_DALIV2PhilipsQueryMainsVoltage	Reads the mains voltage.
FB_DALIV2PhilipsQueryTimeout	Reads the time-out.
FB_DALIV2PhilipsQueryUICByte	Reads a byte of the UIC.
FB_DALIV2PhilipsSetDTRAsSegmentAddress	Saves the data contained in the Data Transfer Register (DTR) to the Segment Address Register.
FB_DALIV2PhilipsSetTestMode	Sets the control gear to test mode.
FB_DALIV2PhilipsStoreDTRAsLampType	Saves the data in the Data Transfer Register (DTR) as lamp type.
FB_DALIV2PhilipsStoreDTRAsOvertempLevel	Saves the data in the Data Transfer Register (DTR) as overtemperature threshold value.
FB_DALIV2PhilipsStoreDTRAsSegmentAddress	Saves the data in the Data Transfer Register (DTR) as Segment Address Register.

Name	Description
FB_DALIV2PhilipsStoreDTRAsTimeout	Saves the data in the Data Transfer Register (DTR) as time-out.

These commands belong to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

5.1.12 Steinel - function blocks

The company Steinel has defined its own special DALI commands. These commands extend beyond the possible DALI commands in accordance with the IEC 62386 standard. However, these commands are only applicable to certain Steinel devices. Please contact Steinel for a more detailed description of the commands.

Name	Description
FB_DALIV2xSteinelLiveLinkMotionSensor [▶ 357]	This function block evaluates the measured brightness and presence of the Steinel LiveLink DALI control unit. This function block can also be used to initialize the DALI control units.
FB_DALIV2xSteinelSetBrightnessChangeLevel	Sets the *Brightness Change Level* value.
FB_DALIV2xSteinelSetBrightnessChangeTimer	Sets the time for the *Brightness Change Timer*.
FB_DALIV2xSteinelSetEventFilter	This function block sets the event filter for the respective control unit instance.
FB_DALIV2xSteinelSetMotionDetectionRange	Sets the size of the detection area.
FB_DALIV2xSteinelSetMotionTimer	Sets the time for the *Motion Timer*.
FB_DALIV2xSteinelSetMotionTimerRepeat	Sets the time for *Motion Timer Repeat*.
FB_DALIV2xSteinelSetMotionVerificationLevel	Sets the sensitivity for the motion sensor.
FB_DALIV2xSteinelSetSignalLedStatus	Switches the LEDs in the control unit.
FB_DALIV2xSteinelQueryBrightnessChangeLevel	Queries the *Brightness Change Level* value.
FB_DALIV2xSteinelQueryBrightnessChangeTimer	Queries the time for the *Brightness Change Timer*.
FB_DALIV2xSteinelQueryEventFilter	Queries the event filter for the respective control unit instance.
FB_DALIV2xSteinelQueryMotionDetectionRange	Queries the size of the detection area.
FB_DALIV2xSteinelQueryMotionTimer	Queries the time of the *Motion Timer*.
FB_DALIV2xSteinelQueryMotionTimerRepeat	Queries the time of the *Motion Timer Repeat*.
FB_DALIV2xSteinelQueryMotionVerificationLevel	Queries the sensitivity of the motion sensor.
FB_DALIV2xSteinelQuerySensorType	Queries the sensor type.
FB_DALIV2xSteinelQuerySignalLedStatus	Queries the LEDs in the control unit.

5.1.13 Tridonic - function blocks

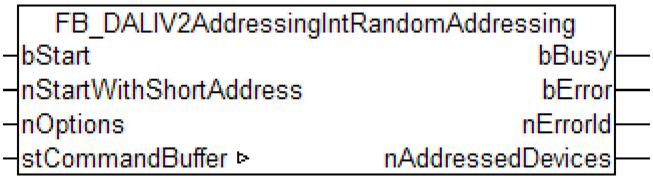
Name	Description
FB_DALIV2SmartSPOT [▶ 359]	This block cyclically reads the status of a smartSPOT sensor or MSensor 02 and scales the measured brightness and detected presence based on the received value.

5.1.14 Theben HTS - function blocks

The company Theben has defined its own special DALI commands. These commands extend beyond the possible DALI commands in accordance with the IEC 62386 standard. However, these commands are only applicable to certain Theben devices. Please contact Theben for a more detailed description of the commands.

Name	Description
FB_DALIV2xThebenPlanoSpot360 [▶ 361]	This function block evaluates the 3 measured brightness values and the presence of the ThebenHTS PlanoSpot DALI control unit. This function block can also be used to initialize the DALI control units.

5.1.15 FB_DALIV2AddressingIntRandomAddressing



This function block addresses the ballasts in random order. The user has no influence over which short address is assigned to which ballast. Short addresses are allocated in ascending order.

Applying a rising edge to the *bStart* input initiates the block, and the *bBusy* output goes TRUE. Depending on the selected options (parameter *nOptions*) the group associations and scenarios are subsequently deleted. The terminal now addresses all ballasts independently. Once all ballasts have been addressed, the *bBusy* output switches back to FALSE. The *nAddressedDevices* output variable supplies information about how many ballasts have received a short address. Processing this function block can take several minutes, depending on how many ballasts are attached. Since the addressing is performed directly by the terminal, this method is somewhat faster than the FB_DALIV2AddressingRandomAddressing() [▶ 45] function block. However, this function block does not supply any feedback during addressing. In addition to that, addressing cannot be terminated prematurely.

This function block can only be executed if the terminal has the firmware version 2A or newer.

VAR_INPUT

```
bStart                    : BOOL;
nStartWithShortAddress    : BYTE;
nOptions                  : DWORD := DALIV2_OPTION_OPTICAL_FEEDBACK;
```

bStart: A rising edge at this input activates the block, thereby starting the addressing sequence.

nStartWithShortAddress: Short address allocated to the first ballast (0 ... 63).

nOptions: Options for addressing the ballasts (see table). The individual constants must be linked with OR operators.

Constant	Description
DALIV2_OPTION_COMPLETE_NEW_INSTALLATION	All ballasts are re-addressed, including ballasts that already have a short address.
DALIV2_OPTION_DELETE_ALL_GROUP_ASSIGNMENTS	Prior to addressing, the group associations are deleted for any ballasts, even those which may not be addressed by the addressing method (see variables GROUP 0-7 and GROUP 8-15 [▶ 366])

Constant	Description
DALIV2_OPTION_DELETE_ALL_SCENE_ASSIGNMENTS	Prior to addressing, the scenes are deleted for any ballasts, even those which may not be addressed by the addressing method (see variables SCENE 0 to SCENE 15 [▸ 366])
DALIV2_OPTION_OPTICAL_FEEDBACK	Prior to addressing, all ballasts are set to MIN LEVEL [▸ 364] Newly addressed ballasts are assigned MAX LEVEL

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nAddressedDevices  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

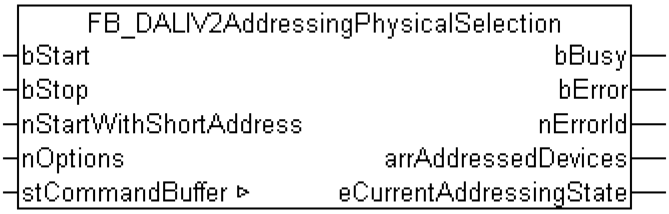
nAddressedDevices: If addressing has been completed (bBusy *is FALSE),* then the number of addressed ballasts is shown at this output.

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.16 FB_DALIV2AddressingPhysicalSelection



This function block addresses the ballasts through 'physical selection' based on the addressing technique. This means that the individual ballasts are selected (and therefore addressed) by removing the lamps. Short addresses are allocated (ascending) in the same order in which the lamps are removed.

Applying a rising edge to the *bStart* input starts the block, and the *bBusy* output goes TRUE. Depending on the selected options (parameter *nOptions*) the group associations and scenarios are subsequently deleted. The *eCurrentAddressingState* output specifies the next required user operation. It determines whether for the next ballast the lamp should be removed or reinserted. The *arrAddressedDevices* output variable provides information about which ballasts have already been allocated a short address. Once all ballasts have been addressed, the addressing procedure is completed through a rising edge at input *bStop*, and the *bBusy* output switches back to FALSE.

VAR_INPUT

```
bStart                  : BOOL;
bStop                   : BOOL;
nStartWithShortAddress  : BYTE;
nOptions                : DWORD := DALIV2_OPTION_OPTICAL_FEEDBACK;
```

bStart: A rising edge at this input activates the block, thereby starting the addressing sequence.

bStop: A rising edge at this input deactivates the block, thereby stopping the addressing sequence.

nStartWithShortAddress: Short address allocated to the first ballast (0 ... 63).

nOptions: Options for addressing the ballasts (see table). The individual constants must be linked with OR operators.

Constants	Description
DALIV2_OPTION_COMPLETE_NEW_INSTALLATION	All ballasts are re-addressed, including ballasts that already have a short address.
DALIV2_OPTION_DELETE_ALL_GROUP_ASSIGNMENTS	Prior to addressing, the group associations are deleted for any ballasts, even those which may not be addressed by the addressing method (see variables GROUP 0-7 and GROUP 8-15 [▶ 366])
DALIV2_OPTION_DELETE_ALL_SCENE_ASSIGNMENTS	Prior to addressing, the scenes are deleted for any ballasts, even those which may not be addressed by the addressing method (see variables SCENE 0 to SCENE 15 [▶ 366])
DALIV2_OPTION_WITHOUT_RANDOMISE	Prior to addressing, all ballasts are set to MIN LEVEL [▶ 364] Newly addressed ballasts are assigned MAX LEVEL

VAR_OUTPUT

```
bBusy                       : BOOL;
bError                      : BOOL;
nErrorId                    : UDINT;
arrAddressedDevices         : ARRAY[0..63] OF BOOL;
eCurrentAddressingState     : E_DALIV2CurrentAddressingState;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

arrAddressedDevices: Once a short address is assigned to a ballast, the associated element is set in the array. The array index reflects the short address of the ballast.

eCurrentAddressingState: The output variable indicates the current step (see table).

Element	Description
eDALIV2AddrStateIdle	No addressing takes place.
eDALIV2AddrStateRemoveLamp	The block waits for a lamp to be removed at a ballast.
eDALIV2AddrStateReinsertLamp	The block has detected the ballast on which the lamp was removed (the ballast is selected) and now waits for it to be inserted again.
eDALIV2AddrStateAddressingLamp	The selected ballast is addressed.

VAR_IN_OUT

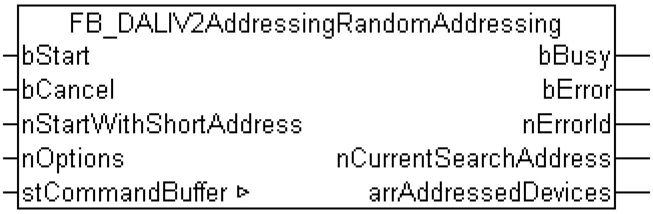
```
stCommandBuffer       : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Also see about this

- E_DALIV2CurrentAddressingState [▸ 381]

5.1.17 FB_DALIV2AddressingRandomAddressing



This function block addresses the ballasts in random order. The user has no influence over which short address is assigned to which ballast. Short addresses are allocated in ascending order.

Applying a rising edge to the *bStart* input starts the block, and the *bBusy* output goes TRUE. Depending on the selected options (parameter *nOptions*) the group associations and scenarios are subsequently deleted. The block now addresses all ballasts independently. The *arrAddressedDevices* output variable provides information about which ballasts have already been allocated a short address. Once all ballasts have been addressed, the *bBusy* output switches back to FALSE. Addressing can be aborted through a rising edge at input *bCancel*. Processing this function block can take several minutes, depending on how many ballasts are attached.

VAR_INPUT

```
bStart                    : BOOL;
bCancel                   : BOOL;
nStartWithShortAddress    : BYTE;
nOptions                  : DWORD := DALIV2_OPTION_OPTICAL_FEEDBACK;
```

bStart: A rising edge at this input activates the block, thereby starting the addressing sequence.

bCancel: A rising edge at this input deactivates the block, thereby interrupting the addressing sequence.

nStartWithShortAddress: Short address allocated to the first ballast (0 ... 63).

nOptions: Options for addressing the ballasts (see table). The individual constants must be linked with OR operators.

Constant	Description
DALIV2_OPTION_COMPLETE_NEW_INSTALLATION	All ballasts are re-addressed, including ballasts that already have a short address.
DALIV2_OPTION_DELETE_ALL_GROUP_ASSIGNMENTS	Prior to addressing, the group associations are deleted for any ballasts, even those which may not be addressed by the addressing method (see variables GROUP 0-7 and GROUP 8-15 [▸ 366])
DALIV2_OPTION_DELETE_ALL_SCENE_ASSIGNMENTS	Prior to addressing, the scenes are deleted for any ballasts, even those which may not be addressed by the addressing method (see variables SCENE 0 to SCENE 15 [▸ 366])
DALIV2_OPTION_OPTICAL_FEEDBACK	Prior to addressing, all ballasts are set to MIN LEVEL [▸ 364] Newly addressed ballasts are assigned MAX LEVEL
DALIV2_OPTION_WITHOUT_RANDOMISE	The RANDOMISE command is not called before the addressing sequence. This means that all ballasts retain their existing random address (RANDOM ADDRESS [▸ 365])

VAR_OUTPUT

```
bBusy                    : BOOL;
bError                   : BOOL;
nErrorId                 : UDINT;
nCurrentSearchAddress    : UDINT;
arrAddressedDevices      : ARRAY[0..63] OF BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

nCurrentSearchAddress: Current search address (SEARCH ADDRESS [▸ 365]).

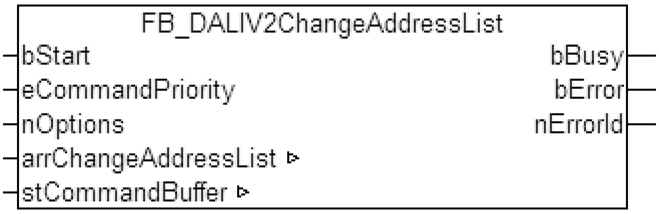
arrAddressedDevices: Once a short address is assigned to a ballast, the associated element is set in the array. The array index reflects the short address of the ballast.

VAR_IN_OUT

```
stCommandBuffer       : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.18 FB_DALIV2ChangeAddressList



The short addresses of several ballasts can be changed using this function block. As opposed to the *FB_DALIV2SwapShortAddressList* block, it is not necessary for a free, unused short address to be present in the DALI line.

A list of the ballasts whose short addresses are to be changed is transferred in the *arrChangeAddressList* array of type ST_DALIV2ChangeAddressList [▸ 382]. The list has 64 entries from 0 to 63. Each entry contains a variable *nOldAddress* and *nNewAddress* with which the address assignment is parameterised. The list end 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 block is started (rising edge on *bStart*), the list end is first determined on the basis of the above-described entry and afterwards the valid list range is examined for the following false entries:

- Address entries > 63
- Double address entry on the original page *nOldAddress* (would not make sense)
- Double address entry on the target page *nNewAddress* (leads to double assignment of an address and, hence, to errors)

The block then determines the internal long addresses of the DALI devices on the basis of the short addresses and enters them respectively to the parameters *nRandomAddressHigh, nRandomAddressMiddle* and *nRandomAddressLow* in the list. If an error occurs during this query, this leads to a false entry for the respective device in the list element *nErrors* (see ST_DALIV2ChangeAddressList [▸ 382]). The further sequence in the block now depends on the option DALIV2_OPTION_SAFE_ADDRESSING (nOptions input).

If it is set, then safe new addressing takes place: first of all, all short addresses of the selected DALI devices are deleted. Afterwards, status queries are sent to all desired new addresses in the DALI line. 2 cases are now possible:

- If a device responds to this query, then this desired new address is already otherwise assigned. The previously “deleted” DALI devices are programmed with their old addresses and an error message is output.
- If no devices respond to this status query, then the previously “deleted” DALI devices are programmed with the desired new addresses.

The reprogramming is checked afterwards in both cases. If an error occurs during deletion, during the status query or during the reprogramming, this leads to a false entry for the respective device in the list element *nErrors* (see ST_DALIV2ChangeAddressList [▶ 382]).
If the option DALIV2_OPTION_SAFE_ADDRESSING (*nOptions* input) is **not** set, then the deletion of the short addresses and the status query for the presence of desired new addresses are omitted and the new addresses are programmed directly. This is possible because programming takes place via the long address determined beforehand. Reprogramming is not verified in this case.

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

Bit	Error
0	Error whilst reading the high byte of the long address (*nRandomAddressHigh*).
1	Error whilst reading the middle byte of the long address (*nRandomAddressMiddle*).
2	Error whilst reading the low byte of the long address (*nRandomAddressLow*).
3	Error whilst deleting a short address.
4	Error whilst verifying a short address.
5	Error whilst programming a short address.

VAR_INPUT

```
bStart              : BOOL;
bCancel             : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityHigh;
nOptions            : DWORD := 0;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nOptions: Options for writing the variables (see table). The individual constants must be linked with OR operators.

Constant	Description
DALIV2_OPTION_SAFE_ADDRESSING	Safe addressing: Old short addresses are deleted, the new ones are checked to see if they already exist and reprogramming is verified.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

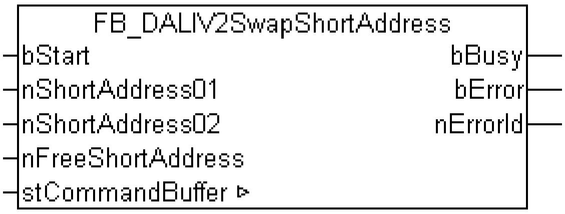
VAR_IN_OUT

```
arrChangeAddressList  : ARRAY[0.. 63] OF ST_DALIV2ChangeAddressList;
stCommandBuffer       : ST_DALIV2CommandBuffer;
```

arrChangeAddressList: A reference to the list [▸ 382] containing the short addresses to be changed.

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.19 FB_DALIV2SwapShortAddress



The short addresses of two ballasts can be swapped using this function block. In order to do this, however, it is necessary that a free, unused short address is present in the DALI line.

VAR_INPUT

```
bStart               : BOOL;
nShortAddress01      : BYTE;
nShortAddress02      : BYTE;
nFreeShortAddress    : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nShortAddress01: Short address of the first ballast (0 – 63).

nShortAddress02: Short address of the second ballast (0 – 63).

nFreeShortAddress: Free short address (0 – 63).

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

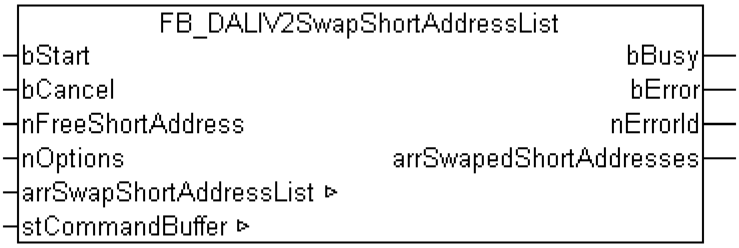
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.20 FB_DALIV2SwapShortAddressList



The short addresses of several ballasts can be swapped using this function block. In order to do this, however, it is necessary that a free, unused short address is present in the DALI line.

In the parameter *arrSwapShortAddressList*, a list of the ballasts whose short addresses are to be changed is transferred. The index of the structure thereby corresponds to the short address of the ballast. The element *nNewShortAddress* contains the new short address. *bShortAddressValid* must test to TRUE in order that the short address of the respective ballast will be changed. Applying a rising edge to the *bStart* input starts the block, and the *bBusy* output goes TRUE. Depending on the chosen options (*nOptions* parameter), all lamps will be set to the value MIN LEVEL. The elements of the output *arrSwapedShortAddresses* are reset. If the new short address is set for a ballast, the corresponding element in the output *arrSwapedShortAddresses* is set to TRUE. If the option DALIV2_OPTION_OPTICAL_FEEDBACK is active, the lamp will in addition be set to the value MAX LEVEL.

VAR_INPUT

```
bStart                : BOOL;
bCancel                : BOOL;
nFreeShortAddress      : BYTE;
nOptions               : DWORD := DALIV2_OPTION_OPTICAL_FEEDBACK;
```

bStart: The block is activated by a rising edge at this input.

bCancel: A rising edge at this input will deactivate the block and hence abort the swapping of the short addresses.

nFreeShortAddress: Free short address (0 – 63).

nOptions: Options for swapping short addresses (see table). The individual constants must be linked with OR operators.

Constants	Description
DALIV2_OPTION_OPTICAL_FEEDBACK	Before swapping the short addresses, all ballasts are set to MIN LEVEL [▶ 364]. After assigning the new short address, the brightness of the respective ballast will be changed to MAX LEVEL

VAR_OUTPUT

```
bBusy                      : BOOL;
bError                     : BOOL;
nErrorId                   : UDINT;
arrSwapedShortAddresses    : ARRAY[0..63] OF BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

arrSwapedShortAddresses: If the new short address has been set for a ballast, the corresponding element will be set in the array. The array index reflects the short address of the ballast.

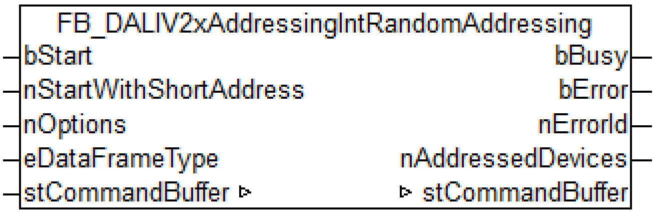
VAR_IN_OUT

```
arrSwapShortAddressList  : ARRAY[0.. 63] OF ST_DALIV2SwapShortAddressList;
stCommandBuffer          : ST_DALIV2CommandBuffer;
```

arrSwapShortAddressList: A reference to the list [▶ 384] containing the short addresses to be swapped.

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.21 FB_DALIV2xAddressingIntRandomAddressing



This function block addresses the control units (sensors) at random. The user has no influence on which control unit 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 control units independently. If all control units are addressed, the *bBusy* output goes back to FALSE. The output variable *nAddressedDevices* provides information on how many control units were assigned a short address. Depending on the number of connected control units, processing of this function block can take several minutes.

VAR_INPUT

```
bStart                  : BOOL;
nStartWithShortAddress  : BYTE := 0;
nOptions                : DWORD := DALIV2_OPTION_OPTICAL_FEEDBACK;
eDataFrameType          : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: A rising edge at this input activates the function block, thereby starting the addressing sequence.

nStartWithShortAddress: Short address assigned to the first control unit (0... 63).

nOptions: Options for addressing control units (see table). The individual constants must be linked with OR operators.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

Constant	Description
DALIV2_OPTION_COMPLETE_NEW_INSTALLATIO N	All control units are redirected, even those that already have a short address.
DALIV2_OPTION_OPTICAL_FEEDBACK	Before addressing, all devices are set to MIN LEVEL [▶ 364], MAX LEVEL brightness after allocation of the short address.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nAddressedDevices  : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nAddressedDevices: When addressing is completed (*bBusy* is FALSE), the number of addressed control units is displayed at this output.

VAR_IN_OUT

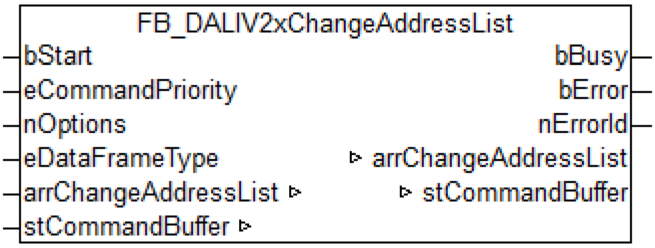
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.22 FB_DALIV2xChangeAddressList



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

A list of the control units for which the short address is to be changed is transferred in the array *arrChangeAddressList* of type ST_DALIV2ChangeAddressList [▶ 382]. 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 list end 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 (rising edge on *bStart*), the list end is first determined based on the above-described entry and afterwards the valid list range is examined for the following false entries:

- Address entries > 63
- Double address entry on the original page *nOldAddress* (would not make sense)
- Double address entry on the target page *nNewAddress* (leads to double assignment of an address and, hence, to errors)

The function block then determines the internal long addresses of the DALI devices on the basis of the short addresses and enters them respectively to the parameters *nRandomAddressHigh, nRandomAddressMiddle* and *nRandomAddressLow* in the list. If an error occurs during this query, this leads to a false entry for the respective device in the list element *nErrors* (see ST_DALIV2ChangeAddressList [▶ 382]). The further sequence in the function block now depends on the option DALIV2_OPTION_SAFE_ADDRESSING (*nOptions* input). If it is set, then safe new addressing takes place: first of all, all short addresses of the selected DALI devices are deleted. Afterwards, status queries are sent to all desired new addresses in the DALI line. 2 cases are now possible:

- If a device responds to this query, then this desired new address is already otherwise assigned. The previously “deleted” DALI devices are programmed with their old addresses and an error message is output.

- If no devices respond to this status query, then the previously “deleted” DALI devices are programmed with the desired new addresses.

The reprogramming is checked afterwards in both cases. If an error occurs during deletion, during the status query or during the reprogramming, this leads to a false entry for the respective device in the list element *nErrors* (see ST_DALIV2ChangeAddressList [▸ 382]).

If the option DALIV2_OPTION_SAFE_ADDRESSING (*nOptions* input) is **not** set, then the deletion of the short addresses and the status query for the presence of desired new addresses are omitted and the new addresses are programmed directly. This is possible because programming takes place via the long address determined beforehand. Reprogramming is not verified in this case.

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

Bit	Error
0	Error whilst reading the high byte of the long address (*nRandomAddressHigh*).
1	Error whilst reading the middle byte of the long address (*nRandomAddressMiddle*).
2	Error whilst reading the low byte of the long address (*nRandomAddressLow*).
3	Error whilst deleting a short address.
4	Error whilst verifying a short address.
5	Error whilst programming a short address.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityHigh;
nOptions           : DWORD := 0;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▸ 381] (high, medium, or low) with which the command is processed by the library.

nOptions: Options for writing the variables (see table). The individual constants must be linked with OR operators.

eDataFrameType: Output format [▸ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

Constant	Description
DALIV2_OPTION_SAFE_ADDRESSING	Safe addressing: Old short addresses are deleted, the new ones are checked to see if they already exist and reprogramming is verified.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
arrChangeAddressList   : ARRAY [0..63] OF ST_DALIV2ChangeAddressList;
stCommandBuffer        : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

arrChangeAddressList: A reference to the list containing the short addresses to be changed.

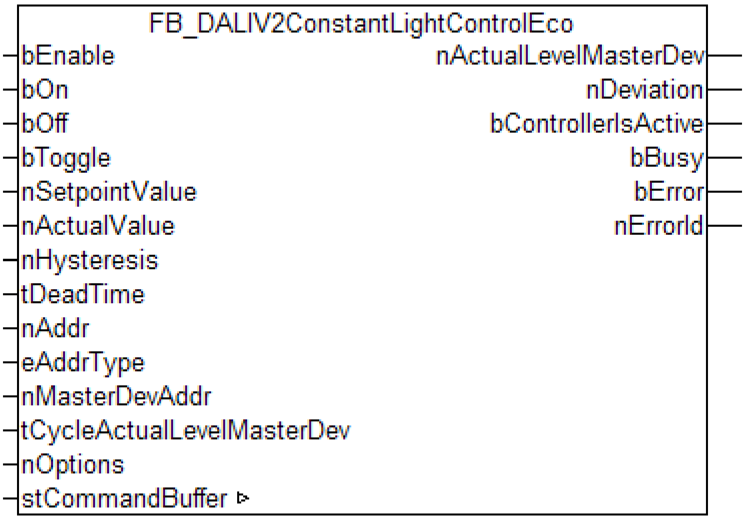
Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

Also see about this

- ST_DALIV2ChangeAddressList [▶ 382]

5.1.23 FB_DALIV2ConstantLightControlEco



The block FB_DALIV2ConstantLightControlEco() is used for constant light control with DALI ballasts.

The system tries to match a specified set value through cyclic dimming. The control dynamics are determined by a dead time (*tDeadTime*). The dead time defines the delay between the individual commands for changing the control value. The smaller the dead time, the faster the control. A freely definable hysteresis (*nHysteresis*) prevents continuous oscillation around the set value. If the actual value is within the hysteresis range around the set value, the lamps brightness remains unchanged. An option is available for specifying whether the lamps should be switched on and off automatically (see table below).

Parameter *nMasterDevAddr*

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.

VAR_INPUT

```
bEnable                    : BOOL := TRUE;
bOn                        : BOOL;
bOff                       : BOOL;
bToggle                    : BOOL;
nSetpointValue             : UINT := 500;
nActualValue               : UINT;
nHysteresis                : UINT := 50;
```

```
tDeadTime                   : TIME := t#10s;
nAddr                       : BYTE := 0;
eAddrType                   : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr              : BYTE := 0;
tCycleActualLevelMasterDev : TIME := t#0s;
nOptions                    : DWORD := 0;
```

bEnable: Enables the block. If this input is FALSE, the inputs *bOn*, *bOff* and *bToogle* are disabled. No control values are output.

bOn: Switches the addressed devices to MAX_LEVEL [▸ 364] and activates constant light control.

bOff: Switches the addressed devices off and disables constant light control.

bToggle: The lighting is switched on or off, depending on the state of the reference device.

nSetpointValue: This input is used for specifying the set value.

nActualValue: The actual value is applied at this input.

nHysteresis: Control hysteresis around the set value. If the actual value is within this range, the control values for the lamps remain unchanged.

tDeadTime: Dead time between the individual commands used for changing the control value for the DALI lamps.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL_DIM_LEVEL [▸ 364]) in the background. So that the dimming of the lamps is not disturbed, reading always has the lowest priority. If the value is set to 0, reading is prohibited.

nOptions: Options (see table). The individual constants must be linked with OR operators.

Constants	Description
DALIV2_OPTION_SWITCH_ON_AND_OFF	The DALI commands ON_AND_STEP_UP [▸ 123] and STEP_DOWN_AND_OFF [▸ 127] are used for changing the control value. This causes the lamps to be switched off when MIN_LEVEL STEP_UP [▸ 127] and STEP_DOWN [▸ 126] are used. In this cases the lamps remain switched on continuously.
DALIV2_OPTION_SWITCH_ON_WITH_MIN_LEVEL	If the light is switched on again by the constant light control, this option always uses the command MIN_LEVEL [▸ 364]. If the option is not set, MAX_LEVEL [▸ 364] is used. This option is available from V2.9.3 of the PLC library.

VAR_OUTPUT

```
nActualLevelMasterDev: BYTE;
nDeviation           : INT;
bControllerIsActive  : BOOL;
bBusy                : BOOL;
bError               : BOOL;
nErrorId             : UDINT;
```

nActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

nDeviation: Current control deviation (set value/actual value).

bControllerIsActive: This output is set once the control is activated.

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

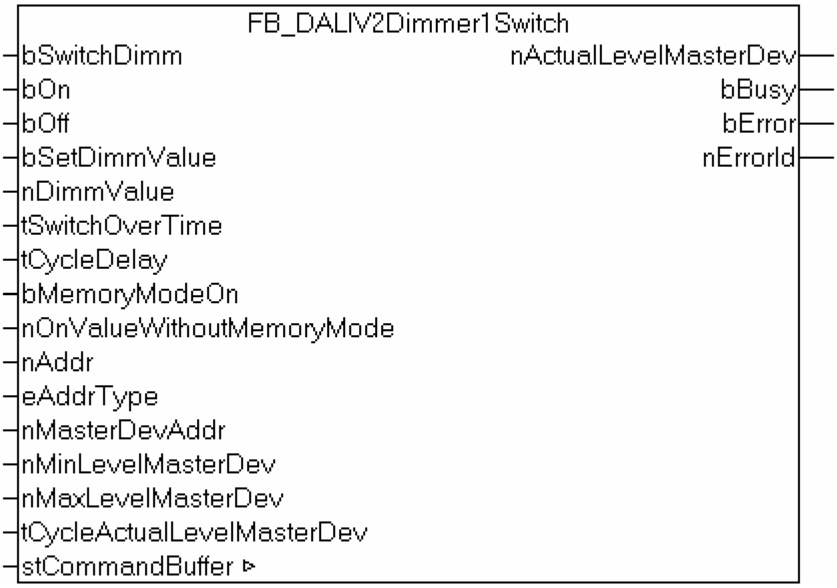
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer          : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.24 FB_DALIV2Dimmer1Switch



An individual DALI lamp, a DALI group or a complete DALI line can be switched and dimmed using a single switch through this block.

Operating by means of the bSwitchDimm input

The light is switched on or off by a short signal at the *bSwitchDimm* input. Dimmer mode will be activated if the signal remains for longer than *tSwitchOverTime* (typical recommended value: 200 ms). The output signal then cycles between *nMinLevelMasterDev* and *nMaxLevelMasterDev*. In order to be able to set the maximum or minimum value more easily, the output signal pauses at the level of the maximum and minimum values for the time given by *tCycleDelay*. When the signal is once more removed, the output signal being generated at that time is retained. Another pulse at the input will set the output to 0. If the *bSwitchDimm* signal is briefly removed in dimmer mode, the function block changes its dimming direction.

Operation by means of the bOn and bOff inputs

If a positive edge is applied to the *bOn* or *bOff* inputs, the light is switched on or off directly. The output value is set to 0 when switching off. The switch-on behavior can be affected by the memory function (see below).

Operation by means of the bSetDimmValue and nDimmValue inputs

If the value *nDimmValue* changes, the addressed devices will be switched directly to this brightness value. The significant point here is that the value changes. The lighting is switched off by changing the value to 0. If there is a positive edge at the *bSetDimmValue* input, the value of *nDimmValue* immediately appears at the output. Immediate modification of the output can be suppressed by a static 1-signal at the *bSetDimmValue* input. This allows a value to be applied to the *nDimmValue* input, which is not passed to the output until the next positive edge of *bSetDimmValue*.

The *bSetDimmValue* and *nDimmValue* inputs can be used to implement a variety of lighting scenarios. Direct setting of the output, by means of *nDimmValue*, can be used to achieve particular brightness levels. Either directly or by continuously changing the value. *nDimmValue* must have a value between *nMinLevelMasterDev* and *nMaxLevelMasterDev*. The value 0 is an exception. If the value is outside this range, the output value is limited to the upper or lower limit, as appropriate.

The memory function

It is necessary to determine whether the memory function (*bMemoryModeOn* input) is active or not at switch-on. If the memory function is active, then the last set value is adopted as the brightness value as soon as the device is switched on. If the memory function is not active, a brightness specified by the *nOnValueWithoutMemoryMode* parameter is assigned to the devices concerned. It is irrelevant, in this case, whether the light it has been switched on by means of the *bOn* input or the *bSwitchDimm* input. It should be noted that the *nOnValueWithoutMemoryMode* parameter must lie between *nMinLevelMasterDev* and *nMaxLevelMasterDev*. If this is not the case, the output value is adjusted to the upper or lower limit, as appropriate.

Parameter tSwitchOverTime

If a duration of 0 is specified for the parameter *tSwitchOverTime*, the *bSwitchDimm* input can only be used to dim the light. Switching on and off is only possible with the *bOn* and *bOff* inputs.

Parameter nMasterDevAddr

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.

VAR_INPUT

```
bSwitchDimm                  : BOOL;
bOn                          : BOOL;
bOff                         : BOOL;
bSetDimmValue                : BOOL;
nDimmValue                   : BYTE;
tSwitchOverTime              : TIME := t#400ms;
tCycleDelay                  : TIME := t#500ms;
bMemoryModeOn                : BOOL := FALSE;
nOnValueWithoutMemoryMode    : BYTE := 254;
nAddr                        : BYTE := 0;
eAddrType                    : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr               : BYTE := 0;
nMinLevelMasterDev           : BYTE := 126;
nMaxLevelMasterDev           : BYTE := 254;
tCycleActualLevelMasterDev : TIME := t#0s;
```

bSwitchDimm: Switches or dims the addressed devices.

bOn: Switches the addressed devices to the last output value, or to the value specified by *nOnValueWithoutMemoryMode*.

bOff: Switches the addressed devices off (value 0).

bSetDimmValue: A positive edge at this input sets the addressed devices immediately to the brightness value that is asserted at the *nDimmValue* input. If the value of *nDimmValue* changes, the brightness value is set immediately to the changed value if the *bSetDimmValue* input is FALSE.

nDimmValue: see *bSetDimmValue.*

tSwitchOverTime: Time for switching between the light on/off and dimming functions for the *bSwitchDimm* input.

tCycleDelay: Delay time, if either the minimum or maximum value is reached.

bMemoryModeOn: Switches over to use the memory function, so that the previous value is written to the output as soon as it is switched on.

nOnValueWithoutMemoryMode: Value at switch on if the memory function is not active.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

nMinLevelMasterDev: The minimum value of the master device.

nMaxLevelMasterDev: The maximum value of the master device.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▸ 364]) in the background. So that the dimming of the lamps is not disturbed, reading always has the lowest priority. If the value is set to 0, reading is prohibited.

VAR_OUTPUT

```
nActualLevelMasterDev  : BYTE;
bBusy                  : BOOL;
bError                 : BOOL;
nErrorId               : UDINT;
```

nActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

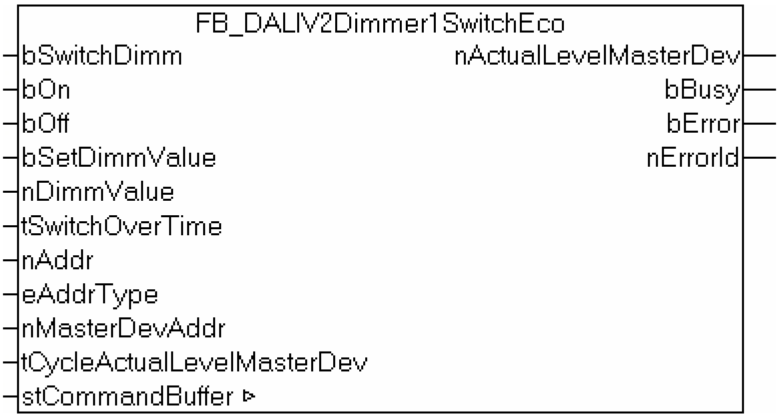
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer        : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.25 FB_DALIV2Dimmer1SwitchEco



The FB_DALIDimmer1SwitchEco() block is a variant of FB_DALIV2Dimmer1Switch() [▶ 55] that saves memory space. It does not include the special "Switch memory function off" function.

Operating by means of the *bSwitchDimm* input

The light is switched on or off by a short signal at the *bSwitchDimm* input. If the signal remains for longer than *tSwitchOverTime* (recommended value: 200 ms), dimmer mode is activated and the brightness increases or decreases steadily. The dimming direction is changed by briefly removing the *bSwitchDimm* signal.

Operation by means of the *bOn* and *bOff* inputs

The light is immediately switched on or off if a rising edge is applied to the *bOn* or *bOff* inputs. The output value is set to 0 when switching off.

Operation by means of the *bSetDimmValue* and *nDimmValue* inputs

If the value of *nDimmValue* changes the devices concerned will be switched to this brightness value immediately. The significant point here is that the value changes. The lighting is switched off by changing the value to 0. If there is a rising edge at the *bSetDimmValue* input, the value of *nDimmValue* immediately appears at the output. Immediate modification of the output can be suppressed by a static 1- signal at the *bSetDimmValue* input. This makes it possible to apply a value to the *nDimmValue* input, but for this value only to be passed to the output at the next rising edge of *bSetDimmValue*.

The *bSetDimmValue* and *nDimmValue* inputs can be used to implement a variety of lighting scenarios.Using *nDimmValue* to set the outputs directly can be used to achieve particular brightness levels, either directly or by continuously changing the value.

The memory function

In contrast to FB_DALIV2Dimmer1Switch() [▶ 55], where the memory function can be switched on or off through the *bMemoryModeOn* input, the memory function is always active on this memory-saving version. This means that the most recently set value is adopted for the brightness when switching on. It is irrelevant, in this case, whether the light it has been switched on by means of the *bOn* input or the *bSwitchDimm* input.

Parameter *tSwitchOverTime*

If a duration of 0 is specified for the parameter *tSwitchOverTime*, the *bSwitchDimm* input can only be used to dim the light. Switching on and off is only possible with the *bOn* and *bOff* inputs.

Parameter *nMasterDevAddr*

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.

VAR_INPUT

```
bSwitchDimm                : BOOL;
bOn                        : BOOL;
bOff                       : BOOL;
bSetDimmValue              : BOOL;
nDimmValue                 : BYTE;
tSwitchOverTime            : TIME := t#400ms;
nAddr                      : BYTE := 0;
eAddrType                  : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr             : BYTE := 0;
tCycleActualLevelMasterDev : TIME := t#0s;
```

bSwitchDimm: Switches or dims the addressed devices.

bOn: Switches the addressed devices to the most recent output value.

bOff: Switches the addressed devices off (value 0).

bSetDimmValue: A positive edge at this input sets the addressed devices immediately to the brightness value that is asserted at the *nDimmValue* input. If the value of *nDimmValue* changes, the brightness value is set immediately to the changed value if the *bSetDimmValue* input is FALSE.

nDimmValue: see *bSetDimmValue.*

tSwitchOverTime: Time for switching between the light on/off and dimming functions for the *bSwitchDimm* input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▶ 364]) in the background. So that the dimming of the lamps is not disturbed, reading always has the lowest priority. If the value is set to 0, reading is prohibited.

VAR_OUTPUT

```
nActualLevelMasterDev    : BYTE;
bBusy                    : BOOL;
bError                   : BOOL;
nErrorId                 : UDINT;
```

nActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

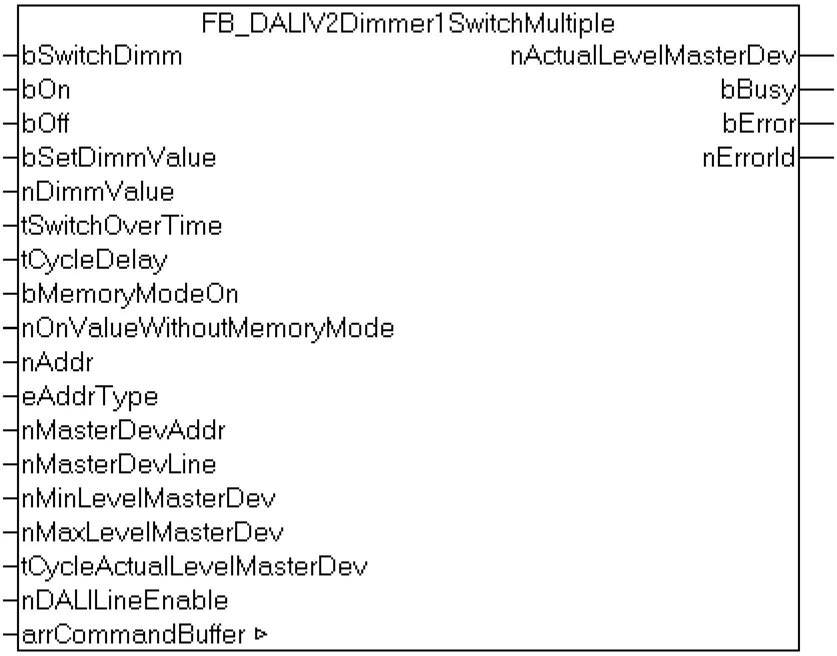
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer          : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.26 FB_DALIV2Dimmer1SwitchMultiple



Function block for switching and dimming DALI devices with one button. For applications in which up to five DALI lines (0..4) can be installed. The basic function of this function block can be found in the description of FB_DALIV2Dimmer1Switch() [▶ 55].

VAR_INPUT

```
bSwitchDimm                 : BOOL;
bOn                         : BOOL;
bOff                        : BOOL;
bSetDimmValue               : BOOL;
nDimmValue                  : BYTE;
tSwitchOverTime             : TIME := t#400ms;
tCycleDelay                 : TIME := t#500ms;
bMemoryModeOn               : BOOL := FALSE;
nOnValueWithoutMemoryMode   : BYTE := 254;
nAddr                       : BYTE := 0;
eAddrType                   : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr              : BYTE := 0;
nMasterDevLine              : BYTE := 0;
nMinLevelMasterDev          : BYTE := 126;
nMaxLevelMasterDev          : BYTE := 254;
tCycleActualLevelMasterDev  : TIME := t#0s;
nDALILineEnable             : BYTE := 2#0000_0001;
```

bSwitchDimm: Switches or dims the addressed devices on all the activated DALI lines.

bOn: Switches the addressed devices on all the activated DALI lines to the last output value, or to the value specified by *nOnValueWithoutMemoryMode*.

bOff: Switches the addressed devices on all the activated DALI lines off (value 0).

bSetDimmValue: A positive edge at this input sets the addressed devices on all the activated DALI lines immediately to the brightness value that is asserted at the *nDimmValue* input. If the value of *nDimmValue* changes, the brightness value is set immediately to the changed value if the *bSetDimmValue* input is FALSE.

nDimmValue: see *bSetDimmValue.*

tSwitchOverTime: Time for switching between the light on/off and dimming functions for the *bSwitchDimm* input.

tCycleDelay: Delay time, if either the minimum or maximum value is reached.

bMemoryModeOn: Switches over to use the memory function, so that the previous value is written to the output as soon as it is switched on.

nOnValueWithoutMemoryMode: Value at switch on if the memory function is not active.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations (0 - 63).

nMasterDevLine: The line on which the master device for group and common configurations is located. Depending on the function block, there is only one master device that is to be selected from one of the activated DALI lines. The lines are numbered from 0 to 4.

nMinLevelMasterDev: The minimum value of the master device.

nMaxLevelMasterDev: The maximum value of the master device.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▶ 364]) in the background. So that the dimming of the lamps is not disturbed, reading always has the lowest priority. If the value is set to 0, reading is prohibited.

nDALILineEnable: Input variable in the form of a bit-pattern. A 1 in the bit-pattern indicates that the DALI line is active. **Example: 2#01001** indicates that DALI lines **0** and **3** are active.

VAR_OUTPUT

```
nActualLevelMasterDev   : BYTE;
bBusy                   : BOOL;
bError                  : BOOL;
nErrorId                : UDINT;
```

nActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

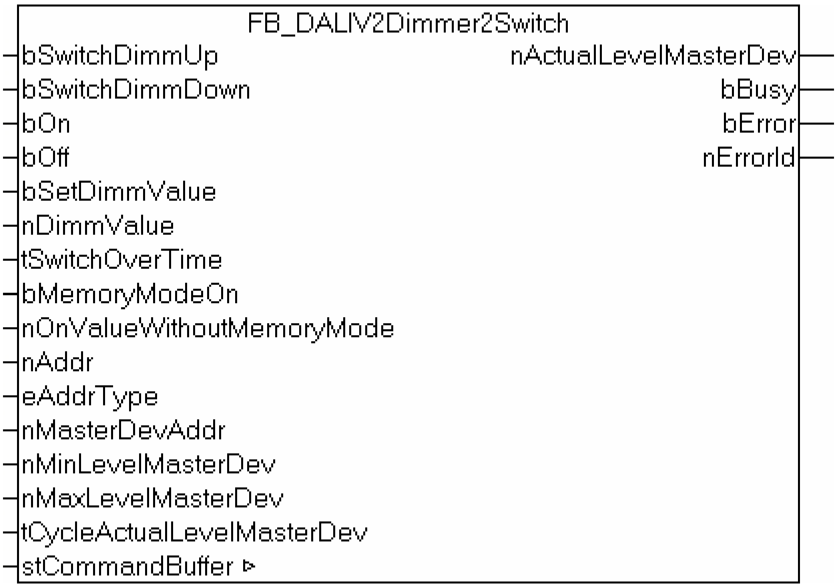
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
arrCommandBuffer        : ARRAY [0..4] OF ST_DALIV2CommandBuffer;
```

arrCommandBuffer: A reference to the internal structures for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.27 FB_DALIV2Dimmer2Switch



The functions available in the FB_DALIDimmer2Switch() function block correspond to those in the FB_DALIV2Dimmer1Switch() [▶ 55] function block. The difference is simply that two switches are connected to the FB_DALIDimmer2Switch() function block. This allows the user to choose specifically between dimming up or dimming down.

Operation by means of the *bSwitchDimmUp* and *bSwitchDimmDown* inputs

The light is switched on or off by a short signal at the *bSwitchDimmUp* or *bSwitchDimmDown* inputs. Dimmer mode will be activated if the signal remains for longer than *tSwitchOverTime* (typical recommended value: 200ms). The addressed devices are now dimmed to the levels specified by *nMaxLevelMasterDev* and *nMinLevelMasterDev*. When the signal is once more removed, the output signal being generated at that time is retained. Another pulse at one of the inputs will set the output to 0.

Operation by means of the *bOn* and *bOff* inputs

The light is immediately switched on or off if a rising edge is applied to the *bOn* or *bOff* inputs. The output value is set to 0 when switching off. The switch-on behaviour can be affected by the memory function (see below).

Operation by means of the *bSetDimmValue* and *nDimmValue* inputs

If the value of *nDimmValue* changes the devices concerned will be switched to this brightness value immediately. The significant point here is that the value changes. The lighting is switched off by changing the value to 0. If there is a rising edge at the *bSetDimmValue* input, the value of *nDimmValue* immediately appears at the output. Immediate modification of the output can be suppressed by a static 1- signal at the *bSetDimmValue* input. This makes it possible to apply a value to the *nDimmValue* input, but for this value only to be passed to the output at the next rising edge of *bSetDimmValue*.
The *bSetDimmValue* and *nDimmValue* inputs can be used to implement a variety of lighting scenarios. Using *nDimmValue* to set the outputs directly can be used to achieve particular brightness levels, either directly or by continuously changing the value. *nDimmValue* must have a value between *nMinLevelMasterDev* and *nMaxLevelMasterDev*. The value 0 is an exception. If the value is outside this range, the output value is limited to the upper or lower limit, as appropriate.

The memory function

It is necessary to determine whether the memory function (*bMemoryModeOn* input) is active or not at switch-on. If the memory function is active, then the last set value is adopted as the brightness value as soon as the device is switched on. If the memory function is not active, a brightness specified by the *nOnValueWithoutMemoryMode* parameter is assigned to the devices concerned. It is irrelevant, in this case, whether the light it has been switched on by means of the *bOn* input or the *bSwitchDimmUp* / *bSwitchDimmDown* input. It should be noted that the *nOnValueWithoutMemoryMode* parameter must lie between *nMinLevelMasterDev* and *nMaxLevelMasterDev*. If this is not the case, the output value is adjusted to the upper or lower limit, as appropriate.

Parameter *tSwitchOverTime*

If a duration of 0 is specified for the parameter *tSwitchOverTime*, the *bSwitchDimmUp* and *bSwitchDimmDown* inputs can only be used to dim the light. Switching on and off is only possible with the *bOn* and *bOff* inputs.

Parameter *nMasterDevAddr*

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.

VAR_INPUT

```
bSwitchDimmUp                : BOOL;
bSwitchDimmDown              : BOOL;
bOn                          : BOOL;
bOff                         : BOOL;
bSetDimmValue                : BOOL;
nDimmValue                   : BYTE;
tSwitchOverTime              : TIME := t#400ms;
bMemoryModeOn                : BOOL := FALSE;
nOnValueWithoutMemoryMode    : BYTE := 254;
nAddr                        : BYTE := 0;
eAddrType                    : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr               : BYTE := 0;
nMinLevelMasterDev           : BYTE := 126;
nMaxLevelMasterDev           : BYTE := 254;
tCycleActualLevelMasterDev   : TIME := t#0s;
```

bSwitchDimmUp: Switches or dims the addressed devices up.

bSwitchDimmDown: Switches or dims the addressed devices down.

bOn: Switches the addressed devices to the last output value, or to the value specified by *nOnValueWithoutMemoryMode*.

bOff: Switches the addressed devices off (value 0).

bSetDimmValue: A positive edge at this input sets the addressed devices immediately to the brightness value that is asserted at the *nDimmValue* input. If the value of *nDimmValue* changes, the brightness value is set immediately to the changed value if the *bSetDimmValue* input is FALSE.

nDimmValue: see *bSetDimmValue.*

tSwitchOverTime: Time for switching between the light on/off and dimming functions for the *bSwitchDimmUp* and *bSwitchDimDown* inputs.

bMemoryModeOn: Switches over to use the memory function, so that the previous value is written to the output as soon as it is switched on.

nOnValueWithoutMemoryMode: Value at switch on if the memory function is not active.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

nMinLevelMasterDev: The minimum value of the master device.

nMaxLevelMasterDev: The maximum value of the master device.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▸ 364]) in the background. So that the dimming of the lamps is not disturbed, reading always has the lowest priority. If the value is set to 0, reading is prohibited.

VAR_OUTPUT

```
nActualLevelMasterDev  : BYTE;
bBusy                  : BOOL;
bError                 : BOOL;
nErrorId               : UDINT;
```

nActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

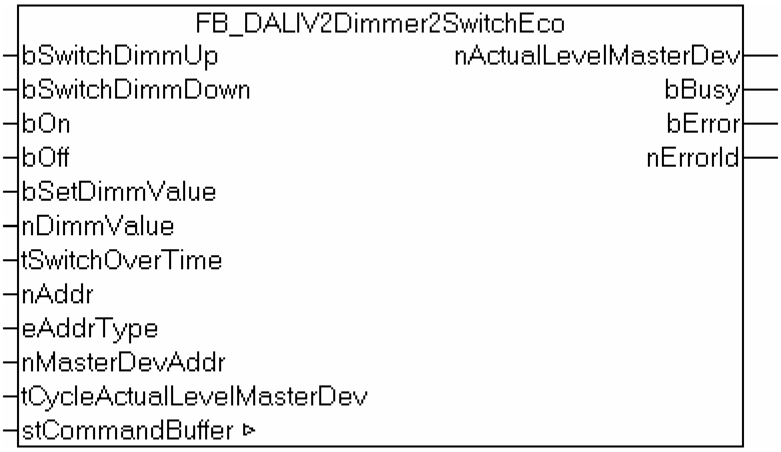
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer          : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.28 FB_DALIV2Dimmer2SwitchEco



The FB_DALIDimmer2SwitchEco() block is a variant of FB_DALIV2Dimmer2Switch() [▸ 62] that saves memory space. It does not include the special "Switch memory function off" function.

Operation by means of the *bSwitchDimmUp* and *bSwitchDimmDown* inputs

The light is switched on or off by a short signal at the *bSwitchDimmUp* or *bSwitchDimmDown* inputs. Dimmer mode will be activated if the signal remains for longer than *tSwitchOverTime* (typical recommended value: 200ms). The addressed devices are now dimmed. When the signal is once more removed, the output signal being generated at that time is retained. Another pulse at one of the inputs will set the output to 0.

Operation by means of the *bOn* and *bOff* inputs

The light is immediately switched on or off if a rising edge is applied to the *bOn* or *bOff* inputs. The output value is set to 0 when switching off.

The memory function

In contrast to FB_DALIV2Dimmer2Switch() [▶ 62], where the memory function can be switched on or off through the *bMemoryModeOn* input, the memory function is always active on this memory-saving version. This means that the most recently set value is adopted for the brightness when switching on. It is irrelevant, in this case, whether the light it has been switched on by means of the *bOn* input or the *bSwitchDimm* input.

Parameter *tSwitchOverTime*

If a duration of 0 is specified for the parameter *tSwitchOverTime*, the *bSwitchDimmUp* and *bSwitchDimmDown* inputs can only be used to dim the light. Switching on and off is only possible with the *bOn* and *bOff* inputs.

Parameter *nMasterDevAddr*

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.

VAR_INPUT

```
bSwitchDimmUp                : BOOL;
bSwitchDimmDown              : BOOL;
bOn                          : BOOL;
bOff                         : BOOL;
bSetDimmValue                : BOOL;
nDimmValue                   : BYTE;
tSwitchOverTime              : TIME := t#400ms;
nAddr                        : BYTE := 0;
eAddrType                    : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr               : BYTE := 0;
tCycleActualLevelMasterDev   : TIME := t#0s;
```

bSwitchDimmUp: Switches or dims the addressed devices up.

bSwitchDimmDown: Switches or dims the addressed devices down.

bOn: Switches the addressed devices to the most recent output value.

bOff: Switches the addressed devices off (value 0).

bSetDimmValue: A positive edge at this input sets the addressed devices immediately to the brightness value that is asserted at the *nDimmValue* input. If the value of *nDimmValue* changes, the brightness value is set immediately to the changed value if the *bSetDimmValue* input is FALSE.

nDimmValue: see *bSetDimmValue.*

tSwitchOverTime: Time for switching between the light on/off and dimming functions for the *bSwitchDimm* input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▶ 364]) in the background. So that the dimming of the lamps is not disturbed, reading always has the lowest priority. If the value is set to 0, reading is prohibited.

VAR_OUTPUT

```
nActualLevelMasterDev  : BYTE;
bBusy                  : BOOL;
bError                 : BOOL;
nErrorId               : UDINT;
```

nActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

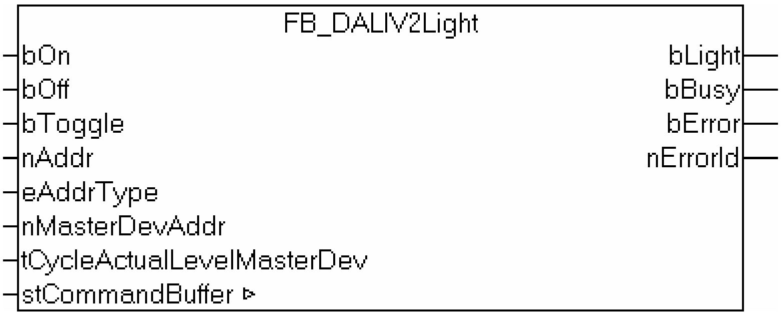
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer         : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.29 FB_DALIV2Light



The FB_DALIV2Light() block is a simple block for switching DALI lamps on and off.

Operation

A positive edge applied to the *bOn* input will switch the light to the maximum value (MAX LEVEL [▶ 364]) of the ballast. If the function block is executed successfully, the *bLight* output is set to TRUE. Applying a positive edge to the *bOff* input will switch the light off, and the *bLight* output will be set to FALSE. If a positive edge is applied to *bToggle*, the function block first reads the current light value from the master device, and then decides whether the status of the lamp is on or off. Once this decision has been reached, the lamp is then placed into whatever the other state is, i.e. it is switched from on to off or from off to on.

Parameter *nMasterDevAddr*

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.

VAR_INPUT

```
bOn                        : BOOL;
bOff                       : BOOL;
bToggle                    : BOOL;
nAddr                      : BYTE := 0;
eAddrType                  : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr             : BYTE := 0;
tCycleActualLevelMasterDev : TIME := t#0s;
```

bOn: Switches the addressed devices on (to the value MAX LEVEL).

bOff: Switches the addressed devices off (to value 0).

bToggle: Reverses the status of the addressed devices.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▶ 364]) in the background. So that the dimming of the lamps is not disturbed, reading always has the lowest priority. If the value is set to 0, reading is prohibited.

VAR_OUTPUT

```
bLight               : BOOL;
bBusy                : BOOL;
bError               : BOOL;
nErrorId             : UDINT;
```

bLight: The status of the lamp or group after the block has been called.

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError : This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

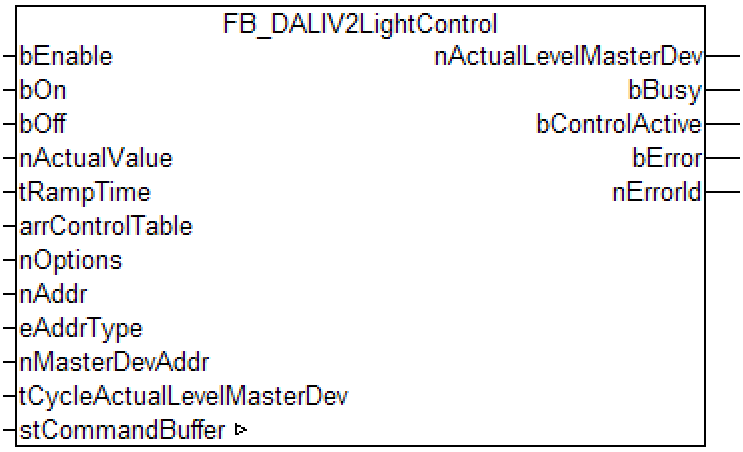
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer         : ST_DALIV2CommandBuffer;
```

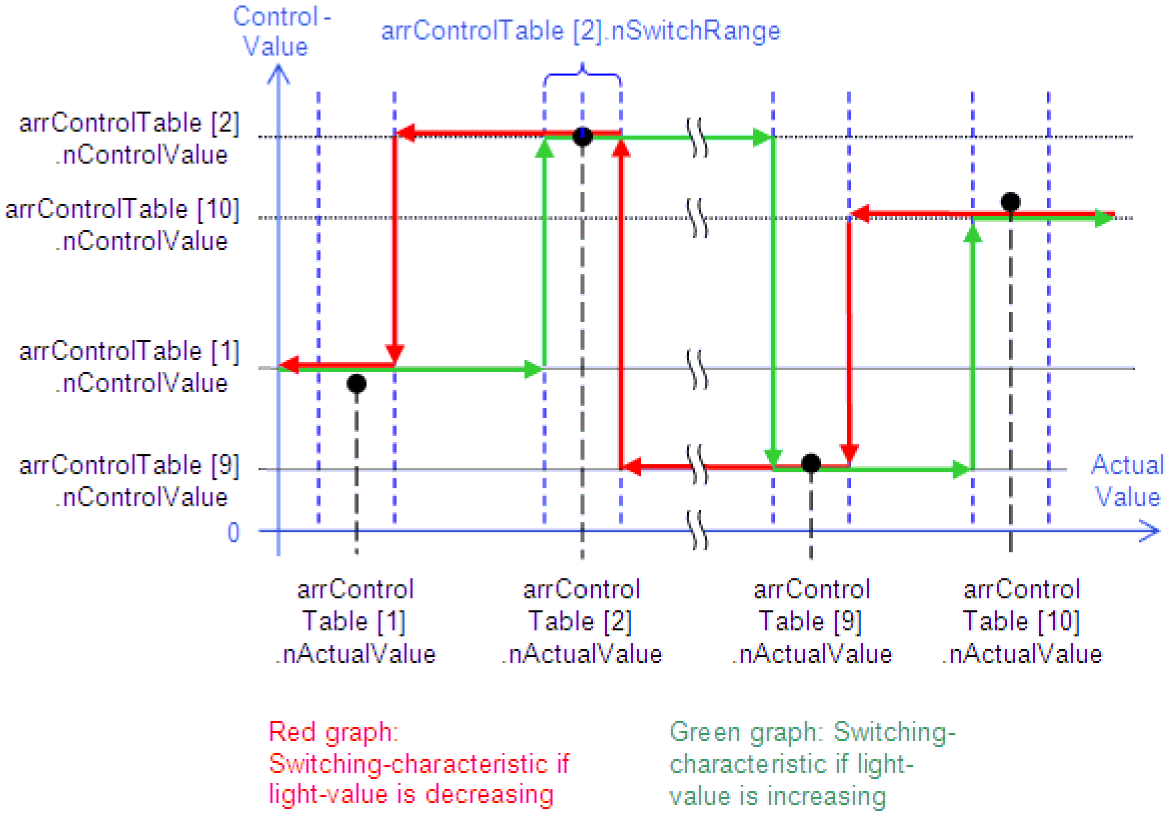
stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.30 FB_DALIV2LightControl



Daylight-Lamp-control.

This function-block is based upon a table of 30 nodes containing measured input- and control-values for threshold-switching. If the actual value comes within the range of a new node (*arrControlTable[n].nActualValue-arrControlTable[n].nSwitchRange/2 ...arrControlTable[n].Input+arrControlTable[n].nSwitchRange/2*), the actual control value will change (see diagram). The threshold-switch is followed by a ramp-function which ramps the control value to the new control value over the time *tRampTime.* With a rising-edge at *bOn* the light is switched immediately to the nearest control value and similarly a rising edge at *bOff* switches the light off without the delay of a ramp. It is possible to trigger a positive edge on *bOn* or bOff *at* anytime.


Control - Value
arrControlTable [2].nSwitchRange
arrControlTable [2] .nControlValue
arrControlTable [10] .nControlValue
arrControlTable [1] .nControlValue
arrControlTable [9] .nControlValue
0
Actual Value
arrControl Table [1] .nActualValue
arrControl Table [2] .nActualValue
arrControl Table [9] .nActualValue
arrControl Table [10] .nActualValue
Red graph: Switching-characteristic if light-value is decreasing
Green graph: Switching-characteristic if light-value is increasing

It is not required to use all 30 entries in the node table. The first element with a *nSwitchRange* of "0" will mark the beginning of the unused table-range.

Parameter *nMasterDevAddr*

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.

VAR_INPUT

```
bEnable                      : BOOL := TRUE;
bOn                          : BOOL;
bOff                         : BOOL;
nActualValue                 : UINT;
tRampTime                    : TIME := t#30s;
arrControlTable              : ARRAY[1..30] OF ST_DALIV2ControlTable;
nOptions                     : DWORD := 0;
nAddr                        : BYTE := 0;
eAddrType                    : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr               : BYTE := 0;
tCycleActualLevelMasterDev   : TIME := t#0s;
```

bEnable: A positive input enables the function block. A negative state deactivates the inputs and sets the function-block to the idle-mode, after the last necessary DALI-commands are sent. Except for the cyclic query of the light-level of the master-device, the function-block sends no further DALI-commands.

bOn: A rising edge sets the addressed lamps directly to the next control value.

bOff: A rising edge turns the addressed lamps immediately off.

nActualValue: measured light-value.

tRampTime: time to drive the lamp from the actual value to the new control value. (Preset value: 30s).

arrControlTable: Input-/control-value-table [▶ 383]. *arrControlTable[1]* to *arrControlTable[30]* of *ST_DALIV2ControlTable*

nOptions: Reserved for future developments.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▶ 364]) in the background. In order not to disturb the dimming of the lamps, reading always has the lowest priority. If the value is set to 0, reading is deactivated.

VAR_OUTPUT

```
nActualLevelMasterDev  : BYTE;
bBusy                  : BOOL;
bControlActive         : BOOL;
bError                 : BOOL;
nErrorId               : UDINT;
```

nActualLevelMasterDev: The current control value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bControlActive: This output is set once the control is activated.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. *bError* is reset to FALSE by the execution of an instruction at the inputs.

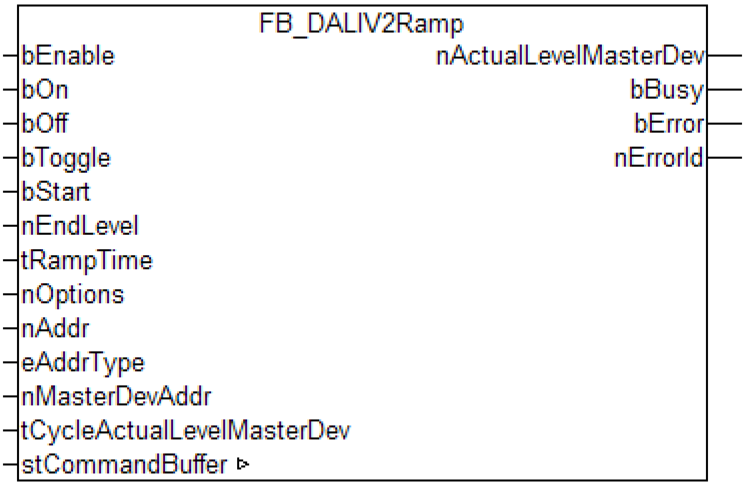
nErrorId: Contains the command-specific error code of the most recently executed command. *nErrorId* is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer              : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.31 FB_DALIV2Ramp



Function-block creating a light-ramp.

With a rising-edge at *bOn* the light will be switched immediately to the maximum-level of the master-device and a rising edge at *bOff* turns the light off. Triggering the input *bToggle* inverts the actual light-state. A rising-edge at *bStart* starts dimming the light from the actual to the end-level (*nEndLevel*) - the required time is defined by *tRampTime*.

As long as *bEnable* is TRUE all inputs are active, otherwise the controlling inputs *bOn*, *bOff*, *bToggle* and *bStart* are deactivated and the function-block turns to its idle-mode. With every start of the dim-ramp the target-value *nEndLevel* is checked if it is within the allowed range of the DALI-Master-Device (MIN LEVEL ... MAX LEVEL [▶ 364] or „0“).

Ramp-Calculation

The functionality of this function-block is based on whether up- or down-dimming has been selected through the *OnAndStepUp*- or *StepDownAndOff*-command. The required number of step-commands are calculated and sent to the lamps equally distributed over the ramp-time.
For ramp times less than 11s it is possible that the calculated time for one step will be less than the time required to run the commands themselves. To achieve these short ramp times (<11s), the function block switches its mode. It will no longer work with a calculated number of steps, but rather with a single *DirectArcPower*-command. This command dims the addressed lamps from their actual-level to the end-level in a specified FADE TIME [▶ 365] which is programmed in the control-devices of the lamps. 16 different values are possible for this *Fade-Time*:

nFadeTime	tFadeTime (s)
0	<0,0707
1	0,707
2	1,000
3	1,414
4	2,000
5	2,828

nFadeTime	tFadeTime (s)
6	4,000
7	5,657
8	8,000
9	11,314
10	16,000
11	22,627
12	32,000
13	45,255
14	64,000
15	90,510

The value closest to the desired ramp-time will be taken and programmed into all selected devices. For example, with a desired ramp-time of 6s the function block will select level 7 = 5,657 s. After driving the ramp successfully, all selected devices will be re-programmed with the original *Fade-Time of* the master-device.

During the ramp-driving the output-value *nActualLevelMasterDev* will be given as a calculated value, in order not to stress the DALI-Bus with query-commands. This calculation is based upon the difference of start- and end-level and the selected ramp-time from the table above. Because this value is a calculated one it may not be exact at any time and is only to be seen as an orientation. After the ramp-driving a query-command is sent to read an exact value for *nActualLevelMasterDev*.

Parameter *nMasterDevAddr*

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.

VAR_INPUT

```
bEnable                      : BOOL := TRUE;
bOn                          : BOOL;
bOff                         : BOOL;
bToggle                      : BOOL;
bStart                       : BOOL;
nEndLevel                    : BYTE := 254;
tRampTime                    : TIME := t#8s;
nOptions                     : DWORD := 0;
nAddr                        : BYTE := 0;
eAddrType                    : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr               : BYTE := 0;
tCycleActualLevelMasterDev   : TIME := t#0s;
```

bEnable: A positive input enables the function block. A negative state deactivates the inputs *bOn*, *bOff*, *bToggle* and *bStart* and sets the function-block to the idle-mode, after the last necessary DALI-commands are sent. Except for the cyclic query of the light-level of the master-device, the function-block sends no further DALI-commands.

bOn: A rising edge sets the addressed lamps directly to the maximum-level of the master-device.

bOff: A rising edge turns the addressed lamps immediately off.

bToggle: Rising edges at this input toggle the state the addressed lamps.

bStart: This input starts the dim-ramp from the actual value (of the master-device) to the target-value *nEndLevel* within the time defined as *tRampTime.* This can be interrupted by *bOn*, *bOff* or *bToggle* at any time.

nEndLevel: Target-value of the dim-ramp. (Valid range: *0* or MIN LEVEL ... MAX LEVEL).

tRampTime: Ramp-time, see *bStart*. (initial value: 8s).

nOptions: Reserved for future developments.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▶ 364]) in the background. In order not to disturb the dimming of the lamps, reading always has the lowest priority. If the value is set to 0, reading is deactivated.

VAR_OUTPUT

```
nActualLevelMasterDev  : BYTE;
bBusy                  : BOOL;
bError                 : BOOL;
nErrorId               : UDINT;
```

ActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. *bError* is reset to FALSE by the execution of an instruction at the inputs.

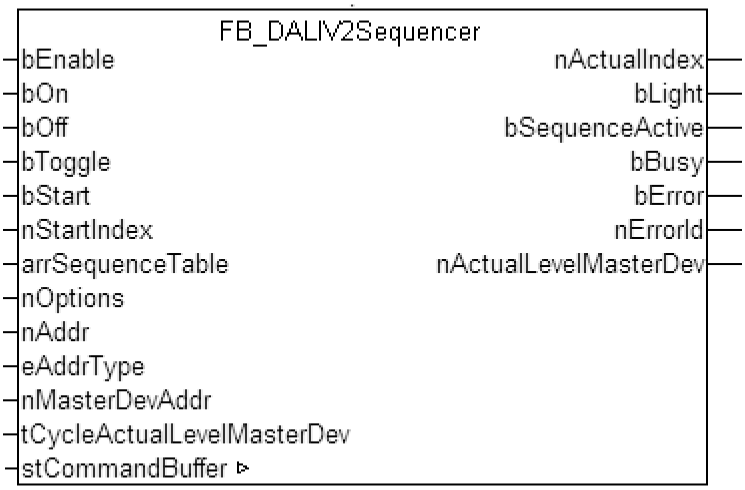
nErrorId: Contains the command-specific error code of the most recently executed command. *nErrorId* is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer           : ST_DALIV2CommandBuffer;
```

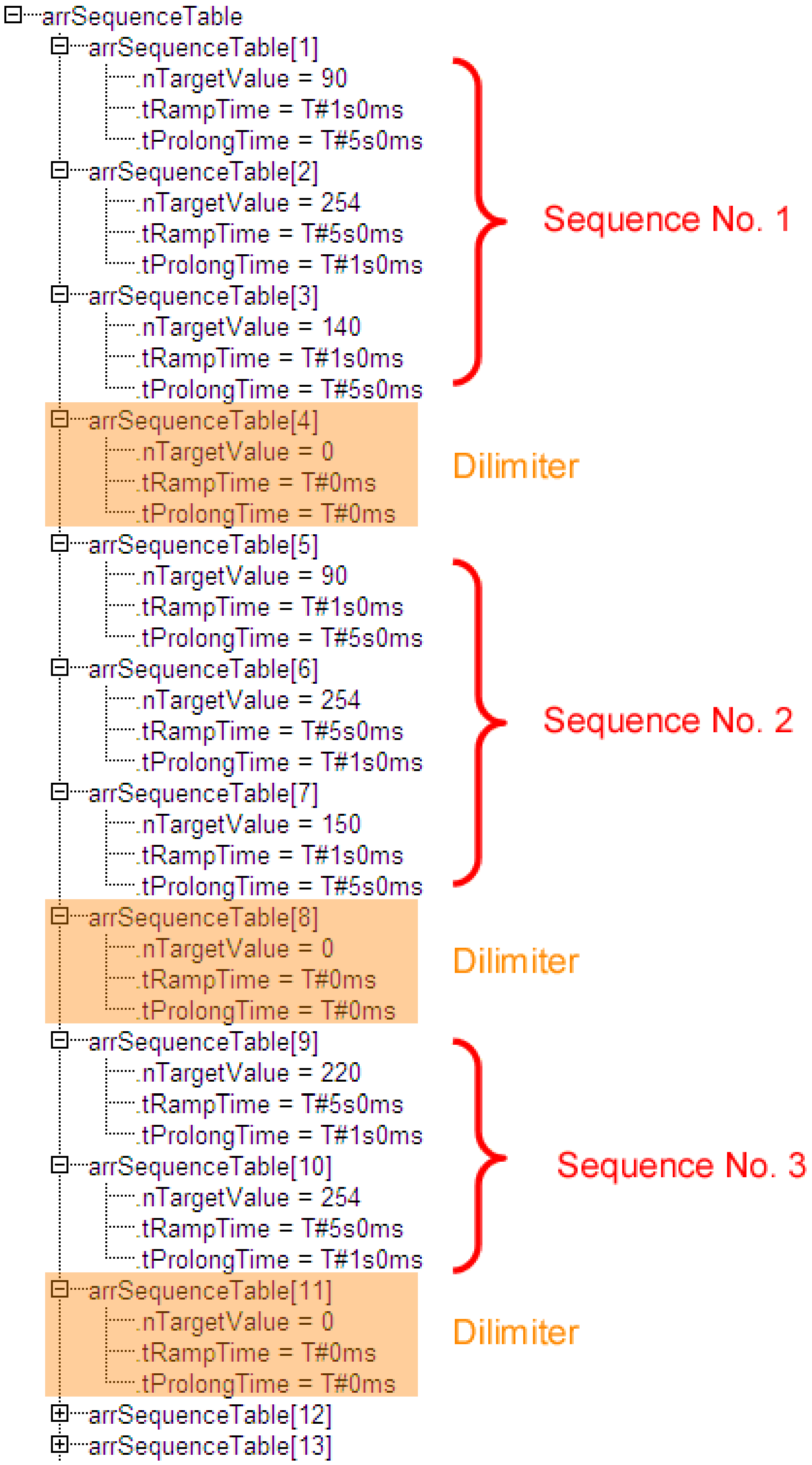
stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.32 FB_DALIV2Sequencer



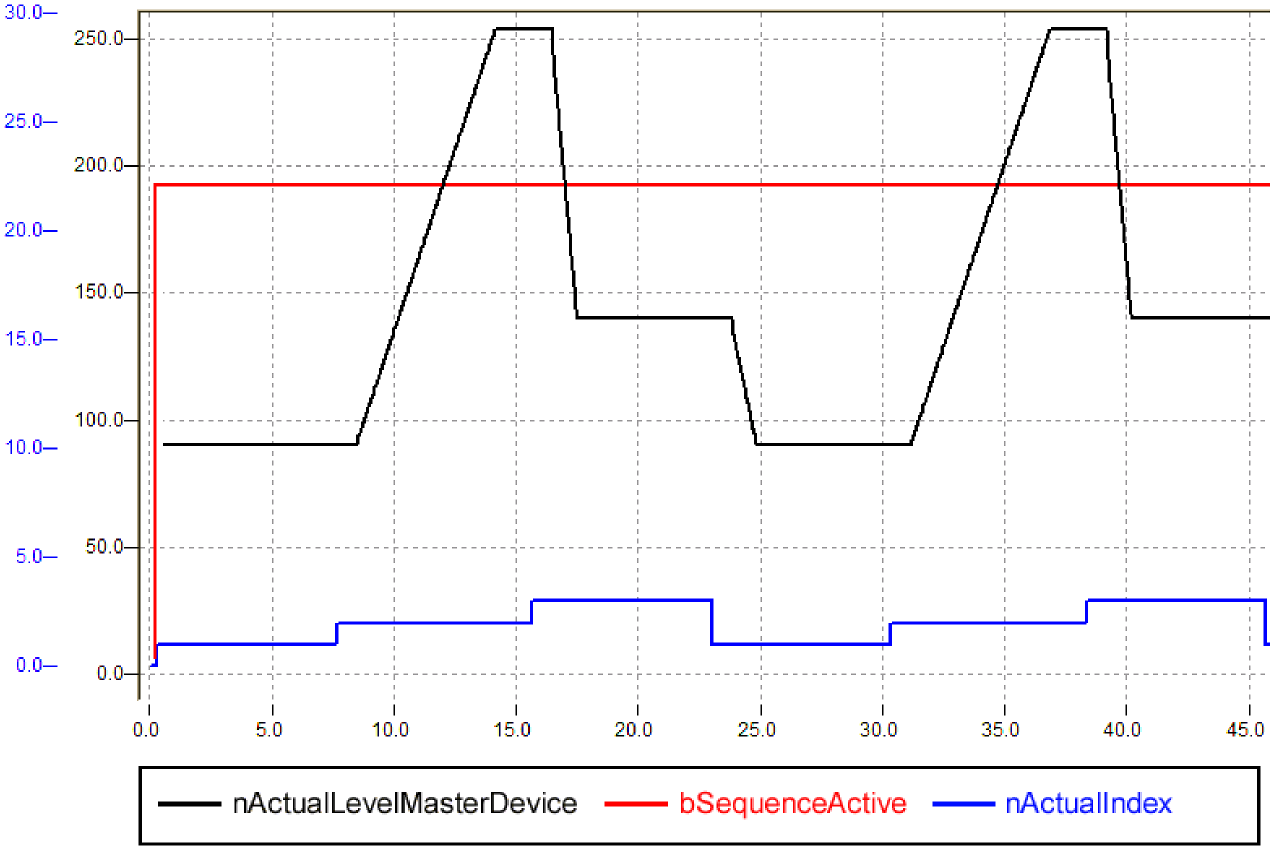
Function block for the implementation of light sequences with up to 50 interpolation points.

The core of this function block is a ramp block that drives over an adjustable time to individual brightness values defined in a table and then remains at this brightness value for a similarly definable time. After the dwell time the next value is then driven to. As already mentioned, the table *arrSequenceTable* consists of 50 entries with the values for *nTargetValue* (target value), *tRampTime* (time taken to reach the target value) and *tProlongTime* (dwell time at the target value). It is not absolutely necessary to use all 50 values. A 0 entry of all 3 values marks the end of a sequence. Beyond that it is possible using the *nStartIndex* input to have a light sequence begin at any desired place in the table. This allows several different light sequences to be programmed even within the 50 entries, the sequences being separated from one another by 0 entry elements:



```
⊟arrSequenceTable
  ⊟arrSequenceTable[1]
     .nTargetValue = 90
     .tRampTime = T#1s0ms
     .tProlongTime = T#5s0ms
  ⊟arrSequenceTable[2]
     .nTargetValue = 254
     .tRampTime = T#5s0ms
     .tProlongTime = T#1s0ms
  ⊟arrSequenceTable[3]
     .nTargetValue = 140
     .tRampTime = T#1s0ms
     .tProlongTime = T#5s0ms
  ⊟arrSequenceTable[4]
     .nTargetValue = 0
     .tRampTime = T#0ms
     .tProlongTime = T#0ms
  ⊟arrSequenceTable[5]
     .nTargetValue = 90
     .tRampTime = T#1s0ms
     .tProlongTime = T#5s0ms
  ⊟arrSequenceTable[6]
     .nTargetValue = 254
     .tRampTime = T#5s0ms
     .tProlongTime = T#1s0ms
  ⊟arrSequenceTable[7]
     .nTargetValue = 150
     .tRampTime = T#1s0ms
     .tProlongTime = T#5s0ms
  ⊟arrSequenceTable[8]
     .nTargetValue = 0
     .tRampTime = T#0ms
     .tProlongTime = T#0ms
  ⊟arrSequenceTable[9]
     .nTargetValue = 220
     .tRampTime = T#5s0ms
     .tProlongTime = T#1s0ms
  ⊟arrSequenceTable[10]
     .nTargetValue = 254
     .tRampTime = T#5s0ms
     .tProlongTime = T#1s0ms
  ⊟arrSequenceTable[11]
     .nTargetValue = 0
     .tRampTime = T#0ms
     .tProlongTime = T#0ms
  ⊞arrSequenceTable[12]
  ⊞arrSequenceTable[13]
```

Over the course of time sequence 1, for example, looks like the following (*nStartIndex=1, nOptions.bit0=TRUE*, see below for explanation):



Beyond that the function block can be switched "normally" on and off (On: maximum value of the lamps, Off: 0) and switched back and forth between "On" and "Off" using the *bToggle* input. However, none of the command inputs is active unless the *bEnable* input is *TRUE*. If it is reset to *FALSE*, no more commands are accepted and the light value retains its current state – even from a ramp.

As explained at the beginning, this function block is based on the FB_DALIV2Ramp [▶ 70]. The ramp block tries to map the set ramp time as accurately as possible. Nevertheless it is necessary to query data from the DALI control gears both once and cyclically, which takes a different amount of time depending on the set PLC cycle time. Therefore inaccuracies in the ramp time cannot be ruled out.

VAR_INPUT

```
bEnable                      : BOOL := TRUE;
bOn                          : BOOL;
bOff                         : BOOL;
bToggle                      : BOOL;
bStart                       : BOOL;
nStartIndex                  : USINT := 0;
arrSequenceTable             : ARRAY [1..nMaxSequenceValues] OF ST_DALIV2SequenceTable;
nOptions                     : DWORD := 0;
nAddr                        : BYTE := 0;
eAddrType                    : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr               : BYTE := 0;
tCycleActualLevelMasterDev   : TIME := t#0s;
```

bEnable: A positive input enables the function block. A negative state deactivates the inputs the inputs *bOn*, *bOff*, *bToggle* and *bStart* and sets the function-block to the idle-mode.

bOn: A rising edge sets the output *nLightLevel* directly to the maximum-level.

bOff: A rising edge sets the output *nLightLevel* immediately to "*0*".

bToggle: Rising edges at this input toggle the light.

bStart: This input lets the sequence begin with the element defined with *nStartIndex*.

nStartIndex: See *bStart.*

arrSequenceTable: Light-value-table [▸ 384] with the information about the target-value, the ramp-time and the prolong-time.

nOptions: Parameter-input. Setting (resp. not-setting) of the single bits will affect the behavior of the function-block as follows:

Bit	Description
0	**not set:** After running through a sequence, the function-block will stop its activity. To start again, a rising edge at *bStart* is necessary. **set:** After running through a sequence, the function-block will automatically restart at the element defined with *nStartIndex*.
1 - 31	-- reserved for future*developments* --

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▸ 364]) in the background. In order not to disturb the dimming of the lamps, reading always has the lowest priority. If the value is set to 0, reading is deactivated.

VAR_OUTPUT

```
nActualLevelMasterDev  : BYTE;
nActualIndex           : USINT;
bLight                 : BOOL;
bSequenceActive        : BOOL;
bBusy                  : BOOL;
bError                 : BOOL;
nErrorId               : UDINT;
```

nActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

nActualIndex : This output shows the actual element of the light-sequence. If the sequence is finished or stopped (*bSequenceActive* = *FALSE* , see below) , the output will fall back to "0".

bLight: As long as *nLightLevel* is greater than "*0*", this output is set to *TRUE*.

bSequenceActive: If a light-sequence is running, this output is set to *TRUE*.

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to *TRUE* if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. *bError* is reset to *FALSE* by the execution of an instruction at the inputs.

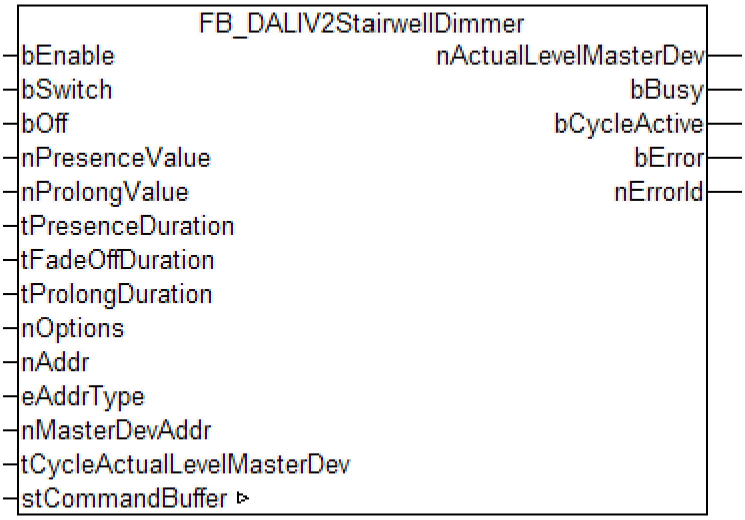
nErrorId: Contains the command-specific error code of the most recently executed command. *nErrorId* is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer          : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.33 FB_DALIV2StairwellDimmer

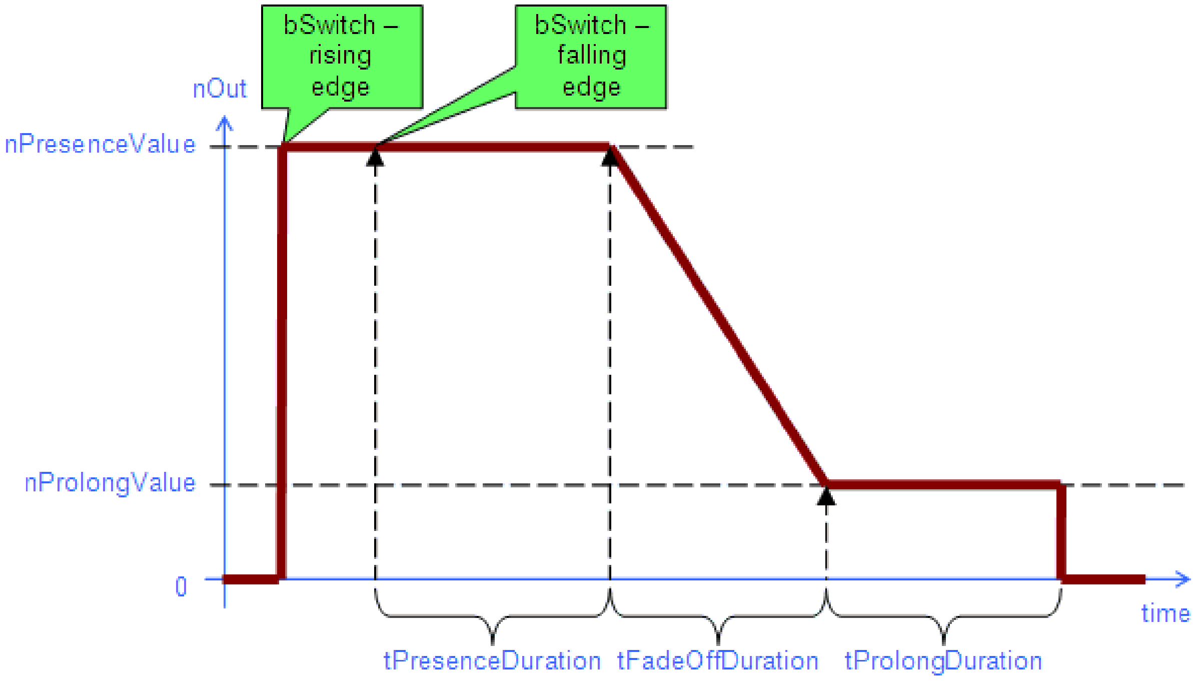


Function block to control stairwell lighting.

A rising edge at the input *bSwitch* switches the addressed lamps to the value *nPresenceValue.* A falling edge on *bSwitch* starts or restarts a timer with the time *tPresenceDuration.* Following the expiry of this timer, the lamps are dimmed to the value *nProlongValue* over the time period *tFadeOffDuration.* This value is maintained for the time period *tProlongDuration.* After that, the light is switched off. A rising edge at the input *bOff* turns the light off immediately. Triggering the input bSwitch again will always restart the cycle, even during the periods of *tPresenceDuration* and *tFadeOffDuration.* With every start the setpoint-values are checked if they are within the allowed range of the DALI-Master-Device (MIN LEVEL … MAX LEVEL [▸ 364] or „0“) .

Parameter *nMasterDevAddr*

The DALI system provides facilities not just for controlling lamps individually, but also for addressing them as groups or through common commands. Because the individual devices can be members of a variety of groups, it can happen that, prior to the issue of a group or common control command, the individual lamps have different brightness levels. So that it is nevertheless possible to be clear whether the lamps now are to be switched on or off, a master device is assigned to each group, whose state is followed by the other devices. It is not necessary to specify a master device if the function block is to be used to control a single lamp, *eAddrType* = *eAddrTypeShort*. In this case, the *nMasterDevAddr* parameter has no significance.



VAR_INPUT

```
bEnable                    : BOOL := TRUE;
bSwitch                    : BOOL;
bOff                       : BOOL;
nPresenceValue             : BYTE := 254;
nProlongValue              : BYTE := 200;
tPresenceDuration          : TIME := t#30s;
tFadeOffDuration           : TIME := t#10s;
tProlongDuration           : TIME := t#20s;
nOptions                   : DWORD := 0;
nAddr                      : BYTE := 0;
eAddrType                  : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nMasterDevAddr             : BYTE := 0;
tCycleActualLevelMasterDev : TIME := t#0s;
```

bEnable: A positive input enables the function block. A negative state deactivates the inputs and sets the function-block to the idle-mode, after the last necessary DALI-commands are sent. Except for the cyclic query of the light-level of the master-device, the function-block sends no further DALI-commands.

bSwitch: A rising edge sets the light to *nPresenceValue*. A falling edge starts the presence time (see diagram).

bOff: Turns the light off immediately.

nPresenceValue: Value to which the light should be set during the presence time. (Valid range: *0* or MIN LEVEL ... MAX LEVEL - Preset value: 254).

nProlongValue: Value to which the light should be set to following a falling edge on *bSwitch* after the time *tPresenceDuration*. (Valid range: *0* or MIN LEVEL ... MAX LEVEL - Preset value: 200).

tPesenceDuration: Duration of the presence time in which the light is set to *nPresenceValue* following a falling edge on *bSwitch*. (Preset value: 30 seconds).

tFadeOffDuration: Within this time the light is driven from the presence- to the prolong-value (Preset value: 10 seconds).

tProlongDuration: Duration of the dwell time. (Preset value: 20 seconds).

nOptions: Reserved for future developments.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nMasterDevAddr: The address of the master device for group and common switching operations.

tCycleActualLevelMasterDev: Cycle time required to read the current actual value (ACTUAL DIM LEVEL [▶ 364]) in the background. In order not to disturb the dimming of the lamps, reading always has the lowest priority. If the value is set to 0, reading is deactivated.

VAR_OUTPUT

```
nActualLevelMasterDev  : BYTE;
bBusy                  : BOOL;
bError                 : BOOL;
nErrorId               : UDINT;
```

nActualLevelMasterDev: The current output value of the master device (if *eAddrType* = *eAddrTypeShort* this is always the device being addressed at the time).

bBusy: This output is set during light-change, which means turning on, off and driving a ramp. The start- and end-value are not significant. Even driving a ramp from 100 to 100 in 10s will set this output to TRUE.

bCycleActive: When the block is activated the output is set, and it remains active until the cycle has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. *bError* is reset to FALSE by the execution of an instruction at the inputs.

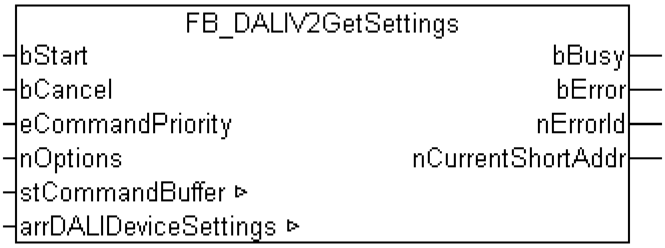
nErrorId: Contains the command-specific error code of the most recently executed command. *nErrorId* is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer         : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.34 FB_DALIV2GetSettings



This block reads the variables (MIN LEVEL, MAX LEVEL, FADE TIME, ...) of all ballasts within DALI line and saves them in a structure of type ST_DALIV2DeviceSettings [▶ 383].

Applying a positive edge to the *bStart* input starts the block, and the *bBusy* output goes TRUE. A check is first made as to whether a ballast is present at all. If this is the case, then the *bPresent* bit is set in the corresponding structure (see ST_DALIV2DeviceSettings [▶ 383]), after which the settings of the ballast are read out one by one and written into 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 *nErrors* element 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 ballast.

Bit	Error
0	An error occurred whilst attempting to seek the ballast.

Bit	Error
1	Error whilst reading the variable ACTUAL DIM LEVEL [▶ 364]
2	Error whilst reading the variable POWER ON LEVEL [▶ 364]
3	Error whilst reading the variable SYSTEM FAILURE LEVEL [▶ 364]
4	Error whilst reading the variable MIN LEVEL [▶ 364]
5	Error whilst reading the variable MAX LEVEL
6	Error whilst reading the variable FADE RATE [▶ 364]
7	Error whilst reading the variable FADE TIME [▶ 365]
8	Error whilst reading the variable RANDOM ADDRESS [▶ 365]
9	Error whilst reading the variables GROUP 0-7, GROUP 8-15 [▶ 366]
10	Error whilst reading the variables SCENE 0 to SCENE 15 [▶ 366]
11	Error whilst reading the variable STATUS INFORMATION [▶ 366]
12	Error whilst reading the variable VERSION NUMBER [▶ 366]
13	Error whilst reading the variable DEVICE TYPE [▶ 366]
14	Error whilst reading the variable PHYSICAL MIN LEVEL [▶ 367]

When the function block has been processed, the *bBusy* output changes from TRUE to FALSE. Processing this function block can take several seconds, depending on how many ballasts are attached.

VAR_INPUT

```
bStart              : BOOL;
bCancel             : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nOptions            : DWORD;
```

bStart: The block is activated by a rising edge at this input.

bCancel: A rising edge at this input will deactivate the block and hence abort the reading of the variable.

eCommandPriority: The priority (high, middle, low) this command has when executed by the library.

nOptions: Options for reading the variables (see table). The individual constants must be linked with OR operators.

Constants	Description
DALIV2_OPTION_ACTUAL_DIM_LEVEL	The variable ACTUAL DIM LEVEL
DALIV2_OPTION_POWER_ON_LEVEL	The variable POWER ON LEVEL
DALIV2_OPTION_SYSTEM_FAILURE_LEVEL	The variable SYSTEM FAILURE LEVEL
DALIV2_OPTION_MIN_LEVEL	The variable MIN LEVEL
DALIV2_OPTION_MAX_LEVEL	The variable MAX LEVEL
DALIV2_OPTION_FADE_RATE_FADE_TIME	The variables FADE RATE
DALIV2_OPTION_RANDOM_ADDRESS	The variable RANDOM ADDRESS
DALIV2_OPTION_GROUPS	The variables GROUP 0-7 GROUP 8-15 are read
DALIV2_OPTION_SCENE_LEVELS	The variables SCENE 0 to SCENE 15
DALIV2_OPTION_STATUS_INFORMATION	The variable STATUS INFORMATION
DALIV2_OPTION_VERSION_NUMBER	The variable VERSION NUMBER
DALIV2_OPTION_DEVICE_TYPE	The variable DEVICE TYPE
DALIV2_OPTION_PHYSICAL_MIN_LEVEL	The variable PHYSICAL MIN LEVEL
DALIV2_OPTION_DONT_CLEAR_DEVICE_SETTINGS	Before reading doesn't delete the variable *arrDALIDeviceSettings*.
DALIV2_OPTION_ALL	All variables are read.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nCurrentShortAddr  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

nCurrentShortAddr: Short address of the ballast from which variables are being read.

VAR_IN_OUT

```
stCommandBuffer        : ST_DALIV2CommandBuffer;
arrDALIDeviceSettings  : ARRAY[0..63] OF ST_DALIV2DeviceSettings;
```

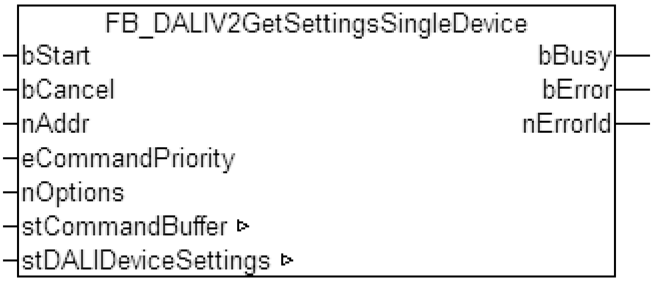
stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

arrDALIDeviceSettings: Reference to an array of 64 elements of type ST_DALIV2DeviceSettings [▸ 383]. The settings of each individual DALI ballast are stored in these variables.

Also see about this

- E_DALIV2CommandPriority [▸ 381]
- DALI ballast variables [▸ 364]
- DALI ballast variables [▸ 364]
- DALI ballast variables [▸ 364]
- DALI ballast variables [▸ 364]
- DALI ballast variables [▸ 364]
- DALI ballast variables [▸ 365]
- DALI ballast variables [▸ 366]
- DALI ballast variables [▸ 366]
- DALI ballast variables [▸ 366]
- DALI ballast variables [▸ 366]
- DALI ballast variables [▸ 366]
- DALI ballast variables [▸ 367]

5.1.35 FB_DALIV2GetSettingsSingleDevice



This block reads the variables (MIN LEVEL, MAX LEVEL, FADE TIME, ...) of a single DALI-device and saves them in a structure of the type ST_DALIV2DeviceSettings [▸ 383].

With a positive edge at the input *bStart* the function-block begins its operation, reads all values one after another out of the addressed device and stores them into ST_DALIV2DeviceSettings [▶ 383]. If an error occurs, the specific bit of the variable *nErrors* in the structure will be set. The following table gives an overview of all possible errors:

Bit	Error
0	An error occurred whilst attempting to seek the ballast.
1	Error whilst reading the variable ACTUAL DIM LEVEL [▶ 364]
2	Error whilst reading the variable POWER ON LEVEL [▶ 364]
3	Error whilst reading the variable SYSTEM FAILURE LEVEL [▶ 364]
4	Error whilst reading the variable MIN LEVEL [▶ 364]
5	Error whilst reading the variable MAX LEVEL
6	Error whilst reading the variable FADE RATE [▶ 364]
7	Error whilst reading the variable FADE TIME [▶ 365]
8	Error whilst reading the variable RANDOM ADDRESS [▶ 365]
9	Error whilst reading the variables GROUP 0-7 and GROUP 8-15 [▶ 366]
10	Error whilst reading the variables SCENE 0 to SCENE 15 [▶ 366]
11	Error whilst reading the variable STATUS INFORMATION [▶ 366]
12	Error whilst reading the variable VERSION NUMBER [▶ 366]
13	Error whilst reading the variable DEVICE TYPE [▶ 366]
14	Error whilst reading the variable PHYSICAL MIN LEVEL [▶ 367]

When the function block has been processed, the *bBusy* output changes from TRUE to FALSE.

VAR_INPUT

```
bStart              : BOOL;
bCancel             : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nOptions            : DWORD;
```

bStart: The block is activated by a rising edge at this input.

bCancel: A rising edge at this input will deactivate the block and hence abort the reading of the variable.

nAddr: The address of the device, which values shall be read.

eCommandPriority: The priority (high, middle, low) this command has when executed by the library.

nOptions: Options for reading the variables (see table). The individual constants must be linked with OR operators.

Constants	Description
DALIV2_OPTION_ACTUAL_DIM_LEVEL	The variable ACTUAL DIM LEVEL
DALIV2_OPTION_POWER_ON_LEVEL	The variable POWER ON LEVEL
DALIV2_OPTION_SYSTEM_FAILURE_LEVEL	The variable SYSTEM FAILURE LEVEL
DALIV2_OPTION_MIN_LEVEL	The variable MIN LEVEL
DALIV2_OPTION_MAX_LEVEL	The variable MAX LEVEL
DALIV2_OPTION_FADE_RATE_FADE_TIME	The variables FADE RATE and FADE TIME
DALIV2_OPTION_RANDOM_ADDRESS	The variable RANDOM ADDRESS
DALIV2_OPTION_GROUPS	The variables GROUP 0-7 and GROUP 8-15
DALIV2_OPTION_SCENE_LEVELS	The variables SCENE 0 to SCENE 15
DALIV2_OPTION_STATUS_INFORMATION	The variable STATUS INFORMATIO
DALIV2_OPTION_VERSION_NUMBER	The variable VERSION NUMBER
DALIV2_OPTION_DEVICE_TYPE	The variable DEVICE TYPE
DALIV2_OPTION_PHYSICAL_MIN_LEVEL	The variable PHYSICAL MIN LEVEL

Constants	Description
DALIV2_OPTION_DONT_CLEAR_DEVICE_SETTINGS	Before reading doesn't delete the variable *stDALIDeviceSettings*.
DALIV2_OPTION_ALL	All variables are read.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
stDALIDeviceSettings : ST_DALIV2DeviceSettings;
```

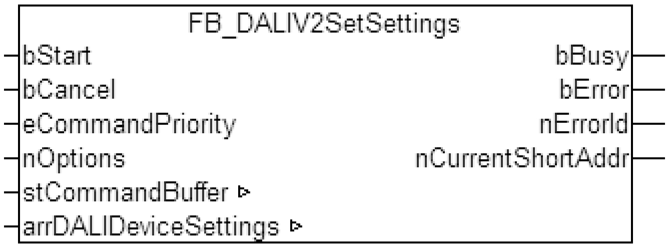
stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

stDALIDeviceSettings: A reference to the structure of type ST_DALIV2DeviceSettings [▶ 383]. The settings of the DALI ballast are stored in these variable.

Also see about this

- FB_DALIV2GetSettings [▶ 78]
- E_DALIV2CommandPriority [▶ 381]
- DALI ballast variables [▶ 364]
- DALI ballast variables [▶ 364]
- DALI ballast variables [▶ 364]
- DALI ballast variables [▶ 364]
- DALI ballast variables [▶ 364]
- DALI ballast variables [▶ 365]
- DALI ballast variables [▶ 366]
- DALI ballast variables [▶ 366]
- DALI ballast variables [▶ 366]
- DALI ballast variables [▶ 366]
- DALI ballast variables [▶ 367]

5.1.36 FB_DALIV2SetSettings



This block initialises the variables (MIN LEVEL, MAX LEVEL, FADE TIME ...) of all ballasts within a DALI line with the values stored in a structure of type ST_DALIV2DeviceSettings [▶ 383].

Applying a positive edge to the *bStart* input starts the block, and the *bBusy* output goes TRUE. The system first checks whether the *bPresent* bit is set in the respective structure (see ST_DALIV2DeviceSettings [▶ 383]). If this is the case, all ballast variables that are not write-protected are initialised with the respective values of the structure. The structure index here reflects the address of the ballast. 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 write error occurs for a device, the *nErrors* element 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 writing of a variable to a ballast.

Bit	Error
2	Error whilst writing the variable POWER ON LEVEL [▶ 364]
3	Error whilst writing the variable SYSTEM FAILURE LEVEL [▶ 364]
4	Error whilst writing the variable MIN LEVEL [▶ 364]
5	Error whilst writing the variable MAX LEVEL
6	Error whilst writing the variable FADE RATE [▶ 364]
7	Error whilst writing the variable FADE TIME [▶ 365]
9	Error whilst writing the variables GROUP 0-7 [▶ 366] and GROUP 8-15
10	Error whilst writing the variables SCENE 0 to SCENE 15 [▶ 366]

When the function block has been processed, the *bBusy* output changes from TRUE to FALSE. Processing this function block can take several seconds, depending on how many ballasts are attached.

VAR_INPUT

```
bStart               : BOOL;
bCancel              : BOOL;
eCommandPriority     : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nOptions             : DWORD;
```

bStart: The block is activated by a rising edge at this input.

bCancel: A rising edge at this input will deactivate the block and hence abort the initialisation of the variable.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nOptions: Options for writing the variables (see table). The individual constants must be linked with OR operators.

Constants	Description
DALIV2_OPTION_POWER_ON_LEVEL	The variable POWER ON LEVEL
DALIV2_OPTION_SYSTEM_FAILURE_LEVEL	The variable SYSTEM FAILURE LEVEL
DALIV2_OPTION_MIN_LEVEL	The variable MIN LEVEL
DALIV2_OPTION_MAX_LEVEL	The variable MAX LEVEL
DALIV2_OPTION_FADE_RATE	The variable FADE RATE
DALIV2_OPTION_FADE_TIME	The variable FADE TIME
DALIV2_OPTION_GROUPS	The variables GROUP 0-7 and GROUP 8-15
DALIV2_OPTION_SCENE_LEVELS	The variables SCENE 0 to SCENE 15
DALIV2_OPTION_ALL	All variables are initialized.
DALIV2_OPTION_PUSH_DALI_COMMANDS	The buffer, which contains the feedback telegrams from the ballasts, among other things, is **not** read. Hence, writing becomes faster, but errors are not recognised.

VAR_OUTPUT

```
bBusy                : BOOL;
bError               : BOOL;
nErrorId             : UDINT;
nCurrentShortAddr    : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

nCurrentShortAddr: Short address of the ballast for which variables are being initialised.

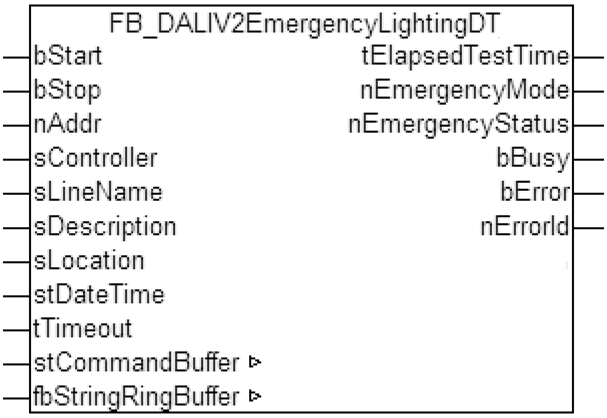
VAR_IN_OUT

```
stCommandBuffer         : ST_DALIV2CommandBuffer;
arrDALIDeviceSettings   : ARRAY[0..63] OF ST_DALIV2DeviceSettings;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

arrDALIDeviceSettings: Reference to an array of 64 elements of type ST_DALIV2DeviceSettings [▶ 383]. The settings of each individual DALI ballast are stored in these variables.

5.1.37 FB_DALIV2EmergencyLightingDT



This function block is for the duration test of a DALI emergency lighting device. At the end of the test the results are written over a FIFO buffer (IN-OUT variable *fbStringRingBuffer*), which in turn is read by the function block FB_DALIV2FileLogging() [▶ 88] into a file. Events that hinder or interrupt the test are displayed in addition to the result message at the *bError* and *nErrorID* outputs.

The following events prevent the execution of a duration test:

- The device is running in automatic duration test mode, i.e. a test interval is programmed in the device.
- The device is currently executing a test or a test is automatically pending (function or duration test)
- The emergency battery is not completely charged
- The device is not in emergency standby ("normal mode") at the start of the test

The events that interrupt a duration test that has begun include:

- The device did not attain the duration test mode after the start of the test
- The test was not correctly completed, i.e. following the test start and a certain waiting period, the device is at some time neither in test mode nor (back) in emergency standby mode ("normal mode")

- A DALI command was incorrectly processed
- The timeout has expired

This function block is available only in the PC version of the DALIV2 library.

It is not possible for the log function block to write data to a file as long as that file is open!

VAR_INPUT

```
bStart        : BOOL;
bStop         : BOOL;
nAddr         : BYTE;
sController   : STRING(20);
sLineName     : STRING(10);
sDescription  : STRING(20);
sLocation     : STRING(20);
stDateTime    : TIMESTRUCT;
tTimeout      : TIME := t#120m;
```

bStart: A rising edge activates the function-block.

bStop: When a Duration-test was successfully started, the Emergency-Mode [▸ 370] and Emergency-Status [▸ 371] will be periodically polled to see, if the test is finished and if any errors occurred. A rising edge at the *bStop*-Input will terminate the test at this stage, giving a special message in the log-file. This feature is neccessary, when it has only to be proven, that the emergency-device is running for a specific time - until the stop-signal is given.

nAddr: Address of the device.

sController: Name of the controller, as a description for the log-file.

sLineName: Name of the line.

sDescription: Further description of the device for the log-file.

sLocation: Location, where the lamp is installed.

stDateTime: Actual date and time.

tTimeout: The test must be successfully executed in this time.

VAR_OUTPUT

```
tElapsedTestTime  : TIME;
nEmergencyMode    : BYTE;
nEmergencyStatus  : BYTE;
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
```

tElapsedTestTime: Test-duration. With a rising edge at *bStart* this value is first set to 0. As long as the function-block is active, it shows the elapsed test-time. With a falling edge at *bBusy* the output remains at its value, so the time, used for the test, is still available.

nEmergencyMode: During the test this output shows the internally queried value of the Emergency-Mode.

nEmergencyStatus: do. Emergency-Status.

bBusy: If the function-block is active, this output is set to TRUE.

bError: This output is switched to TRUE if an error occurs as described above. Is reset by the execution of a new test.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a new test. See Error codes [▸ 385].

VAR_IN_OUT

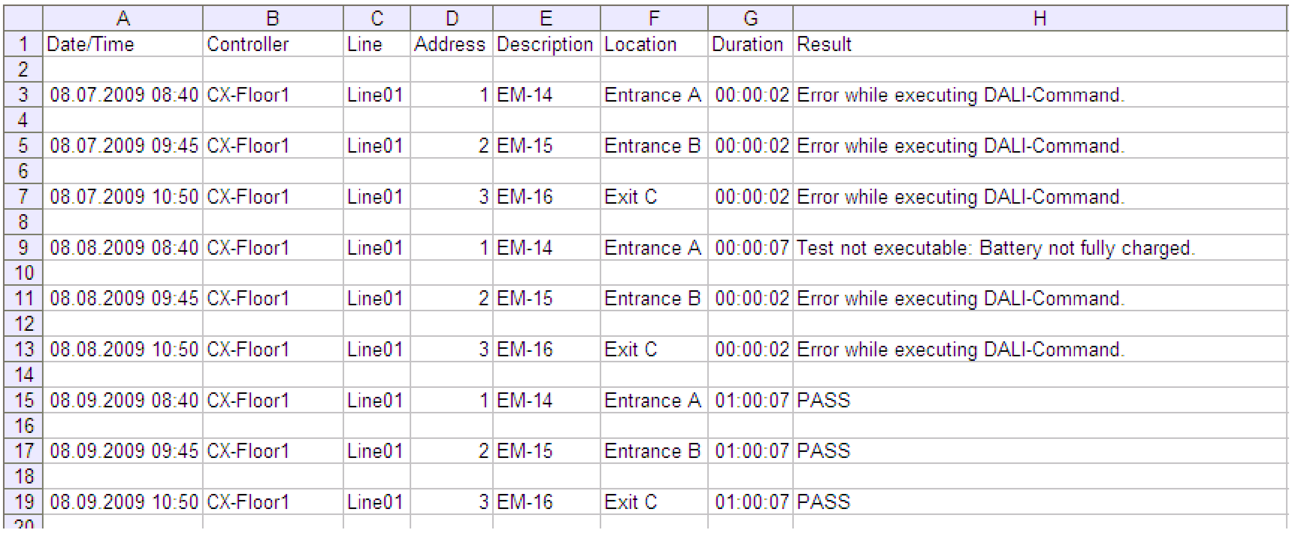
```
stCommandBuffer       : ST_DALIV2CommandBuffer;
fbStringRingBuffer    : FB_MemRingBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

fbStringRingBuffer: Reference to the FIFO-Buffer, in which the log-entries are put.

A https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019292427/.zip shows the correct assignment of the inputs as well as the link to the logger-FB FB_DALIV2FileLogging() [▶ 88]. In this program the Duration-Test for the three emergency-devices will be executed on the 8th of every month.

The log-file will look like this:



	A	B	C	D	E	F	G	H
1	Date/Time	Controller	Line	Address	Description	Location	Duration	Result
2								
3	08.07.2009 08:40	CX-Floor1	Line01	1	EM-14	Entrance A	00:00:02	Error while executing DALI-Command.
4								
5	08.07.2009 09:45	CX-Floor1	Line01	2	EM-15	Entrance B	00:00:02	Error while executing DALI-Command.
6								
7	08.07.2009 10:50	CX-Floor1	Line01	3	EM-16	Exit C	00:00:02	Error while executing DALI-Command.
8								
9	08.08.2009 08:40	CX-Floor1	Line01	1	EM-14	Entrance A	00:00:07	Test not executable: Battery not fully charged.
10								
11	08.08.2009 09:45	CX-Floor1	Line01	2	EM-15	Entrance B	00:00:02	Error while executing DALI-Command.
12								
13	08.08.2009 10:50	CX-Floor1	Line01	3	EM-16	Exit C	00:00:02	Error while executing DALI-Command.
14								
15	08.09.2009 08:40	CX-Floor1	Line01	1	EM-14	Entrance A	01:00:07	PASS
16								
17	08.09.2009 09:45	CX-Floor1	Line01	2	EM-15	Entrance B	01:00:07	PASS
18								
19	08.09.2009 10:50	CX-Floor1	Line01	3	EM-16	Exit C	01:00:07	PASS
20								

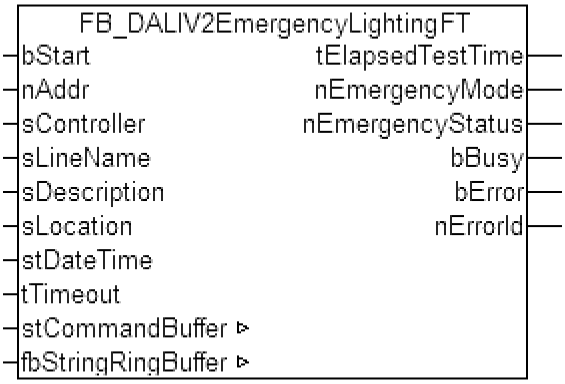
In July the emergency-devices are not installed, yet. The tests result in DALI-command-errors on all devices.

In August the first device is installed, but the battery was not completely charged.

Finally, in September all devices are installed and fully functional. The tests were succesful.

When this file is opened, it may be neccessary to adjust the width of the columns, so all the data is visible.

5.1.38 FB_DALIV2EmergencyLightingFT



This function block is for the function test of a DALI emergency lighting device. At the end of the test the results are written over a FIFO buffer (IN-OUT variable *fbStringRingBuffer*), which in turn is read by the function block FB_DALIV2FileLogging() [▶ 88] into a file. Events that hinder or interrupt the test are displayed in addition to the result message at the *bError* and *nErrorID* outputs.

The following events prevent the execution of a function test:

- The device is running in automatic function test mode, i.e. a test interval is programmed in the device.
- The device is currently executing a test or a test is automatically pending (function or duration test)
- The device is not in emergency standby ("normal mode") at the start of the test

The events that interrupt a function test that has begun include:

- The device did not attain the function test mode after the start of the test
- The test was not correctly completed, i.e. following the test start and a certain waiting period, the device is neither in test mode nor (back) in emergency standby mode ("normal mode")
- A DALI command was incorrectly processed
- The timeout has expired

This function block is available only in the PC version of the DALIV2 library.

It is not possible for the log function block to write data to a file as long as that file is open!

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
sController        : STRING(20);
sLineName          : STRING(10);
sDescription       : STRING(20);
sLocation          : STRING(20);
stDateTime         : TIMESTRUCT;
tTimeout           : TIME := t#120m;
```

bStart: A rising edge activates the function-block.

nAddr: Address of the device.

sController: Name of the controller, as a description for the log-file.

sLineName: Name of the line.

sDescription: Further description of the device for the log-file.

sLocation: Location, where the lamp is installed.

stDateTime: Actual date and time.

tTimeout: The test has to be successfully executed in this time.

VAR_OUTPUT

```
tElapsedTestTime  : TIME;
nEmergencyMode    : BYTE;
nEmergencyStatus  : BYTE;
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
```

tElapsedTestTime: Test-duration. With a rising edge at *bStart* this value is first set to 0. As long as the function-block is active, it shows the elapsed test-time. With a falling edge at *bBusy* the output remains at its value, so the time, used for the test, is still available.

nEmergencyMode: During the test this output shows the internally queried value of the Emergency-Mode [▶ 370].

nEmergencyStatus: do. Emergency-Status [▶ 371].

bBusy: If the function-block is active, this output is set to TRUE.

bError: This output is switched to TRUE if an error occurs as described above. Is reset by the execution of a new test.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a new test. See Error codes [▶ 385].

VAR_IN_OUT

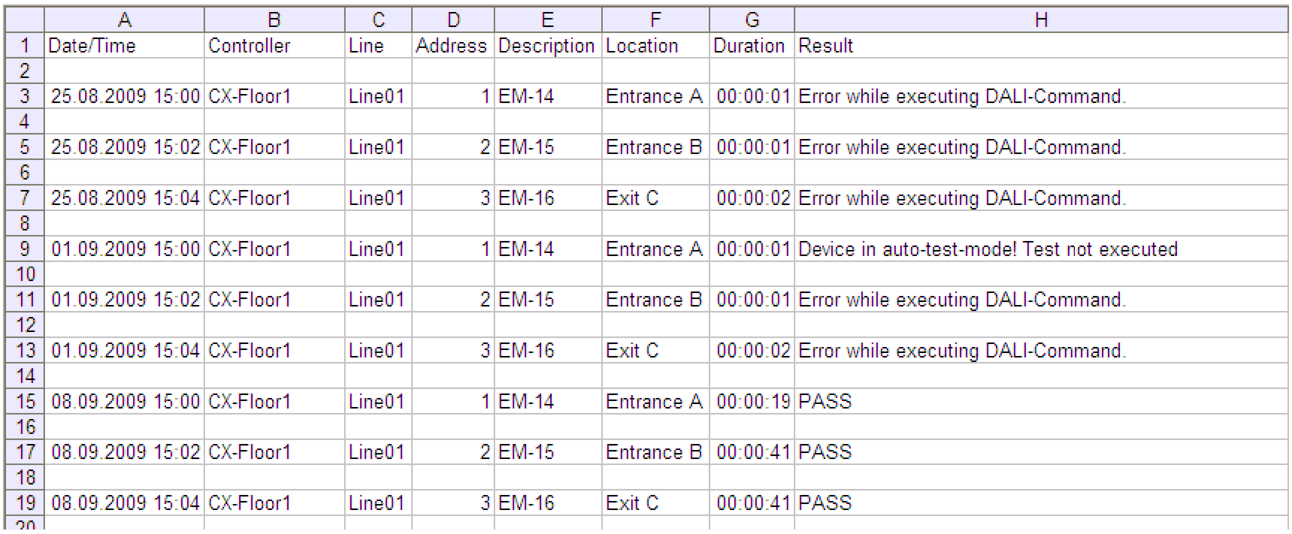
```
stCommandBuffer       : ST_DALIV2CommandBuffer;
fbStringRingBuffer    : FB_MemRingBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

fbStringRingBuffer: Reference to the FIFO-Buffer, in which the log-entries are put.

A https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019292427/.zip shows the correct assignment of the inputs as well as the link to the logger-FB FB_DALIV2FileLogging() [▶ 88]. In this program the Function-Test for the three emergency-devices will be executed every Tuesday.

The log-file will look like this:



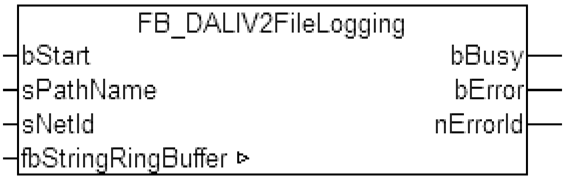
	A	B	C	D	E	F	G	H
1	Date/Time	Controller	Line	Address	Description	Location	Duration	Result
2								
3	25.08.2009 15:00	CX-Floor1	Line01	1	EM-14	Entrance A	00:00:01	Error while executing DALI-Command.
4								
5	25.08.2009 15:02	CX-Floor1	Line01	2	EM-15	Entrance B	00:00:01	Error while executing DALI-Command.
6								
7	25.08.2009 15:04	CX-Floor1	Line01	3	EM-16	Exit C	00:00:02	Error while executing DALI-Command.
8								
9	01.09.2009 15:00	CX-Floor1	Line01	1	EM-14	Entrance A	00:00:01	Device in auto-test-mode! Test not executed
10								
11	01.09.2009 15:02	CX-Floor1	Line01	2	EM-15	Entrance B	00:00:01	Error while executing DALI-Command.
12								
13	01.09.2009 15:04	CX-Floor1	Line01	3	EM-16	Exit C	00:00:02	Error while executing DALI-Command.
14								
15	08.09.2009 15:00	CX-Floor1	Line01	1	EM-14	Entrance A	00:00:19	PASS
16								
17	08.09.2009 15:02	CX-Floor1	Line01	2	EM-15	Entrance B	00:00:41	PASS
18								
19	08.09.2009 15:04	CX-Floor1	Line01	3	EM-16	Exit C	00:00:41	PASS
20								

On 25th August the emergency-devices are not installed, yet. The tests result in DALI-command-errors on all devices.

On 1st September the first device is installed but has still the internal automatic-test activated.
Finally, on 8th September all devices are installed and fully functional. The tests shows good results.

When this file is opened, it may be necessary to adjust the width of the columns, so all the data is visible.

5.1.39 FB_DALIV2FileLogging



This function block reads the respective FIFO buffers (IN-OUT variable *fbStringRingBuffer*) written in the function blocks FB_DALIV2EmergencyLightingFT() [▶ 86] and FB_DALIV2EmergencyLightingDT() [▶ 84] and writes the contents into a log file.

This function block is available only in the PC version of the DALIV2 library.

VAR_INPUT

```
bStart          : BOOL;
sPathName       : STRING;
sNetId          : STRING;
```

bStart: the function block is activated by a positive edge on this input.

sPathName: contains the path and file name of the buffer file to be opened.
The path can only point to the local computer’s file system! This means that network paths cannot be used here!

sNetId: a string containing the network address of the TwinCAT computer where the buffer file is to be written or read can be given here. If it is to be run on the local computer, an empty string can be entered.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
```

bBusy: This output remains TRUE until the Function-block has emptied the log-buffer

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in ‘nErrId’.

nErrorId: Contains the command-specific ADS error code of the most recently executed command.

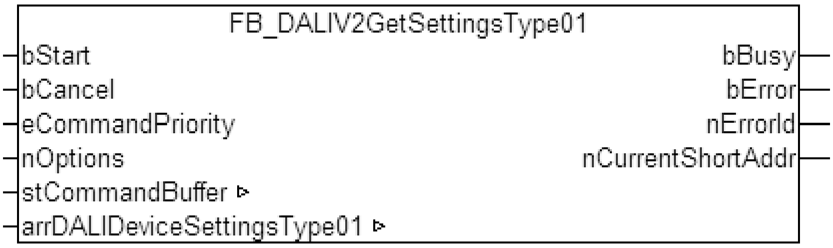
VAR_IN_OUT

```
fbStringRingBuffer    : FB_MemRingBuffer;
```

fbStringRingBuffer: Reference to the FIFO-Buffer, in which the log-entries are put.

A https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019292427/.zip shows the correct assignment of the inputs as well as the link to the logger-FB *FB_DALIV2FileLogging.* The appearance of the log-files is shown under the description of the function-blocks FB_DALIV2EmergencyLightingDT() [▸ 84] and FB_DALIV2EmergencyLightingFT() [▸ 86].

5.1.40 FB_DALIV2GetSettingsType01



This function-block reads the variables (BATTERY CHARGE, DURATION TEST RESULT, LAMP EMERGENCY TIME…) from all emergency lighting ballasts in a DALI line and stores them in a structure of type ST_DALIV2DeviceSettingsType01 [▸ 383].

Applying a positive edge to the *bStart* input starts the block, and the *bBusy* output goes TRUE. A check is first made as to whether a ballast is present at all. If this is the case, then the *bPresent* bit is set in the corresponding structure (see ST_DALIV2DeviceSettingsType01 [▸ 383]), after which the settings of the ballast are read out 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 *arrDALIDeviceSettingsType01[0]*, and so on through to the device with short address 63 having its data at *arrDALIDeviceSettingsType01[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 ballast.

Bit	Error
0	An error occurred whilst attempting to seek the ballast.
1	Error whilst reading the variable BATTERY CHARGE [▶ 370]
2	Error whilst reading the variable DURATION TEST RESULT [▶ 370]
3	Error whilst reading the variable LAMP EMERGENCY TIME [▶ 370]
4	Error whilst reading the variable LAMP TOTAL OPERATION TIME [▶ 370]
5	Error whilst reading the variable EMERGENCY LEVEL [▶ 368]
6	Error whilst reading the variable EMERGENCY MIN LEVEL [▶ 368]
7	Error whilst reading the variable EMERGENCY MAX LEVEL
8	Error whilst reading the variable RATED DURATION [▶ 370]
9	Error whilst reading the variable FUNCTION TEST DELAY TIME [▶ 369]
10	Error whilst reading the variable DURATION TEST DELAY TIME [▶ 369]
11	Error whilst reading the variable FUNCTION TEST INTERVAL [▶ 369]
12	Error whilst reading the variable DURATION TEST INTERVAL [▶ 369]
13	Error whilst reading the variable TEST EXECUTION TIMEOUT [▶ 370]
14	Error whilst reading the variable PROLONG TIME [▶ 368]
15	Error whilst reading the variable EMERGENCY MODE [▶ 370]
16	Error whilst reading the variable FEATURES [▶ 371]
17	Error whilst reading the variable FAILURE STATUS [▶ 371]
18	Error whilst reading the variable EMERGENCY STATUS [▶ 371]

When the function block has been processed, the *bBusy* output changes from TRUE to FALSE. Processing this function block can take several seconds, depending on how many ballasts are attached.

VAR_INPUT

```
bStart             : BOOL;
bCancel            : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityHigh;
nOptions           : DWORD;
```

bStart: The block is activated by a rising edge at this input.

bCancel: A rising edge at this input will deactivate the block and hence abort the reading of the variable.

eCommandPriority: Priority (high, medium or low) with which the command is processed by the library.

nOptions: Options for reading the variables (see table). The individual constants must be linked with OR operators.

Constant	Description
DALIV2_OPTION_BATTERY_CHARGE	The variable BATTERY CHARGE
DALIV2_OPTION_DURATION_TEST_RESULT	The variable DURATION TEST RESULT
DALIV2_OPTION_LAMP_EMERGENCY_TIME	The variable LAMP EMERGENCY TIME
DALIV2_OPTION_LAMP_TOTAL_OPERATION_TIME	The variable LAMP TOTAL OPERATION TIME
DALIV2_OPTION_EMERGENCY_LEVEL	The variable EMERGENCY LEVEL
DALIV2_OPTION_EMERGENCY_MIN_LEVEL	The variable EMERGENCY MIN LEVEL
DALIV2_OPTION_EMERGENCY_MAX_LEVEL	The variable EMERGENCY MAX LEVEL

Constant	Description
DALIV2_OPTION_RATED_DURATION	The variable RATED DURATION
DALIV2_OPTION_NEXT_FUNCTION_TEST	The variable FUNCTION TEST DELAY TIME
DALIV2_OPTION_NEXT_DURATION_TEST	The variable DURATION TEST DELAY TIME
DALIV2_OPTION_FUNCTION_TEST_INTERVAL	The variable FUNCTION TEST INTERVAL
DALIV2_OPTION_DURATION_TEST_INTERVAL	The variable DURATION TEST INTERVAL
DALIV2_OPTION_TEST_EXECUTION_TIMEOUT	The variable TEST EXECUTION TIMEOUT
DALIV2_OPTION_PROLONG_TIME	The variable PROLONG TIME
DALIV2_OPTION_EMERGENCY_MODE	The variable EMERGENCY MODE
DALIV2_OPTION_FEATURES	The variable FEATURES [▶ 371] is read.
DALIV2_OPTION_FAILURE_STATUS	The variable FAILURE STATUS
DALIV2_OPTION_EMERGENCY_STATUS	The variable EMERGENCY STATUS
DALIV2_OPTION_ALL	All variables are read.

VAR_OUTPUT

```
bBusy                 : BOOL;
bError                : BOOL;
nErrorId              : UDINT;
nCurrentShortAddr     : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

nCurrentShortAddr: Short address of the current ballast from which variables are being read.

VAR_IN_OUT

```
stCommandBuffer                : ST_DALIV2CommandBuffer;
arrDALIDeviceSettingsType01    : ARRAY[0..63] OF ST_DALIV2DeviceSettingsType01;
```

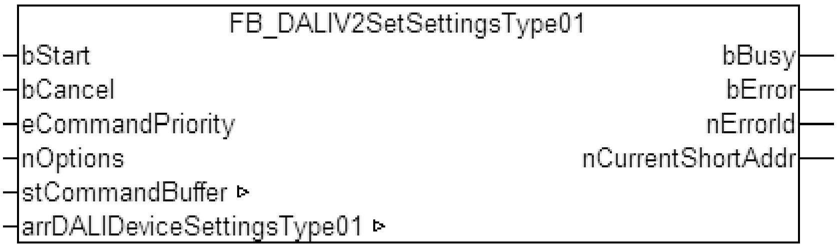
stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

arrDALIDeviceSettingsType01: Reference to an array of 64 elements of type ST_DALIV2DeviceSettingsType01 [▶ 383]. The settings of each individual DALI ballast are stored in these variables.

Also see about this

- E_DALIV2CommandPriority [▶ 381]

5.1.41 FB_DALIV2SetSettingsType01



This function-block initialises the variables (EMERGENCY LEVEL, FUNCTION TEST DELAY TIME, DURATION TEST DELAY TIME…) of all emergency lighting ballasts in a DALI line with the values that are stored in a structure of type ST_DALIV2DeviceSettingsType01 [▶ 383].

Applying a positive edge to the *bStart* input starts the block, and the *bBusy* output goes TRUE. The system first checks whether the *bPresent* bit is set in the respective structure (see ST_DALIV2DeviceSettingsType01 [▶ 383]). If this is the case, all ballast variables that are not write-protected are initialised with the respective values of the structure. The structure index here reflects the address of the ballast. In other words, data for the device with short address 0 is located at *arrDALIDeviceSettingsType01[0]*, and so on through to the ballast with short address 63 having its data at *arrDALIDeviceSettingsType01[63]*. If a write error occurs for a device, the *nErrors* element 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 writing of a variable to a ballast.

Bit	Error
5	Error whilst writing the variable EMERGENCY LEVEL [▶ 368]
9	Error whilst writing the variable FUNCTION TEST DELAY TIME [▶ 369]
10	Error whilst writing the variable DURATION TEST DELAY TIME [▶ 369]
11	Error whilst writing the variable FUNCTION TEST INTERVAL [▶ 369]
12	Error whilst writing the variable DURATION TEST INTERVAL [▶ 369]
13	Error whilst writing the variable TEST EXECUTION TIMEOUT [▶ 370]
14	Error whilst writing the variable PROLONG TIME [▶ 368]

When the function block has been processed, the *bBusy* output changes from TRUE to FALSE. Processing this function block can take several seconds, depending on how many ballasts are attached.

VAR_INPUT

```
bStart                 : BOOL;
bCancel                : BOOL;
eCommandPriority       : E_DALIV2CommandPriority := eDALIV2CommandPriorityHigh;
nOptions               : DWORD;
```

bStart: The block is activated by a rising edge at this input.

bCancel: A rising edge at this input will deactivate the block and hence abort the reading of the variable.

eCommandPriority: Priority (high, medium or low) with which the command is processed by the library.

nOptions: Options for reading the variables (see table). The individual constants must be linked with OR operators.

Constant	Description
DALIV2_OPTION_EMERGENCY_LEVEL	The variable EMERGENCY LEVEL
DALIV2_OPTION_NEXT_FUNCTION_TEST	The variable FUNCTION TEST DELAY TIME
DALIV2_OPTION_NEXT_DURATION_TEST	The variable DURATION TEST DELAY TIME
DALIV2_OPTION_FUNCTION_TEST_INTERVAL	The variable FUNCTION TEST INTERVAL
DALIV2_OPTION_DURATION_TEST_INTERVAL	The variable DURATION TEST INTERVAL
DALIV2_OPTION_TEST_EXECUTION_TIMEOUT	The variable TEST EXECUTION TIMEOUT
DALIV2_OPTION_PROLONG_TIME	The variable PROLONG TIME
DALIV2_OPTION_ALL	All variables are initialised.
DALIV2_OPTION_PUSH_DALI_COMMANDS	The buffer, which contains the feedback telegrams from the ballasts, among other things, is **not** read. Hence, writing becomes faster, but errors are not recognised.

VAR_OUTPUT

```
bBusy                 : BOOL;
bError                : BOOL;
nErrorId              : UDINT;
nCurrentShortAddr     : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

nCurrentShortAddr: Short address of the ballast for which variables are being initialised.

VAR_IN_OUT

```
stCommandBuffer                : ST_DALIV2CommandBuffer;
arrDALIDeviceSettingsType01 : ARRAY[0..63] OF ST_DALIV2DeviceSettingsType01;
```

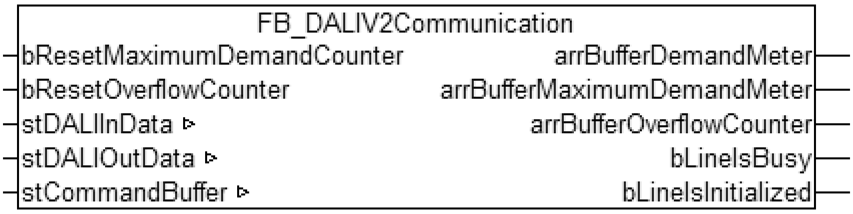
stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

arrDALIDeviceSettingsType01: Reference to an array of 64 elements of type ST_DALIV2DeviceSettingsType01 [▸ 383]. The settings of each individual DALI ballast are stored in these variables.

Also see about this

E_DALIV2CommandPriority [▸ 381]

5.1.42 FB_DALIV2Communication



The blocks for the DALI commands do not access the process image of the KL6811 directly, but place the individual DALI commands in three different buffers. The FB_DALIV2Communication() block sequentially reads the DALI commands from these three buffers and passes the DALI commands to the KL6811. This prevents multiple blocks accessing the KL6811 process image at the same time. Each of these three buffers is processed with a different priority (high, medium or low). The user of the PLC library can use the *eCommandPriority* parameter (available in most blocks) to determine the priority which with the FB_DALIV2Communication() block processes the respective DALI command.

All buffers in which the DALI commands are stored are associated with a variable of type ST_DALIV2CommandBuffer. There is one instance of the FB_DALIV2Communication() block and a variable of type ST_DALIV2CommandBuffer for each KL6811. If possible, the FB_DALIV2Communication() block should be called in a separate, faster task.

The buffer utilisation can be determined via the block outputs based on three arrays in which each element (0, 1 or 2) represents one of the three buffers (high, medium or low). If you detect regular overflow for one of the three buffers, you should consider the following:

- How heavily are the individual PLC tasks utilized? The TwinCAT System Manager offers various analysis tools.

- Try reducing the cycle time of the task in which the FB_DALIV2Communication() block is called. The value should not exceed 6 ms. Ideally it should be 2 ms.

- Check the cycle time of the PLC task in which the blocks for the individual DALI commands are called. This value should be between 10 ms and 60 ms.

- If possible avoid polling (regular reading) of values. Only read values when they are actually required.

- Distribute the individual ballasts evenly over several DALI lines. Overall data throughput is increased by the fact that several DALI lines are processed simultaneously during each PLC cycle.

VAR_INPUT

```
bResetMaximumDemandCounter  : BOOL;
bResetOverflowCounter       : BOOL;
```

bResetMaximumDemandCounter: A rising edge at this input resets the value of the maximum command-buffer-load, *arrBufferMaximumDemandMeter* (0 - 100%, see VAR_OUTPUT).

bResetOverflowCounter: A rising edge at this input resets the value of the total buffer-overflow-counter, *arrBufferOverflowCounter* (see VAR_OUTPUT).

VAR_OUTPUT

```
arrBufferDemandMeter          : ARRAY [0..2] OF BYTE;
arrBufferMaximumDemandMeter   : ARRAY [0..2] OF BYTE;
arrBufferOverflowCounter      : ARRAY [0..2] OF UINT;
bLineIsBusy                   : BOOL;
bLineIsInitialized            : BOOL;
```

arrBufferDemandMeter: Buffer utilisation (0 – 100%).

arrBufferMaximumDemandMeter: Previous maximum utilisation of the respective buffer (0 – 100%).

arrBufferOverflowCounter: Previous number of buffer overflows.

bLineIsBusy: This output is set as long as the FB_DALIV2Communication() block is active.

bLineIsInitialized: If the block is being called for the first time (e.g. when the controller is starting up) an initialisation process is executed. No DALI commands can be processed during this time.

VAR_IN_OUT

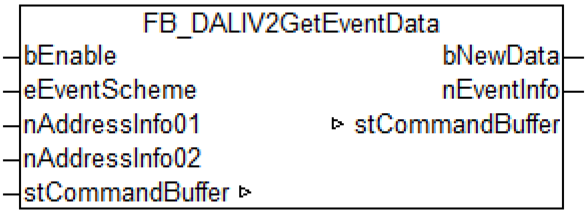
```
stDALIInData            : ST_DALIV2InData;
stDALIOutData           : ST_DALIV2OutData;
stCommandBuffer         : ST_DALIV2CommandBuffer;
```

stDALIInData: Structure [▶ 384] in the input process image of the KL6811. It is used for communication from the KL6811 to the PLC. If FB_KL6811Config() [▶ 99] is used, this structure is linked to the stInData parameter of the function block.

stDALIOutData: Structure [▶ 384] in the output process image of the KL6811. It is used for communication from the PLC to the KL6811. If FB_KL6811Config() [▶ 99] is used, this structure is linked to the *stOutData* parameter of the function block.

stCommandBuffer: A reference to the internal structure for communication with the DALI function blocks.

5.1.43 FB_DALIV2GetEventData



Filters out an event specified by the event scheme.

Each event sent by a DALI device contains two fields that provide information about the event source. These two fields are a combination of the short address, instance number, instance type, instance group or device group. The recipient of the event must know which address scheme is used to send the data.

For each event that is to be received and processed further, an instance of FB_DALIV2GetEventData() must be created and configured with the correct event scheme.

VAR_INPUT

```
bEnable          : BOOL;
eEventScheme     : E_DALIV2EventScheme := eDALIV2EventSchemeDeviceInstance;
nAddressInfo01   : BYTE;
nAddressInfo02   : BYTE;
```

bEnable: Enables the function block. If this input is set to FALSE, no further events are output.

eEventScheme: The event scheme [▸ 382] defines the address information required for filtering the desired result.

nAddressInfo01: (see table below)

nAddressInfo02: (see table below)

eEventScheme	nAddressInfo01	nAddressInfo02
eDALIV2EventSchemeInstance	Instance type (0-31)	Instance number (0-31)
eDALIV2EventSchemeDevice	Short address (0-63)	Instance type (0-31)
eDALIV2EventSchemeDeviceInstance	Short address (0-63)	Instance number (0-31)
eDALIV2EventSchemeDeviceGroup	Device group (0-31)	Instance type (0-31)
eDALIV2EventSchemeInstanceGroup	Instance group (0-31)	Instance type (0-31)

VAR_OUTPUT

```
bNewData      : BOOL;
nEventInfo    : WORD;
```

bNewData: If an event was received that corresponds to the event scheme and address information, this output is set to TRUE for one PLC cycle.

nEventInfo: If the output *bNewData* 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.

VAR_IN_OUT

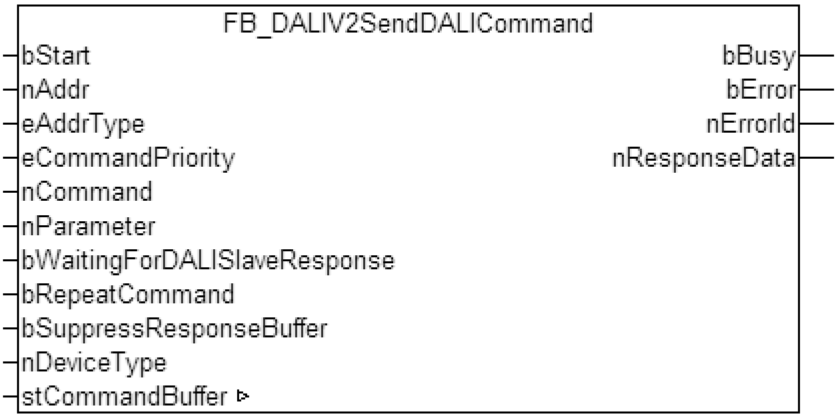
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.44 FB_DALIV2SendDALICommand



This function-block sends a single DALI-command defined by the command-number and, if applicable, by a parameter-value. Moreover it is possible it is possible to decide, if the command is to be sent twice and whether the function-block has to wait for an answer. The last mentioned option is very useful to send fast sequences of commands which do not require an answer from the device, such as the step-up-command.

VAR_INPUT

```
bStart                       : BOOL;
nAddr                        : BYTE;
eAddrType                    : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority             : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nCommand                     : INT;
nParameter                   : BYTE;
bWaitingForDALISlaveResponse : BOOL;
bRepeatCommand               : BOOL;
bSuppressResponseBuffer      : BOOL;
nDeviceType                  : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nCommand: The number of the DALI-command.

nParameter: Parameter to hand over command-specific values, eg. *go to scene xxx*.

bWaitingForDALISlaveResponse: If this input is set to FALSE, the function-block will not wait for an answer of the DALI-device. This feature doesn´t make any sense in combination with query-commands, of course.

bRepeatCommand: The command will be repeated, if this input is set to TRUE.

bSuppressResponseBuffer: If set to TRUE, the answer of the function block FB_DALIV2Communication() [▸ 93] will not be put into the internal software-buffer.

nDeviceType: Identifier for the device type.

Value	Description
0	Standard device
1	Device for emergency lighting [▸ 32].
2	Device for discharge lamps [▸ 36].
3	Device for low-voltage halogen lamps.
4	Device for dimming bulbs.
5	Device for converting digital signals into DC signals.

Value	Description
6	Device for light emitting diodes (LEDs) [▶ 36].
7	Switching function.
8	Device for colour/colour temperature control [▶ 34].

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nResponseData      : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

nResponseData: The received value of the DALI ballast, if a query command is invoked.

VAR_IN_OUT

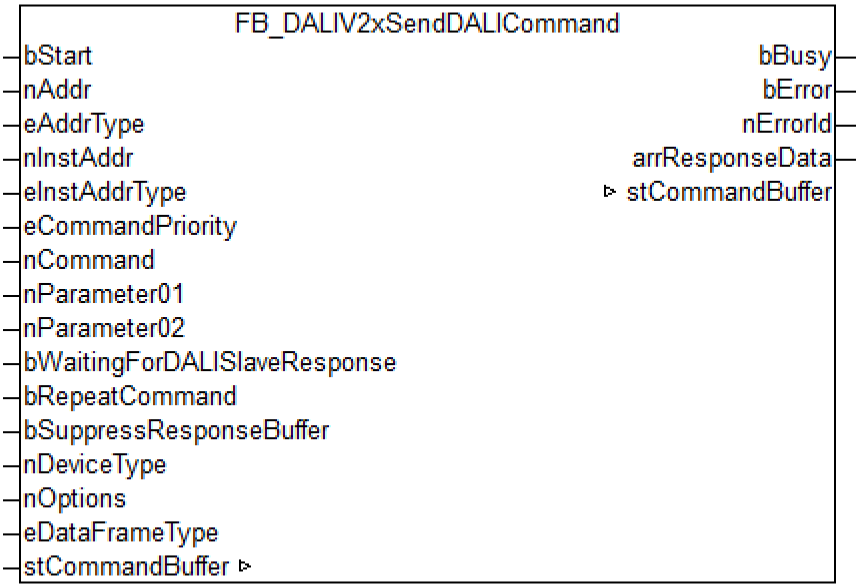
```
stCommandBuffer           : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Also see about this

- FB_DALIV2xSendDALICommand [▶ 97]
- FB_KL6821Communication [▶ 101]

5.1.45 FB_DALIV2xSendDALICommand



This function block is for the general sending of a DALI command, defined by command number and, if necessary, transfer parameter. Moreover, it is possible to set whether the command is sent twice in succession and whether to wait for a response. The latter can be used, for example, to realize a fast sequence of step-up commands.

Compared to the function block FB_DALIV2SendDALICommand() [▸ 96], this function block is also able to control DALI control units (sensors).

VAR_INPUT

```
bStart                       : BOOL;
nAddr                        : BYTE;
eAddrType                    : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr                    : BYTE := 0;
eInstAddrType                : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority             : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nCommand                     : INT := 0;
nParameter01                 : BYTE := 0;
nParameter02                 : DINT := 0;
bWaitingForDALISlaveResponse : BOOL := FALSE;
bRepeatCommand               : BOOL := FALSE;
bSuppressResponseBuffer      : BOOL := FALSE;
nDeviceType                  : BYTE := 0;
nOptions                     : DWORD := 0;
eDataFrameType               : E_DALIV2DataFrameType := eDALIV2DataFrameType16Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the device.

eInstAddrType: Defines the meaning of the variable nInstAddr for addressing the instance (e. g. by instance number, instance type,...)

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nCommand: Number of the DALI command to be sent.

nParameter01: Parameter for the value transfer.

nParameter02: Parameter for the value transfer.

bWaitingForDALISlaveResponse: If *FALSE*, the system does **not** wait for the answer from the DALI device. Its application makes no sense in connection with any kind of query command.

bRepeatCommand: Decides whether the command is to be sent twice in succession.

bSuppressResponseBuffer: If *TRUE*, the internal buffer is **not** filled with the response from the function block FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

nDeviceType: Identifier for the device type.

Value	Description
0	Standard device
1	Device for emergency lighting [▸ 32].
2	Device for discharge lamps [▸ 36].
3	Device for low-voltage halogen lamps.
4	Device for dimming incandescent lamps.
5	Device for converting digital signals into DC signals.
6	Device for light emitting diodes (LEDs) [▸ 36].
7	Device for switching functions.
8	Device for controlling the color/color temperature [▸ 34].
9	Sequencer.

nOptions: reserved for future expansions.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit*, *eDALIV2DataFrameType16Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy                   : BOOL;
bError                  : BOOL;
nErrorId                : UDINT;
arrResponseData         : ARRAY [0..3] OF BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

arr**ResponseData:** The value received from the DALI device if a query command was invoked.

VAR_IN_OUT

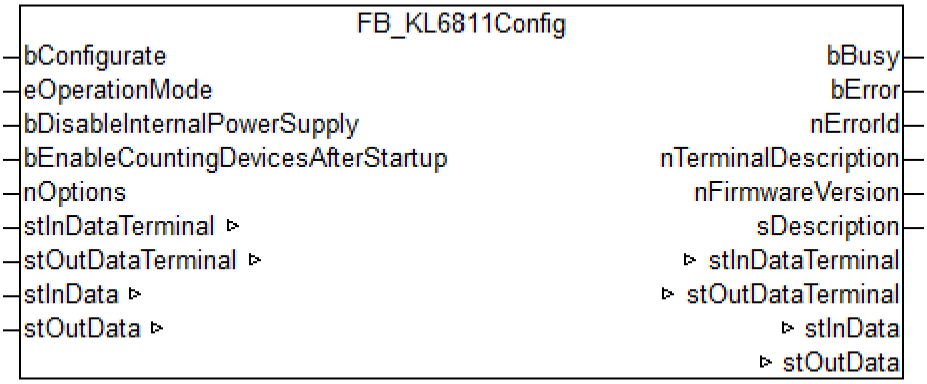
```
stCommandBuffer             : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

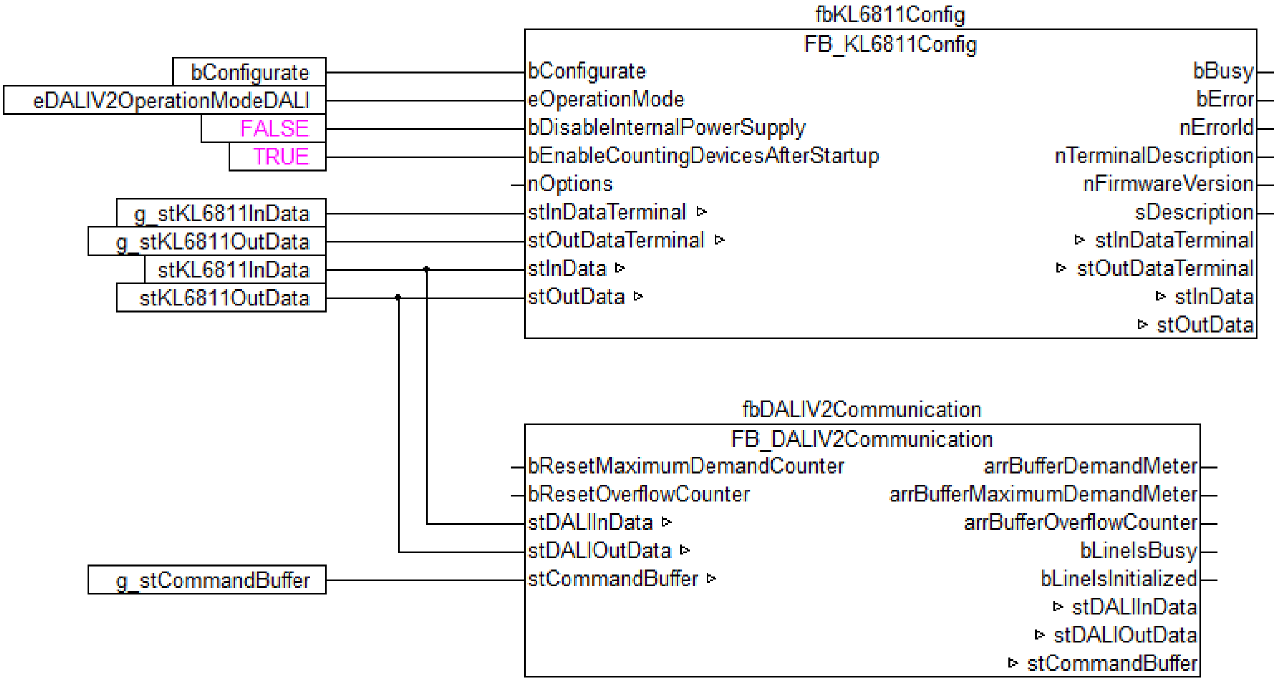
5.1.46 FB_KL6811Config



This block is used to configure the KL6811. The configuration is executed when the PLC program starts, or it can be triggered by a positive edge at the input *bConfigurate*. The parameters are stored in the respective registers of the KL6811 in a fail-safe manner. In addition, some general information, such as the firmware version, is read from the KL6811.

Example

The block is called in the same task as the block FB_DALIV2Communication() [▶ 93].



The block FB_KL6811Config() is linked to the process image of the KL6811. After the configuration, the block FB_DALIV2Communication() is assigned to the process values of the KL6811. During the configuration, it is not possible to send DALI commands.

Unpacking the example files https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/

12019293835/.zip

VAR_INPUT

```
bConfigurate                         : BOOL := FALSE;
eOperationMode                       : E_DALIV2OperationMode := eDALIV2OperationModeDALI;
bDisableInternalPowerSupply          : BOOL := FALSE;
bEnableCountingDevicesAfterStartup   : BOOL := FALSE;
nOptions                             : DWORD := 0;
```

bConfigurate: Configuration of the Bus Terminal is started by a positive edge at this input.

eOperationMode: Defines the operating mode [▸ 382] of the terminal (DALI or DSI). Corresponds to register 32, bits 12 to 15 of the Bus Terminal.

bDisableInternalPowerSupply: If this input is TRUE, the internal DALI power supply unit of the terminal is disabled during the configuration. Corresponds to register 32, bit 3 of the Bus Terminal.

bEnableCountingDevicesAfterStartup: If this input is TRUE, the number of DALI devices is counted when the terminal starts. Corresponds to register 32, bit 4 of the Bus Terminal.

nOptions: Reserved for future expansions.

VAR_OUTPUT

```
bBusy                    : BOOL;
bError                   : BOOL;
nErrorId                 : UDINT;
nTerminalDescription     : WORD;
nFirmwareVersion         : WORD;
sDescription             : STRING;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. The output is again set to FALSE by the reactivation of the function block via the *bConfigurate* input.

nErrorId: Contains the command-specific error code of the most recently executed command. It is set back to 0 by the reactivation of the function block via the *bConfigurate* input. See Error codes [▸ 385].

nTerminalDescription: Contains the terminal designation (e.g. 6811). Corresponds to register 8 of the Bus Terminal.

nFirmwareVersion: Contains the firmware version. Corresponds to register 9 of the Bus Terminal.

sDescription: Terminal designation and firmware version as string (e.g. 'Terminal KL6811 / Firmware 2H').

VAR_IN_OUT

```
stInDataTerminal           : ST_DALIV2InData;
stOutDataTerminal          : ST_DALIV2OutData;
stInData                   : ST_DALIV2InData;
stOutData                  : ST_DALIV2OutData;
```

stInDataTerminal: Reference to the structure [▸ 384] for communication with the KL6811.

stOutDataTerminal: Reference to the structure [▸ 384] for communication with the KL6811.

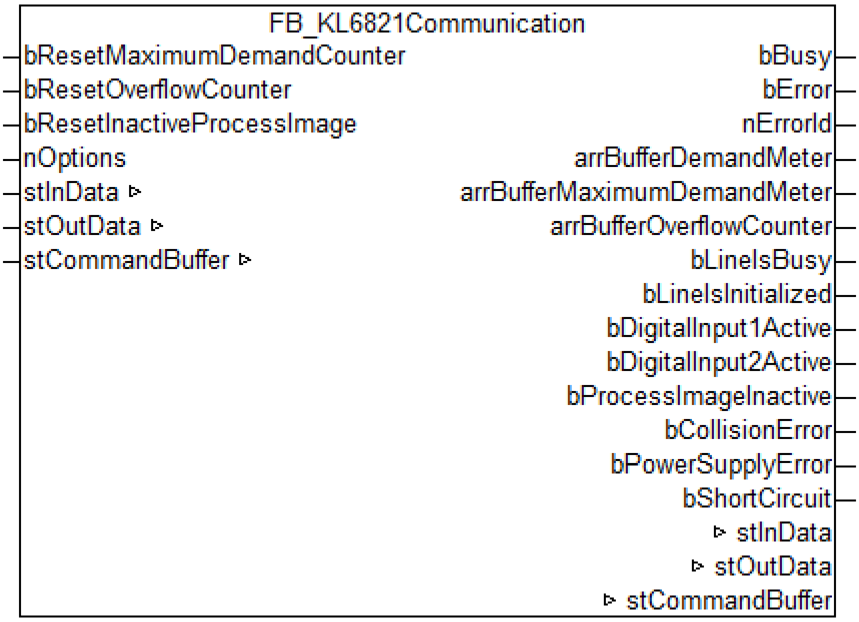
stInData: Reference to the structure for communication with the function block FB_DALIV2Communication() [▸ 93].

stOutData: Reference to the structure for communication with the function block FB_DALIV2Communication() [▸ 93].

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2253	PC/CX, BX or BC	TcDALIV2 library from V2.11.1

5.1.47 FB_KL6821Communication



The function blocks for the DALI commands do not directly access the process image of the DALI Bus Terminal, but store the individual DALI commands in three different buffers. The function block FB_KL6821Communication() sequentially reads the DALI commands from these three buffers and passes the DALI commands to the KL6821. This prevents several function block accessing the process image of the Bus Terminal at the same time. Each of these three buffers is processed with a different priority (high, medium or low). The parameter *eCommandPriority*, which is available for most function blocks, can be used by the PLC library user to influence the priority with which the respective DALI command is processed by the function block FB_KL6821Communication().

All buffers in which the DALI commands are stored are associated with a variable of type ST_DALIV2CommandBuffer. For each KL6821 there is one instance of the function block FB_KL6821Communication() and one variable of type ST_DALIV2CommandBuffer. If possible, the FB_KL6821Communication() function block should be called in a separate, faster task.

The extent to which the buffers are utilized can be determined from the outputs of the function block. Three arrays are output for this in which each element (0, 1 or 2) represents one of the three buffers (high, middle or low). If you find that one of the three buffers overflows on a regular basis, you should consider the following measures:

- How heavily are the individual PLC tasks utilized? The TwinCAT System Manager offers various appropriate utilities for the analysis.

- Try to reduce the cycle time of the task in which the function block FB_KL6821Communication() is called. The value should not exceed 6 ms. Ideally it should be 2 ms.

- Check the cycle time of the PLC task in which the function blocks for the individual DALI commands are called. This value should be between 10 ms and 60 ms.

- If possible avoid polling (regular reading) of values. Only read values when they are actually required.

- Distribute the individual ballasts evenly over several DALI lines. Overall data throughput is increased by the fact that several DALI lines are processed simultaneously during each PLC cycle.

VAR_INPUT

```
bResetMaximumDemandCounter    : BOOL;
bResetOverflowCounter         : BOOL;
bResetInactiveProcessImage    : BOOL;
nOptions                      : DWORD := 0;
```

bResetMaximumDemandCounter: a positive edge resets the stored value of the maximum command buffer utilization, *arrBufferMaximumDemandMeter* (0 - 100%, see VAR_OUTPUT).

bResetOverflowCounter: a positive edge resets the stored value of the number of command buffer overflows, *arrBufferOverflowCounter* (see VAR_OUTPUT).

bResetInactiveProcessImage: A positive edge cancels the blocking of the process image of the terminal. The *bProcessImageInactive*, *bDigitalInput1Active* and *bDigitalInput2Active* outputs are again set to FALSE. The lock is activated when one of the two digital inputs on the terminal is activated.

nOptions: reserved for future expansions.

VAR_OUTPUT

```
bBusy                         : BOOL;
bError                        : BOOL;
nErrorId                      : UDINT;
arrBufferDemandMeter          : ARRAY [0..2] OF BYTE;
arrBufferMaximumDemandMeter   : ARRAY [0..2] OF BYTE;
arrBufferOverflowCounter      : ARRAY [0..2] OF UINT;
bLineIsBusy                   : BOOL;
bLineIsInitialized            : BOOL;
bDigitalInput1Active          : BOOL;
bDigitalInput2Active          : BOOL;
bProcessImageInactive         : BOOL;
bCollisionError               : BOOL;
bPowerSupplyError             : BOOL;
bShortCircuit                 : BOOL;
```

bBusy: This output is set as soon as the function block processes a command and remains active until the command has been processed.

bError: This output is switched to TRUE if an error has occurred during execution of the function block. The command-specific error code is contained in *nErrorId*.

nErrorId: Contains the command-specific error code of the most recently executed command. See Error codes [▶ 385].

arrBufferDemandMeter: Occupancy of the respective buffer (0 - 100%).

arrBufferMaximumDemandMeter: previous maximum occupancy of the respective buffer (0 - 100%).

arrBufferOverflowCounter: Number of buffer overflows to date.

bLineIsBusy: The output is set as long as the function block FB_KL6821Communication() is active.

bLineIsInitialized: if the function block is being called for the first time (e.g. when the controller is starting up) an initialization process is executed. No DALI commands can be processed during this time.

bDigitalInput1Active: The digital input 1 on the terminal was or is actuated (see also terminal documentation). The *bProcessImageInactive* output is set and no further DALI commands can be processed by the controller.

bDigitalInput2Active: The digital input 2 on the terminal was or is actuated (see also terminal documentation). The *bProcessImageInactive* output is set and no further DALI commands can be processed by the controller.

bProcessImageInactive: One of the two digital inputs was actuated at the terminal. No further DALI commands can be processed by the controller. The blockage must be released again via the *bResetInactiveProcessImage* input.

bCollisionError: A data collision on the DALI bus was detected while a command was sent.

bPowerSupplyError: The KL6821 has detected an error in the internal DALI power supply.

bShortCircuit: Short circuit on the DALI bus.

VAR_IN_OUT

```
stInData                : ST_KL6821InData;
stOutData               : ST_KL6821OutData;
stCommandBuffer         : ST_DALIV2CommandBuffer;
```

stInData: Structure [▶ 384] in the input process image of the KL6821. It is used for communication from the KL6821 to the PLC. When using FB_KL6821Config() [▶ 104], this structure is linked to the parameter *stInData*.

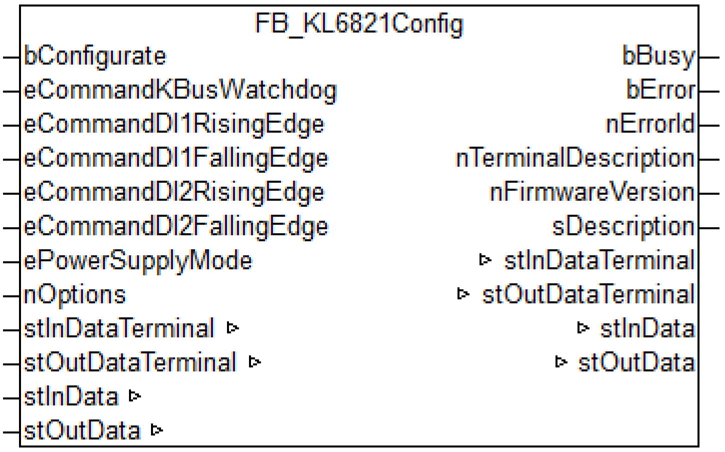
stOutData: Structure [▶ 384] in the output process image of the KL6821. It is used for communication from the KL6821 to the PLC. When using FB_KL6821Config() [▶ 104], this structure is linked to the parameter *stOutData*.

stCommandBuffer: A reference to the internal structure for communication with the DALI function blocks.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

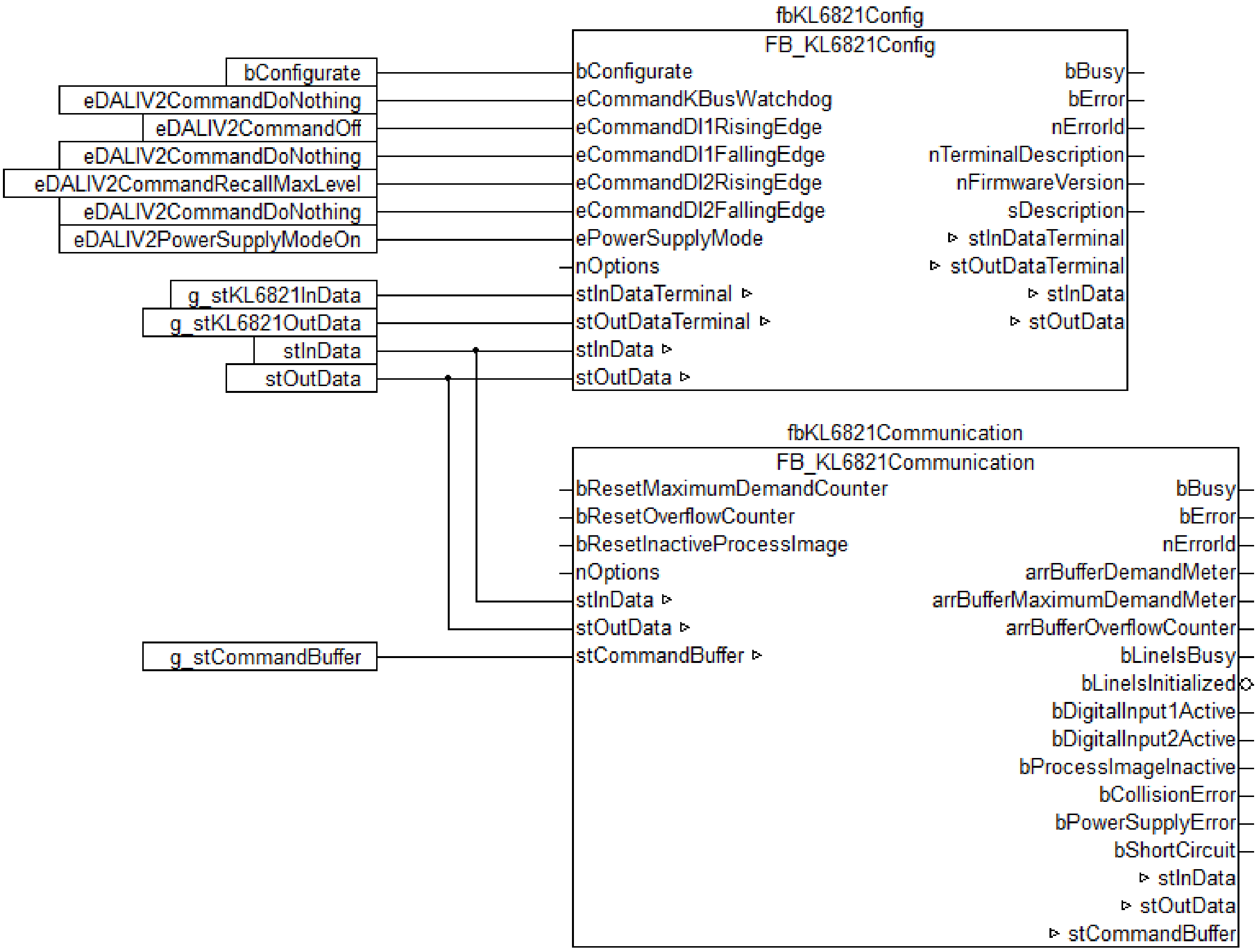
5.1.48 FB_KL6821Config



This function block is used to configure the KL6821. The configuration is executed when the PLC program starts, or it can be triggered by a positive edge at the input *bConfigurate*. The parameters are stored in the respective registers of the KL6821 in a fail-safe manner. In addition, some general information, such as the firmware version, is read from the KL6821.

Example

The function block is called in the same task as the function block FB_KL6821Communication() [▶ 101].



The function block FB_KL6821Config() is linked to the process image of the KL6821. Once the configuration is complete, the function block FB_KL6821Communication() receives the process values of the KL6821. DALI commands cannot be sent during configuration.

Unpacking the example files https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019295243/.zip

VAR_INPUT

```
bConfigurate                  : BOOL := FALSE;
eCommandKBusWatchdog          : E_DALIV2ConfigurationCommands := eDALIV2CommandDoNothing;
eCommandDI1RisingEdge         : E_DALIV2ConfigurationCommands := eDALIV2CommandOff;
eCommandDI1FallingEdge        : E_DALIV2ConfigurationCommands := eDALIV2CommandDoNothing;
eCommandDI2RisingEdge         : E_DALIV2ConfigurationCommands := eDALIV2CommandRecallMaxLevel;
eCommandDI2FallingEdge        : E_DALIV2ConfigurationCommands := eDALIV2CommandDoNothing;
ePowerSupplyMode              : E_DALIV2PowerSupplyMode := eDALIV2PowerSupplyModeOn;
nOptions                      : DWORD := 0;
```

bConfigurate: Configuration of the Bus Terminal is started by a positive edge at this input.

eCommandKBusWatchdog: Defines the DALI command [▸ 381] that is sent as soon as the Bus Terminal is no longer addressed via the K-bus.

eCommandDI1RisingEdge: Defines the DALI command that is sent as soon as a rising edge is detected at input 1 of the Bus Terminal.

eCommandDI1FallingEdge: Defines the DALI command that is sent as soon as a falling edge is detected at input 1 of the Bus Terminal.

eCommandDI2RisingEdge: Defines the DALI command that is sent as soon as a rising edge is detected at input 2 of the Bus Terminal.

eCommandDI2FallingEdge: Defines the DALI command that is sent as soon as a falling edge is detected at input 2 of the Bus Terminal.

ePowerSupplyMode: Defines the operation mode [▸ 382] of the internal DALI power supply.

nOptions: reserved for future expansions.

VAR_OUTPUT

```
bBusy                 : BOOL;
bError                : BOOL;
nErrorId              : UDINT;
nTerminalDescription  : WORD;
nFirmwareVersion      : WORD;
sDescription          : STRING;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Reactivating the function block via the *bConfigurate* input sets the output to FALSE again.

nErrorId: Contains the command-specific error code of the most recently executed command. It is reset to 0 by activating the function block again via the input *bConfigurate*. See Error codes [▸ 385].

nTerminalDescription: Contains the terminal designation (e.g. 6821). Corresponds to register 8 of the Bus Terminal.

nFirmwareVersion: Contains the firmware version. Corresponds to register 9 of the Bus Terminal.

sDescription: Terminal designation and firmware version as string (e.g. 'Terminal KL6821 / Firmware 2H').

VAR_IN_OUT

```
stInDataTerminal         : ST_KL6821InData;
stOutDataTerminal        : ST_KL6821OutData;
stInData                 : ST_KL6821InData;
stOutData                : ST_KL6821OutData;
```

stInDataTerminal: Reference to the structure [▸ 385] for communication with the KL6821.

stOutDataTerminal: Reference to the structure [▸ 385] for communication with the KL6821.

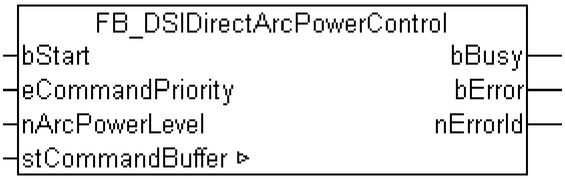
stInData: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

stOutData: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.49 FB_DSIDirectArcPowerControl



The parameter nArcPowerLevel specifies the brightness, the lamp is switched to.

nArcPowerLevel	Comment
0	lamp is switched off
1	lamp is switched to minimum brightness
255	lamp is switched to maximum brightness

Ballasts with DSI interface have no short address. All ballasts connected to a DSI line will get the same value.

Pleas pay attention, that the KL6811 has to be switched to DSI mode. Details may be found in the KL6811 manual.

DSI ballasts and DALI ballasts can not be nixed in the same line. But it is possible to operate several KL6811 with different operation modes (DSI/DALI) at the same controller.

VAR_INPUT

```
bStart               : BOOL;
eCommandPriority     : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nArcPowerLevel       : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nArcPowerLevel: Lamp power level (0 ... 255).

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

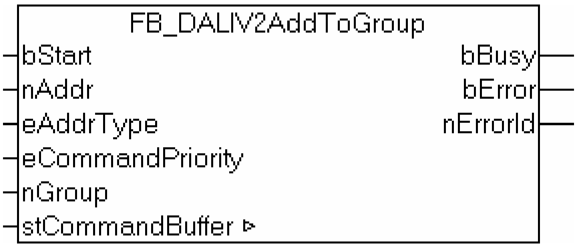
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.50 FB_DALIV2AddToGroup



The ballasts addressed are inserted into the corresponding group (*nGroup*). A valid group number lies in the range between 0 and 15.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nGroup              : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nGroup: Group number (0 - 15).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

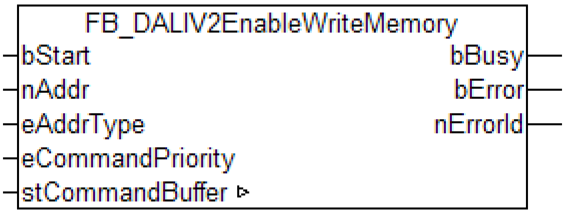
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.51 FB_DALIV2EnableWriteMemory



Enables write access via FB_DALIV2WriteMemoryLocation() [▶ 167] to the internal memory of the control gear.

This command can only be executed by DALI devices that comply with the IEC 62386 standard.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

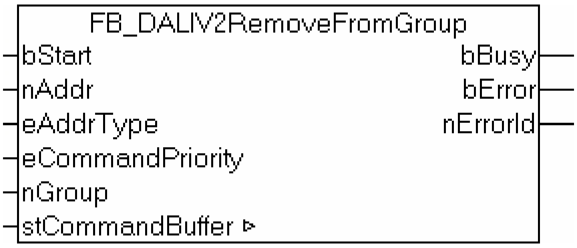
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.52 FB_DALIV2RemoveFromGroup



One or more ballasts are removed from a group. A valid group number lies in the range between 0 and 15.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nGroup              : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nGroup: The group from which the ballast is to be removed.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

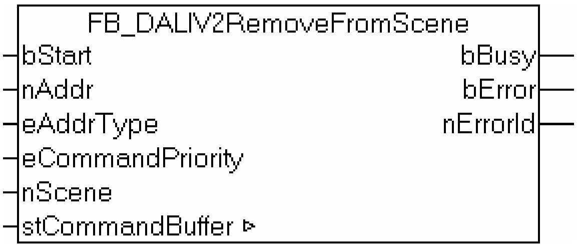
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.53 FB_DALIV2RemoveFromScene



One or more ballasts are removed from a scene. A valid scene number lies in the range between 0 and 15.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nScene              : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nScene: The scene from which the ballast is to be removed.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

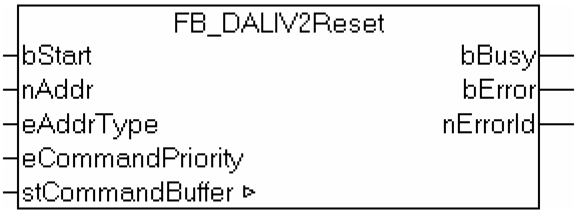
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.54 FB_DALIV2Reset



All the ballast's variables are reset to their standard values with this block.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

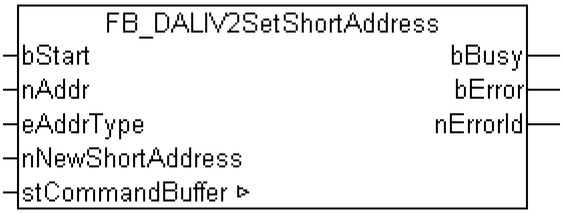
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.55 FB_DALIV2SetShortAddress



One or more ballasts receive a (new) short address with the aid of this block. Valid short addresses lie in the range between 0 and 63. If 255 is specified as the short address, the ballast's short address is deleted.
If you want to give a short address to a device that does not yet have one, you must transmit the command as a broadcast (eAddrType = eDALIV2AddrTypeBroadcast). This gives all the ballasts that are connected to the DALI terminal the short address *nAddr*. This includes the ballasts that did not so far have a short address.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nNewShortAddress    : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nNewShortAddress: New short address (0-15) or mask (255).

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

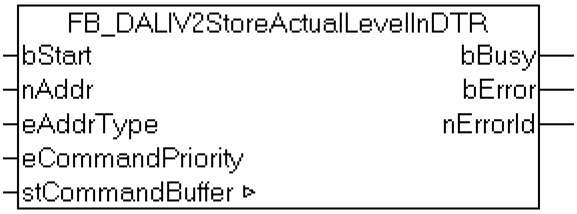
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.56 FB_DALIV2StoreActualLevelInDTR



The block writes the current value of the lamp power into the DTR. This does not change the current value of the lamp power.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

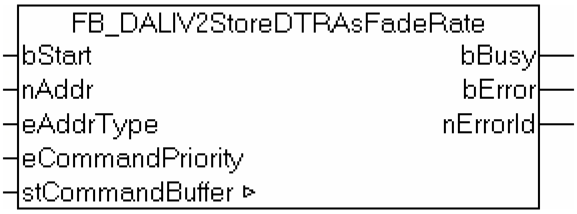
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.57 FB_DALIV2StoreDTRAsFadeRate



Writes the value in the DTR into the FADE RATE [▶ 364] variable. The range of possible values extends from 1 to 15.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

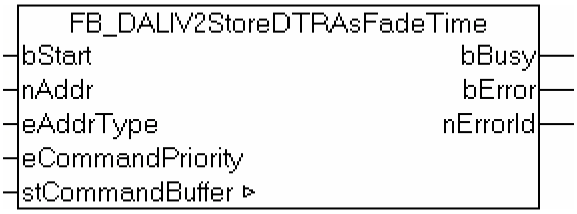
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.58 FB_DALIV2StoreDTRAsFadeTime



The block writes the value of the DTR into the FADE TIME [▶ 365] variable. The range of possible values extends from 0 to 15.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

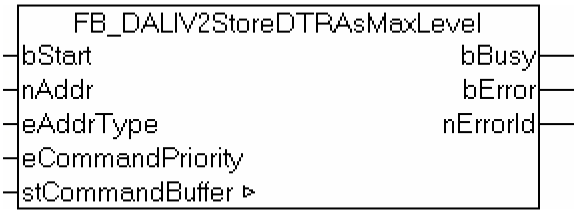
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.59 FB_DALIV2StoreDTRAsMaxLevel



This block writes the value of the DTR into the MAX LEVEL [▸ 364] variable. If the value provided is smaller than MIN LEVEL then the value is simply set to MIN LEVEL.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

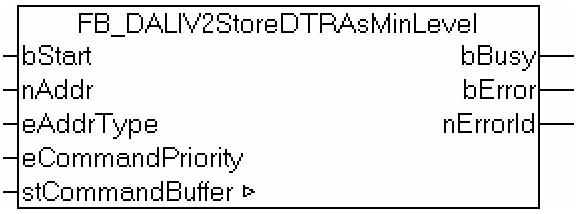
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.60 FB_DALIV2StoreDTRAsMinLevel



This block writes the value of the DTR into the MIN LEVEL [▶ 364] variable (the minimum permitted lamp power). If the value provided is larger than MAX LEVEL then the value is simply set to MAX LEVEL. If the value provided is smaller than the PHYSICAL MIN LEVEL [▶ 367] then MIN LEVEL will be set to PHYSICAL MIN LEVEL.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

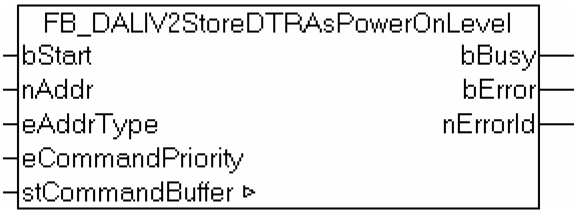
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.61 FB_DALIV2StoreDTRAsPowerOnLevel



This block writes the value of the DTR into the POWER ON LEVEL [▶ 364] variable.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

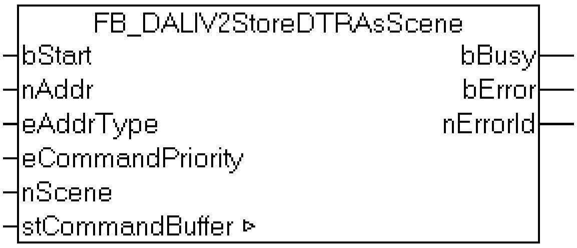
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.62 FB_DALIV2StoreDTRAsScene



The content of the DTR is saved as the value of the lamp power for the given scene. The range of values for the scene number extends from 0 to 15.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
nScene            : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

nScene: The scene for which the value of the lamp power should be changed.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

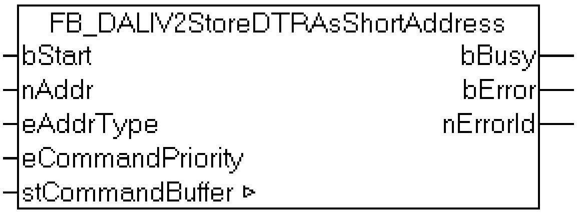
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.63 FB_DALIV2StoreDTRAsShortAddress



The content of the DTR (Data Transfer Register) is saved as the short address at the corresponding ballast. The structure of the DTR is 0AAA AAA1 (A: significant address bit) or 1111 1111 (mask). If the DTR contains 1111 1111 the short address is deleted from the ballast.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

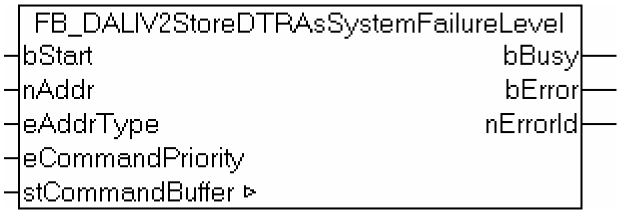
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.64 FB_DALIV2StoreDTRAsSystemFailureLevel



This block writes the value of the DTR into the SYSTEM FAILURE LEVEL [▸ 364] variable.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

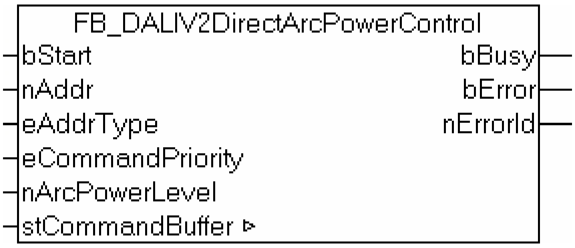
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.65 FB_DALIV2DirectArcPowerControl



If the *nArcPowerLevel* parameter is not within the range between MAX VALUE [▶ 364] and MIN VALUE the lamp is switched to the corresponding minimum or maximum value. If the lamp is switched off this command will switch it on.
The speed with which the specified value should be reached is given by the FADE TIME [▶ 365] variable.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nArcPowerLevel     : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nArcPowerLevel: Lamp power level.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

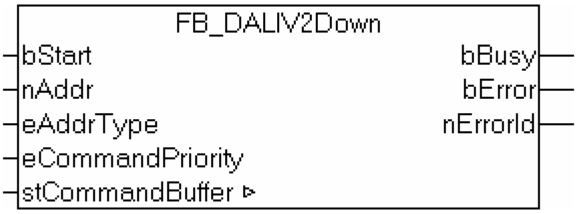
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.66 FB_DALIV2Down



The lamp is made dimmer over a period of 200 ms. If the lamp power has already reached the MIN LEVEL [▶ 364] value the brightness is not changed. This command does not switch the lamp off.
The rate at which dimming takes place during these 200 ms is given by the FADE RATE [▶ 364] variable.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

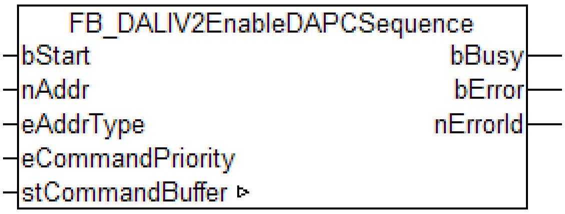
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.67 FB_DALIV2EnableDAPCSequence



This command starts a *Direct Arc Power Control* (DAPC) sequence. Following this command, DirectArcPowerControl commands must be sent using the function block FB_DALIV2DirectArcPowerControl() [▸ 119]. There must not be any more than 200 ms between the individual commands; otherwise the sequence will be ended.

This command can only be executed by DALI devices that comply with the IEC 62386 standard.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

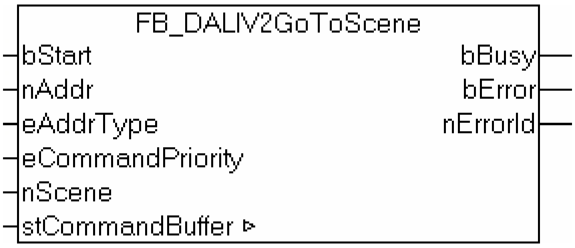
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.68 FB_DALIV2GoToScene



This block sets the value of the lamp power that has been saved for the scene *nScene*. If the ballast does not belong to the scene, the value of the lamp's power is not changed. If the lamp is switched off this command will switch it on.
The speed with which the lamp power should be reached is given by the FADE TIME [▶ 365] variable.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nScene              : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nScene: The scene that is to be activated (0 - 15).

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

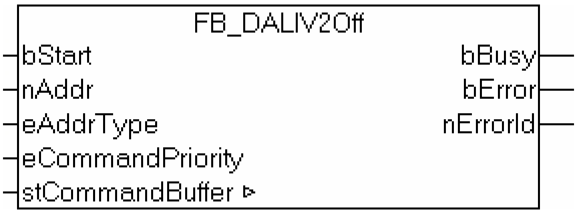
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.69 FB_DALIV2Off



The DALI lamps are switched off immediately.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

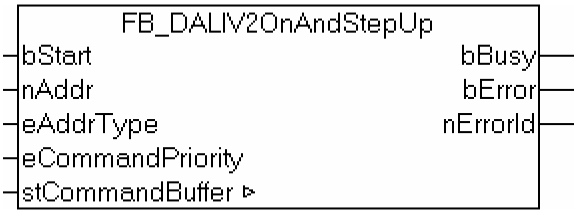
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.70 FB_DALIV2OnAndStepUp



This block immediately sets the current lamp power value one step higher. If the lamp is switched off then it is switched on and set to MIN LEVEL [▶ 364].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

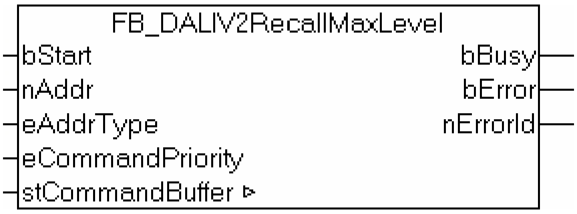
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.71 FB_DALIV2RecallMaxLevel



This block sets the current lamp power smoothly to a MAX LEVEL [▶ 364]. If the lamp is switched off this command will switch it on.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

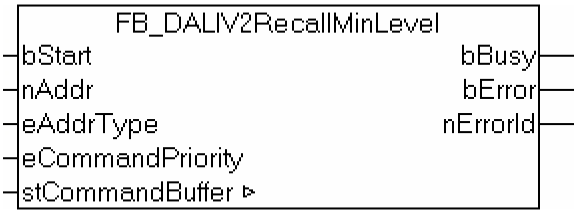
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.72 FB_DALIV2RecallMinLevel



This block sets the current lamp power smoothly to a MIN LEVEL [▶ 364]. If the lamp is switched off this command will switch it on.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

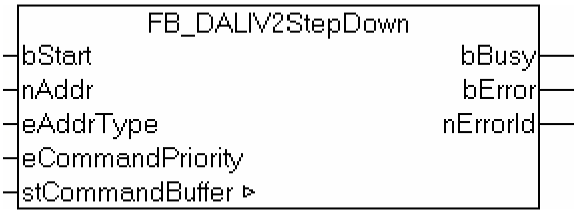
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.73 FB_DALIV2StepDown



This block immediately sets the current lamp power value one step lower. The lamp is not switched off by this command. The power is not further reduced if the lamp power has already reached MIN LEVEL [▶ 364].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

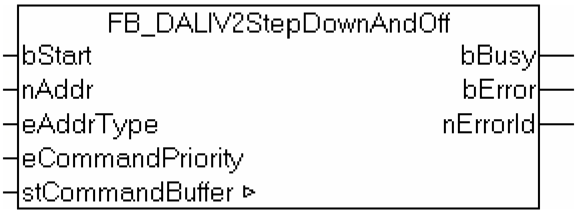
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.74 FB_DALIV2StepDownAndOff



This block immediately sets the current lamp power value one step lower. The lamp is switched off if the power value has already reached MIN LEVEL [▶ 364].

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

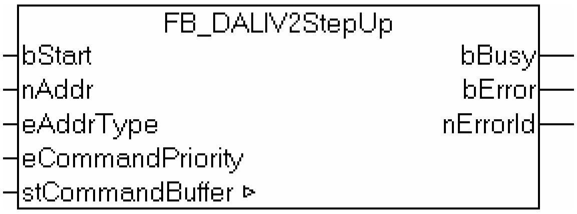
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.75 FB_DALIV2StepUp



This block immediately sets the current lamp power value one step higher. The lamp is not switched on by this command. The power is not further increased if the lamp power has already reached MAX LEVEL [▶ 364].

VAR_INPUT

```
bStart                : BOOL;
nAddr                 : BYTE;
eAddrType             : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority      : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

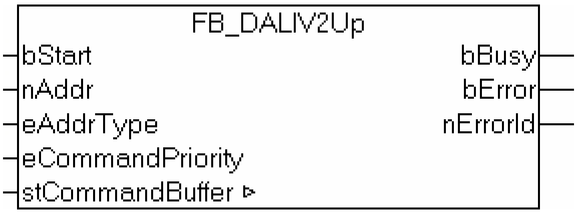
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.76 FB_DALIV2Up



The lamp is made brighter over a period of 200 ms. If the lamp power has already reached the MAX LEVEL [▶ 364] value the brightness is not changed. This command does not switch the lamp on.
The rate at which dimming takes place during these 200 ms is given by the FADE RATE [▶ 364] variable.

VAR_INPUT

```
bStart                : BOOL;
nAddr                 : BYTE;
eAddrType             : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority      : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

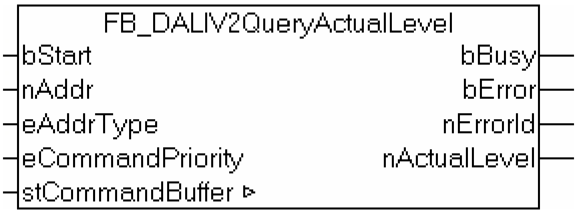
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.77 FB_DALIV2QueryActualLevel



The ACTUAL DIM LEVEL [▶ 364] variable (current lamp power) is read from the ballast.

VAR_INPUT

```
bStart              : BOOL;
nAddr                : BYTE;
eAddrType            : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority     : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nActualLevel    : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

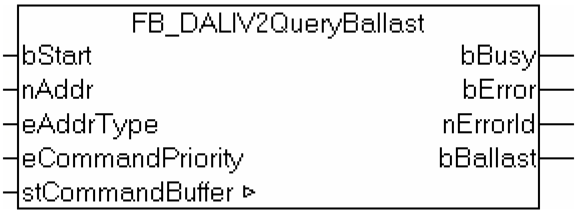
nActualLevel: Lamp power (0-254).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.78 FB_DALIV2QueryBallast



The block provides information as to whether a specific ballast is ready for operation.

Using this command, it can easily be determined whether or not any ballasts at all are connected to a DALI line. To do this, the block with the parameter *eAddrType* = *eDALIV2AddrTypeBroadcast* is called. If the output *bBallast* is FALSE and output *nError* is 0, there is no ballast connected to the DALI line. If the output *nError* is 0 and the output *bBallast* is TRUE, there is exactly one ballast connected to the DALI line. If several ballasts are connected, *nError* will return 5 (several ballasts have replied). In this case it is irrelevant whether or not the ballasts have short addresses.

VAR_INPUT

```
bStart            : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bBallast       : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

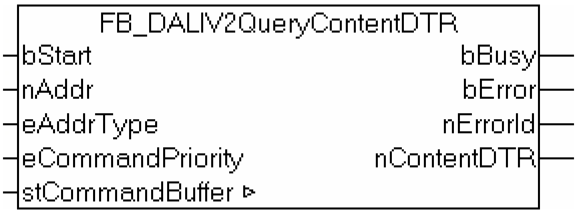
bBallast: If the output is active, the corresponding ballast is ready for operation.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.79 FB_DALIV2QueryContentDTR



The contents of the DTR (Data Transfer Register) is read from the ballast.

VAR_INPUT

```
bStart                : BOOL;
nAddr                  : BYTE;
eAddrType              : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority       : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
nContentDTR      : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

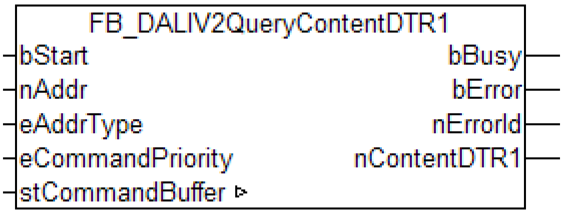
nContentDTR : Contents of the DTR (Data Transfer Register).

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.80 FB_DALIV2QueryContentDTR1



The contents of the DTR1 (Data Transfer Register 1) is read from the control gear.

> This command can only be executed by DALI devices that comply with the IEC 62386 standard.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nContentDTR1   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. It is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

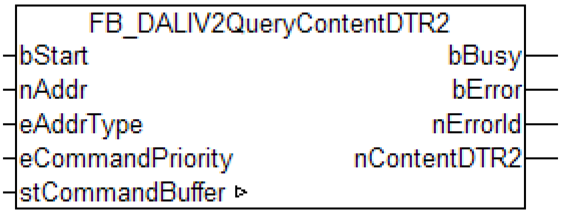
nContentDTR1: Contents of DTR1 (Data Transfer Register 1).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.81 FB_DALIV2QueryContentDTR2



The contents of the DTR2 (Data Transfer Register 2) is read from the control gear.

This command can only be executed by DALI devices that comply with the IEC 62386 standard.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nContentDTR2   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. It is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

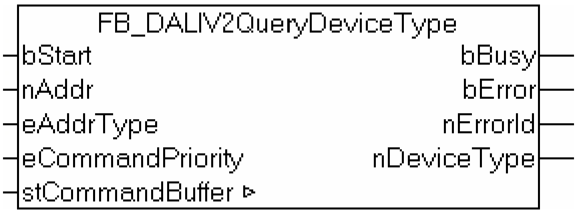
nContentDTR2: Contents of DTR2 (Data Transfer Register 2).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.82 FB_DALIV2QueryDeviceType



The device type (DEVICE TYPE [▶ 366]) is read from the ballast. The following device types are defined according to the IEC 62386 standard:

Value	Description
0	Standard device
1	Device for emergency lighting [▶ 32].
2	Device for discharge lamps [▶ 36].
3	Device for low-voltage halogen lamps.
4	Device for dimming bulbs.
5	Device for converting digital signals into DC signals.
6	Device for light emitting diodes (LEDs) [▶ 36].
7	Switching function.
8	Device for colour/colour temperature control [▶ 34].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
nDeviceType       : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

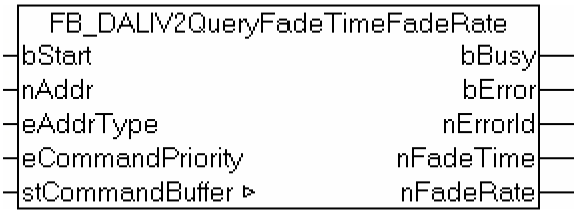
nDeviceType: Identifier for the device type (see table above).

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.83 FB_DALIV2QueryFadeTimeFadeRate



The FADE TIME [▸ 365] and FADE RATE [▸ 364] variables are read from the ballast.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address, group address or broadcast.

eCommandPriority: The priority (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nFadeTime       : BYTE;
nFadeRate       : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

nFadeTime: Fade time (0 to 15).

nFadeRate: Fade rate (1 to 15).

VAR_IN_OUT

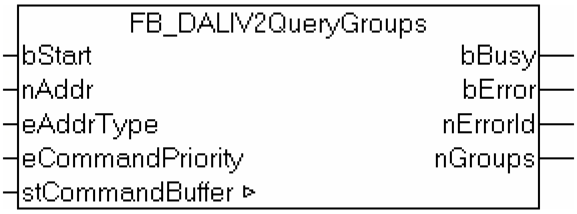
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Also see about this

- E_DALIV2AddrType [▸ 380]
- E_DALIV2CommandPriority [▸ 381]

5.1.84 FB_DALIV2QueryGroups



The GROUP 0-7 and GROUP 8-15 [▶ 366] variables are read from the ballast and combined into a 16 bit value. Each bit represents one group. Bit 0 represents group 0, while bit 15 represents group 15. If the bit is set, the ballast belongs to the corresponding group.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nGroups         : WORD;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

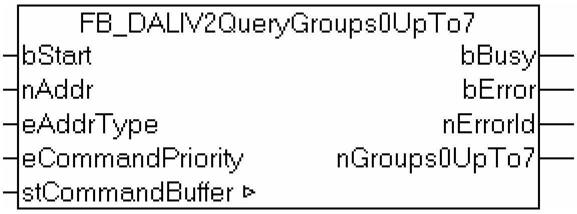
nGroups: Association with a group.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.85 FB_DALIV2QueryGroups0UpTo7



The GROUP 0-7 [▶ 366] variable is read from the ballast and linked to an 8 bit value. Each bit represents one group. Bit 0 represents group 0, while bit 7 represents group 7. If the bit is set, the ballast belongs to the corresponding group.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nGroups0UpTo7 : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

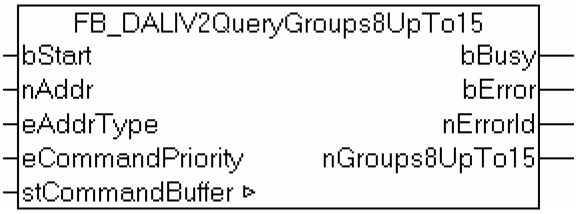
nGroups0UpTo7: Association with a group.

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.86 FB_DALIV2QueryGroups8UpTo15



The GROUP 8-15 [▶ 366] variable is read from the ballast and linked to an 8 bit value. Each bit represents one group. Bit 8 represents group 0, while bit 7 represents group 15. If the bit is set, the ballast belongs to the corresponding group.

VAR_INPUT

```
bStart            : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nGroups8UpTo15  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

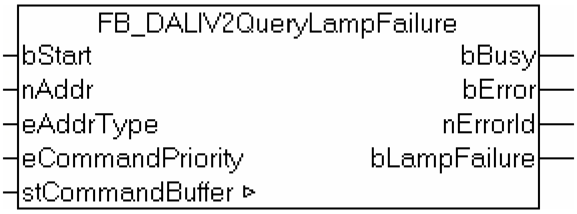
nGroups8UpTo15: Association with a group.

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.87 FB_DALIV2QueryLampFailure



The block provides information as to whether a specific ballast has a lamp problem.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bLampFailure   : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

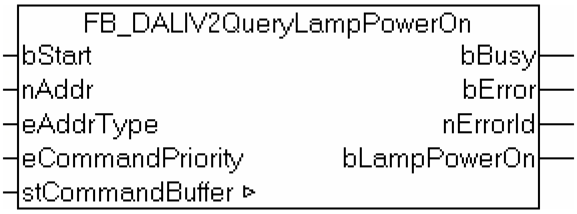
bLampFailure: If the output is active there has been a lamp failure at the corresponding ballast.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.88 FB_DALIV2QueryLampPowerOn



The block returns the information as to whether the lamp associated with a specific ballast is switched on.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
bLampPowerOn    : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

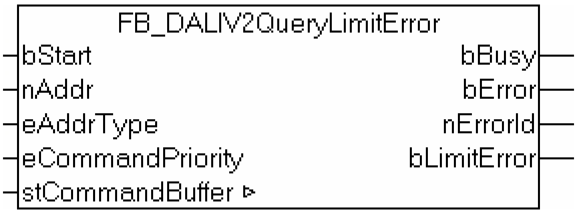
bLampPowerOn: If the output is active the lamp at the corresponding ballast is switched on.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.89 FB_DALIV2QueryLimitError



The block indicates whether the most recent lamp power value at a specific ballast cannot be used on the grounds that it is either above MAX LEVEL or is below MIN LEVEL [▶ 364].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bLimitError    : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

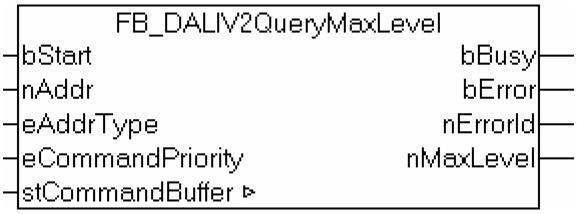
bLimitError: If the output is active, the most recent lamp power value cannot be used.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.90 FB_DALIV2QueryMaxLevel



The MAX LEVEL [▶ 364] variable (maximum permitted lamp power) is read from the ballast. This value specifies the upper limit for lamp power commands.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nMaxLevel      : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

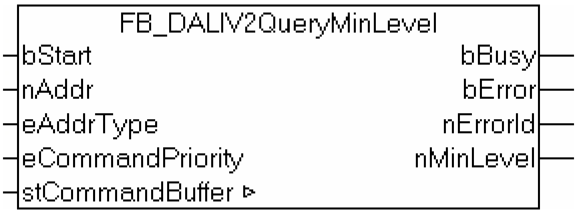
nMaxLevel: maximum permitted lamp power (0 - 254).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.91 FB_DALIV2QueryMinLevel



The MIN LEVEL [▸ 364] variable (minimum permitted lamp power) is read from the ballast. This value specifies the lower limit for lamp power commands.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nMinLevel      : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

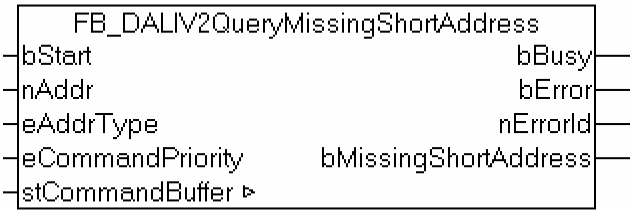
nMinLevel: minimum permitted lamp power (0 - 254).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.92 FB_DALIV2QueryMissingShortAddress



The block provides information as to whether a specific ballast has a short address or not.

Using this command, it can be determined whether or not any ballasts without a short address are connected to a DALI line. To do this, the block with the parameter *eAddrType = eDALIV2AddrTypeBroadcast* is called. If output *bMissingShortAddress* is FALSE and output *nError* is 0, all ballasts have a valid short address. If the output *nError* is 0 and the output *bMissingShortAddress* is TRUE, there is exactly one ballast that has no short address. If several ballasts have no short address, *nError* will return 5 (several ballasts have replied).

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                   : BOOL;
bError                  : BOOL;
nErrorId                : UDINT;
bMissingShortAddress    : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

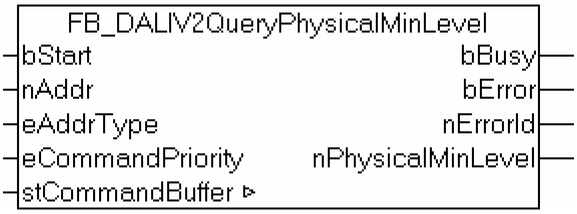
bMissingShortAddress: If the output is active the corresponding ballast does not have a short address.

VAR_IN_OUT

```
stCommandBuffer         : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.93 FB_DALIV2QueryPhysicalMinLevel



The PHYSICAL MIN LEVEL [▶ 367] variable (minimum physically possible lamp power) is read from the ballast. This value can only be read and is specified by the manufacturer.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nPhysicalMinLevel  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

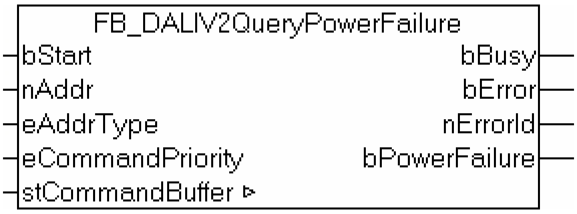
nPhysicalMinLevel: the lowest physically possible lamp power (0 - 254).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.94 FB_DALIV2QueryPowerFailure



Query whether the ballast has received a reset or a control command since it was switched on or not.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bPowerFailure  : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

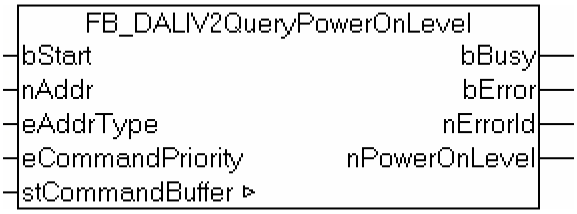
bPowerFailure: If the output is active, no lamp power control command has yet been sent to the ballast.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.95 FB_DALIV2QueryPowerOnLevel



The POWER ON LEVEL [▶ 364] variable (initial lamp power) is read from the ballast. The lamp switches itself to this power value immediately after the power is connected to the ballast.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nPowerOnLevel  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

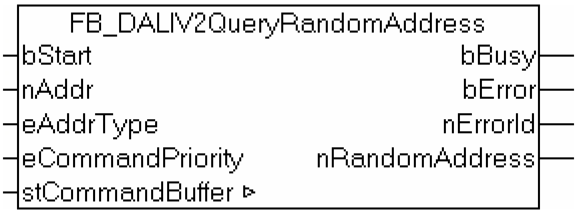
nPowerOnLevel: Lamp power at switch-on (0 - 254).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.96 FB_DALIV2QueryRandomAddress



The RANDOM ADDRESS [▸ 365] variable is read from the ballast.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nRandomAddress : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

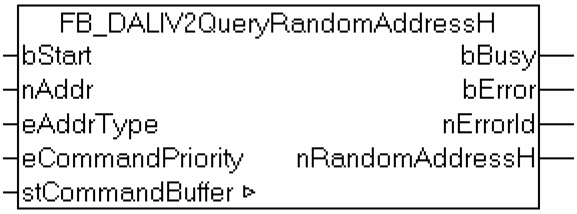
nRandomAddress: Random address/long address of the ballast.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.97 FB_DALIV2QueryRandomAddressH



The high-order byte of the RANDOM ADDRESS [▶ 365] variable is read from the ballast.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nRandomAddressH : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

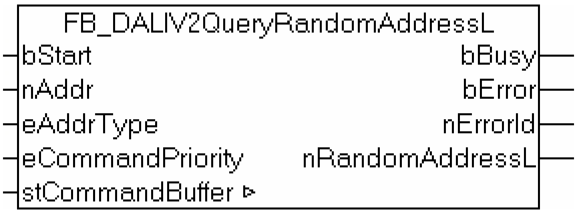
nRandomAddressH: The high-order byte of the random address/long address.

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.98 FB_DALIV2QueryRandomAddressL



The low-order byte of the RANDOM ADDRESS [▶ 365] variable is read from the ballast.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nRandomAddressL : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

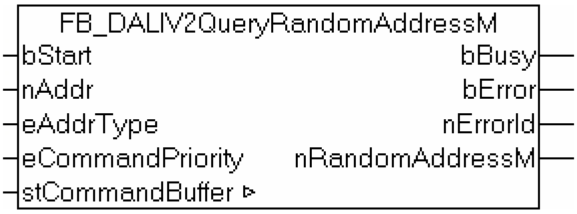
nRandomAddressL: The low-order byte of the random address/long address.

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.99 FB_DALIV2QueryRandomAddressM



The medium-order byte of the RANDOM ADDRESS [▶ 365] variable is read from the ballast.

VAR_INPUT

```
bStart            : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
nRandomAddressM   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

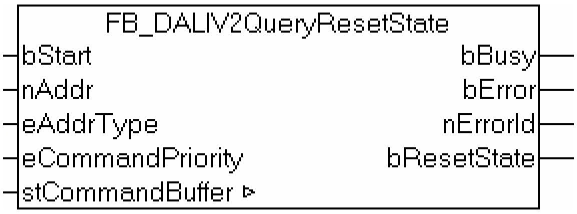
nRandomAddressM: The medium-order byte of the random address/long address.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.100 FB_DALIV2QueryResetState



The block provides information as to whether a specific ballast is in the reset state.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bResetState    : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

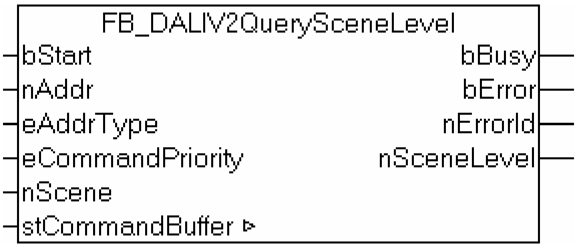
bResetState: If the output is active the corresponding ballast is in the reset state.

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.101 FB_DALIV2QuerySceneLevel



The lamp power value of the corresponding scene is read from the ballast.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nScene             : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nScene: Scene from which the lamp power value is to be read (0 - 15).

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nSceneLevel   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

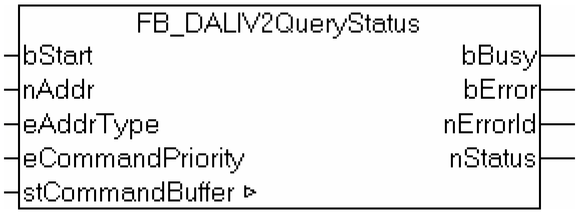
nSceneLevel: The lamp power value associated with scene.

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.102 FB_DALIV2QueryStatus



The STATUS INFORMATION [▶ 366] variable is read by the ballast. The status information contains the eight most important items describing the status of a ballast. The significance of the individual bits is defined as follows:

Bit	Description
0	Status of the ballast. 0: OK.
1	Lamp failure. 0: OK.
2	Lamp power on. 0: OFF.
3	Limit value error. 0: the most recently requested lamp power was either between MIN LEVEL and MAX LEVEL or was OFF.
4	Fading completed: 0: fading finished. 1: fading active.
5	Reset status. 0: No.
6	Missing short address. 0: No.
7	Power supply fault. 0: No, A reset or a lamp power control command has been received since the most recent power up.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nStatus        : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

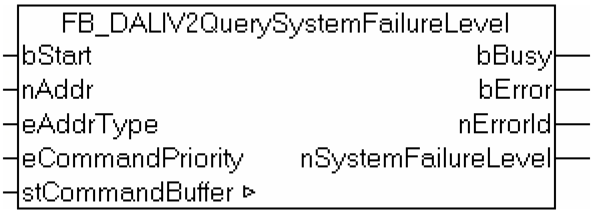
nStatus: Status information (see table above).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.103 FB_DALIV2QuerySystemFailureLevel



The SYSTEM FAILURE LEVEL [▶ 364] variable is read from the ballast. If a fault (such as the absence of the supply voltage) is detected on the DALI bus, the ballast switches the lamp to this power value.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                : BOOL;
bError               : BOOL;
nErrorId             : UDINT;
nSystemFailureLevel  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

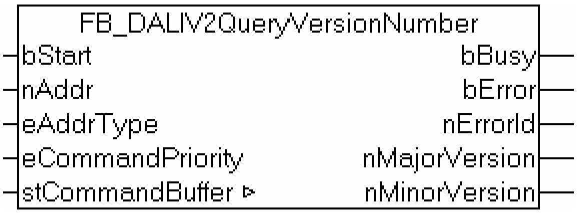
nSystemFailureLevel: The value of lamp power to be adopted in the event of a system fault.

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.104 FB_DALIV2QueryVersionNumber



The VERSION NUMBER [▶ 366] variable is read from the ballast. The version number corresponds to the version number of the IEC standard in accordance with which the software and hardware of the ballast has been developed and manufactured. The version number can only be read, and is specified by the manufacturer. The major version (*nMajorVersion*) and the minor version (*nMinorVersion*) can each have a value in the range from 0 to 15 (4 bits).

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nMajorVersion : BYTE;
nMinorVersion : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

nMajorVersion: Main version number.

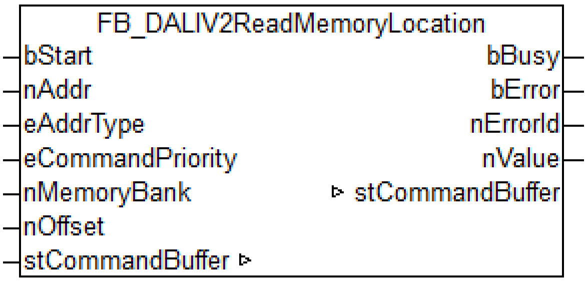
nMinorVersion: Minor version number.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.105 FB_DALIV2ReadMemoryLocation



One byte is read from the control gear memory. The exact memory bank is specified by the parameter *nMemoryBank* and the address within the memory bank by the parameter *nOffset*.

This command can only be executed by DALI devices that comply with the IEC 62386 standard.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nMemoryBank         : BYTE := 0;
nOffset             : BYTE := 0;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nMemoryBank: The memory bank to be accessed.

nOffset: The address within the memory bank to be accessed.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nValue          : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

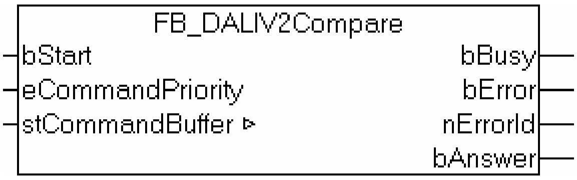
nValue: Byte read from the memory bank of the control unit.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.106 FB_DALIV2Compare



The ballast compares its RANDOM ADDRESS [▶ 365] with the SEARCH ADDRESS [▶ 365]. If the random address is smaller than or equal to the search address, and if the ballast is not connected, then the output *bAnswer* is set to TRUE.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority     : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
bAnswer         : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

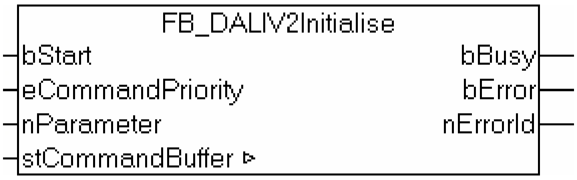
bAnswer: The random address is smaller than or equal to the search address.

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.107 FB_DALIV2Initialise



The addressing of the ballasts is started with this command. The addressing has to be halted again with the FB_DALIV2Terminate() [▶ 164] block. The maximum duration is limited to 15 minutes. Each ballast ends the addressing automatically after 15 minutes had elapsed. The reaction of the ballasts that receive this command depends on the parameter *nParameter*:

Value (binary)	Description
0000 0000	All ballasts react.

Value (binary)	Description
0AAA AAA1	Ballasts with the address AAA AAA react.
1111 1111	Ballasts with no short address react.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nParameter          : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nParameter: Specifies which ballasts should react to this command (see table above).

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

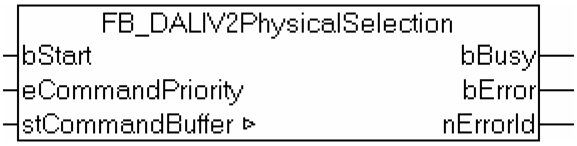
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.108 FB_DALIV2PhysicalSelection



Any ballast that receives this command enters the *physical selection mode*. In this mode, the comparison of the RANDOM ADDRESS [▶ 365] with the SEARCH ADDRESS [▶ 365] is blocked.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

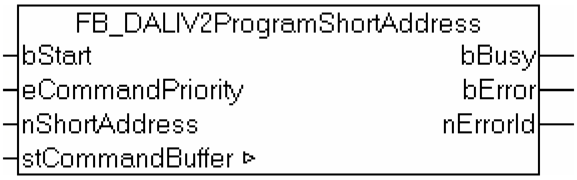
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.109 FB_DALIV2ProgramShortAddress



All the selected ballasts save the *nShortAddress* (SHORT ADDRESS [▶ 365]) value as their short address.

Selected means:

- The RANDOM ADDRESS [▶ 365] of the ballast is the same as the SEARCH ADDRESS [▶ 365].
- Physical selection is determined by the ballast, as the lamp has been disconnected from the ballast (after receiving the FB_DALIV2PhysicalSelection() [▶ 156] command).

VAR_INPUT

```
bStart                : BOOL;
eCommandPriority       : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nShortAddress          : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nShortAddress: Short address to be assigned to the selected ballasts (0 - 63).

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

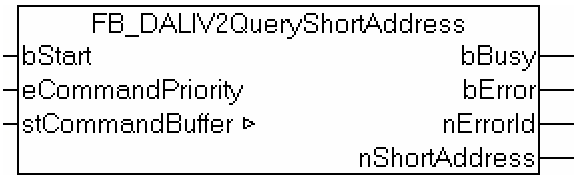
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.110 FB_DALIV2QueryShortAddress



Once the ballast has been selected, it sends its short address (SHORT ADDRESS [▸ 365]).

Selected means:

- The RANDOM ADDRESS [▸ 365] of the ballast is the same as the SEARCH ADDRESS [▸ 365].
- Physical selection is determined by the ballast, as the lamp has been disconnected from the ballast (after receiving the FB_DALIV2PhysicalSelection() [▸ 156] command).

VAR_INPUT

```
bStart            : BOOL;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nShortAddress : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

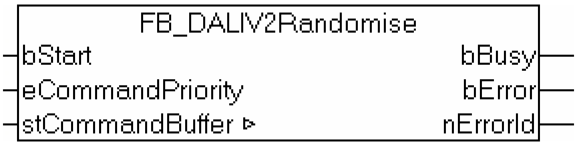
nShortAddress: Short address of the ballast (0 - 63).

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.111 FB_DALIV2Randomise



The ballasts generate a new RANDOM ADDRESS [▶ 365].

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

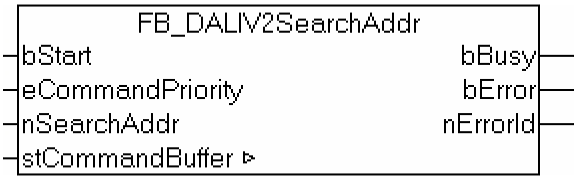
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.112 FB_DALIV2SearchAddr



This block sets a SEARCH ADDRESS [▶ 365].

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nSearchAddr        : UDINT;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nSearchAddr: Search address.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

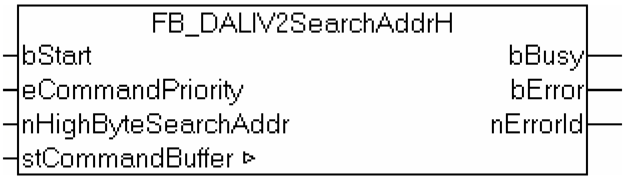
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.113 FB_DALIV2SearchAddrH



This block sets the upper 8 bits of the 24-bit SEARCH ADDRESS [▸ 365].

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nHighByteSearchAddr : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nHighByteSearchAddr: upper 8 bits of the 24-bit search address.

VAR_OUTPUT

```
bBusy               : BOOL;
bError              : BOOL;
nErrorId            : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

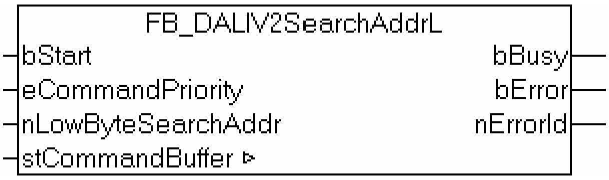
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.114 FB_DALIV2SearchAddrL



This block sets the lower 8 bits of the 24-bit SEARCH ADDRESS [▶ 365].

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nLowByteSearchAddr : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nLowByteSearchAddr: lower 8 bits of the 24-bit search address.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

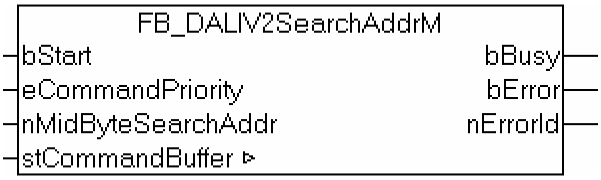
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.115 FB_DALIV2SearchAddrM



This block sets the middle 8 bits of the 24-bit SEARCH ADDRESS [▶ 365].

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nMidByteSearchAddr : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nMidByteSearchAddr: middle 8 bits of the 24-bit search address.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

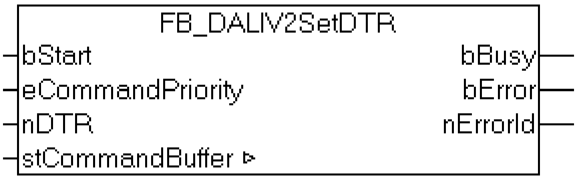
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.116 FB_DALIV2SetDTR



This command is only available as a broadcast. Data is written to the DTR of all the ballasts.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDTR                : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nDTR: The value that is to be written into the DTR.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

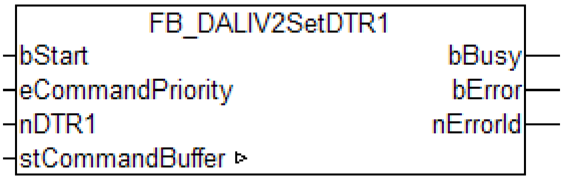
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.117 FB_DALIV2SetDTR1



This command is only available as a broadcast. Data is written to the DTR1 of all the control gears.

This command can only be executed by DALI devices that comply with the IEC 62386 standard.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDTR1              : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nDTR1: The value that is to be written into the DTR1.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

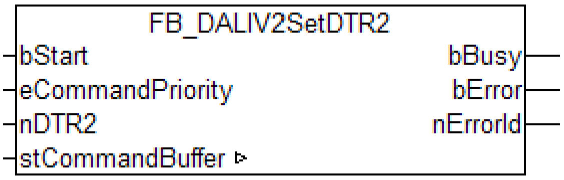
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.118 FB_DALIV2SetDTR2



This command is only available as a broadcast. Data is written to the DTR2 of all the control gears.

This command can only be executed by DALI devices that comply with the IEC 62386 standard.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDTR2               : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nDTR2: The value that is to be written into the DTR2.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

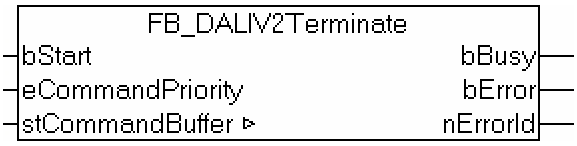
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.119 FB_DALIV2Terminate



The addressing of all the ballasts is halted.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

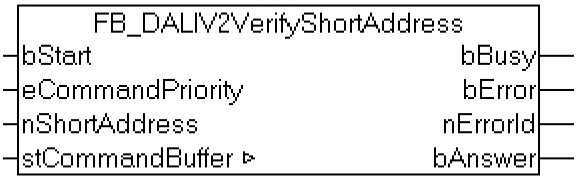
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.120 FB_DALIV2VerifyShortAddress



If the short address in the ballast is equal to the *nShortAddress* parameter, TRUE is asserted at the *bAnswer* output.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nShortAddress      : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nShortAddress: Short address with which the ballast's short address is to be compared.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bAnswer        : BOOL;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

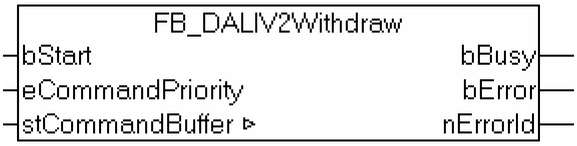
bAnswer: The *nShortAddress* parameter is the same as the ballast's own address.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.121 FB_DALIV2Withdraw



Ballasts in which the RANDOM ADDRESS [▸ 365] is the same as the SEARCH ADDRESS [▸ 365] must no longer react to the FB_DALIV2Compare() [▸ 155] command.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority     : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

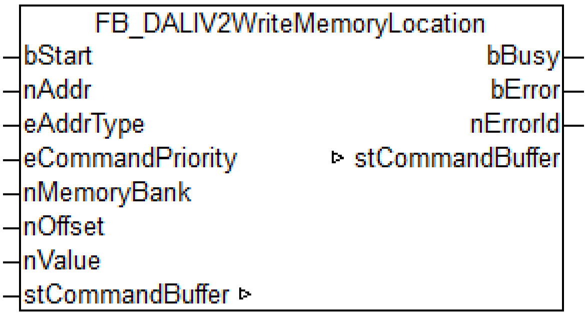
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.122 FB_DALIV2WriteMemoryLocation



The value *nValue* is written in the memory bank of the control gear. The exact memory bank is specified by *nMemoryBank* and the address within the memory bank by *nOffset*.

This command can only be executed by DALI devices that comply with the IEC 62386 standard.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nMemoryBank        : BYTE;
nOffset            : BYTE;
nValue             : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nMemoryBank: The memory bank to be accessed.

nOffset: The address within the memory bank to be accessed.

nValue: Value to be written to the memory bank of the control unit.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

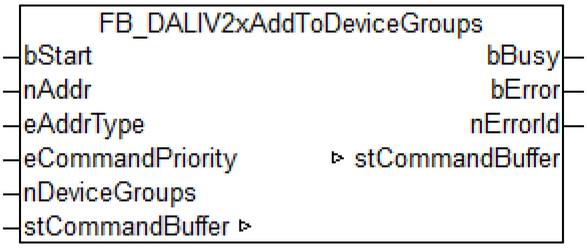
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Also see about this

- FB_DALIV2EnableWriteMemory [▸ 108]

5.1.123 FB_DALIV2xAddToDeviceGroups



Assigns the control unit to one or more groups.

A total of 32 groups are available to which a control unit can be assigned. Each bit of the variable *nDeviceGroups* corresponds to one of these groups. If the bit is set, the control unit is assigned to the respective group. Bit 0 corresponds to group 0, bit 31 to group 31.

The function block FB_DALIV2xRemoveFromDeviceGroups() [▸ 174] can be used to remove a control unit from a group.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDeviceGroups       : DWORD;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nDeviceGroups: 32 bit variable where each bit represents the corresponding group to which the control unit is to be assigned.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

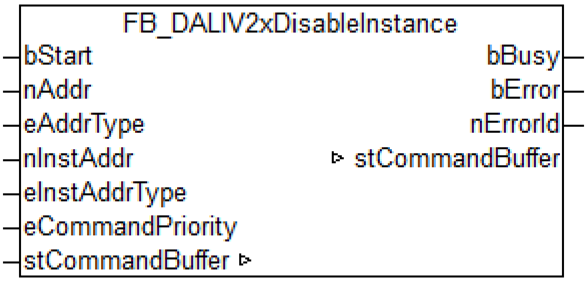
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.124 FB_DALIV2xDisableInstance



The control unit instance is disabled.

The function block FB_DALIV2xEnableInstance() [▶ 171] can be used to enable the instance.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance [▶ 382] within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

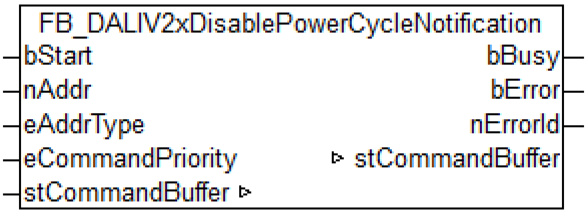
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.125 FB_DALIV2xDisablePowerCycleNotification



This function block blocks the *Power Cycle Notification* event.

The function block FB_DALIV2xEnablePowerCycleNotification() [▸ 172] can be used to enable the event.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

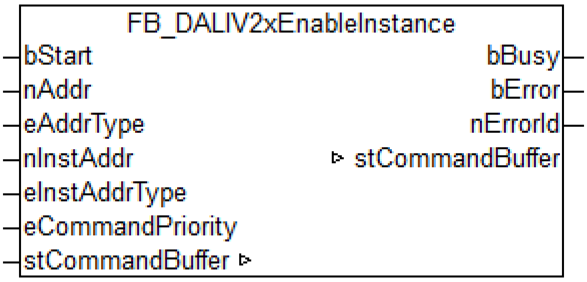
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.126 FB_DALIV2xEnableInstance



The control unit instance is enabled.

The function block FB_DALIV2xDisableInstance() [▸ 169] can be used to disable the instance.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

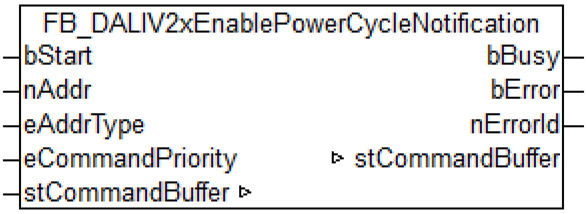
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.127 FB_DALIV2xEnablePowerCycleNotification



This function block enables the *Power Cycle Notification* event.

The function block FB_DALIV2xDisablePowerCycleNotification() [▶ 170] can be used to lock the event.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

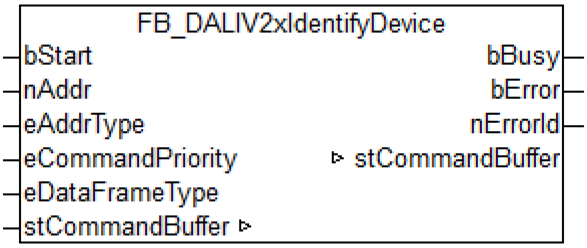
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.128 FB_DALIV2xIdentifyDevice



Starts the identification routine for the control unit.

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

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType      : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

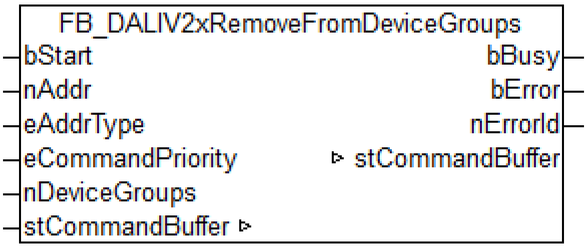
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.129 FB_DALIV2xRemoveFromDeviceGroups



Removes the control unit from one or more groups.

A total of 32 groups are available to which a control unit can be assigned. Each bit of the variable *nDeviceGroups* corresponds to one of these groups. If the bit is set, the control unit is removed from the respective group. Bit 0 corresponds to group 0, bit 31 to group 31.

The function block FB_DALIV2xAddToDeviceGroups() [▶ 168] can be used to assign a control unit to a group.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDeviceGroups       : DWORD;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nDeviceGroups: 32 bit variable where each bit represents the corresponding group from which the control unit is to be removed.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

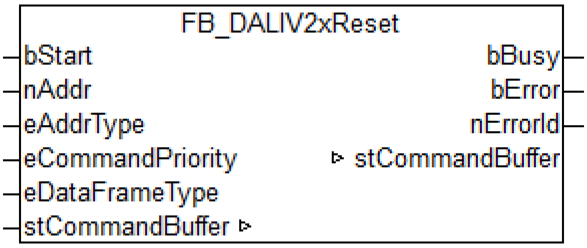
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.130 FB_DALIV2xReset



This function block resets all parameters to their default values.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType      : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

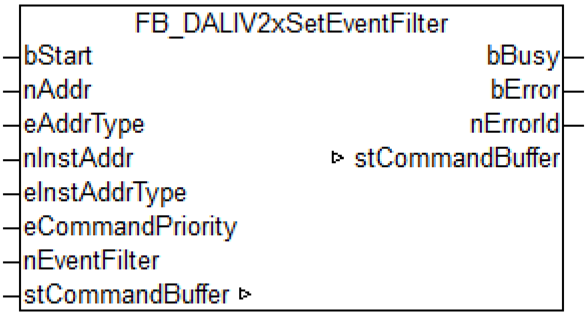
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.131 FB_DALIV2xSetEventFilter



This function block sets the event filter for the respective control unit instance.

Each bit in *nEventFilter* represents an event. If the bit is set, the associated event is enabled. The event is locked if the bit is not set.

The meaning of the individual bits can be found in the documentation of the respective control unit.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nEventFilter        : DWORD := 0;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nEventFilter: Each bit represents an event to be enabled or disabled.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

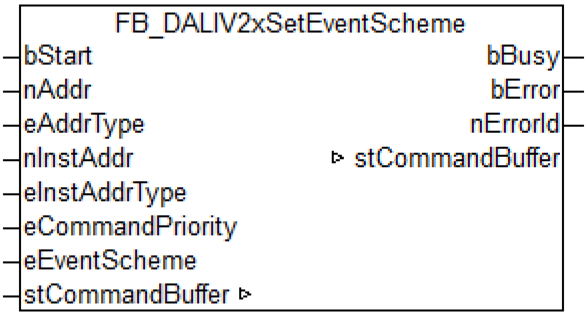
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.132 FB_DALIV2xSetEventScheme



This function block can be used to define the addressing scheme for the events of the respective control unit instance.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eEventScheme        : E_DALIV2EventScheme := eDALIV2EventSchemeInstance;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

eEventScheme: Addressing scheme [▸ 382] for the events.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

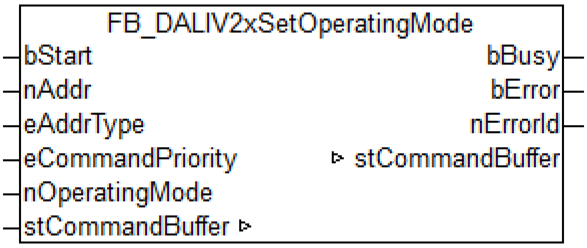
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.133 FB_DALIV2xSetOperatingMode



Sets the *Operating Mode* for the control unit.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nOperatingMode      : BYTE := 0;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nOperatingMode: New operating mode.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

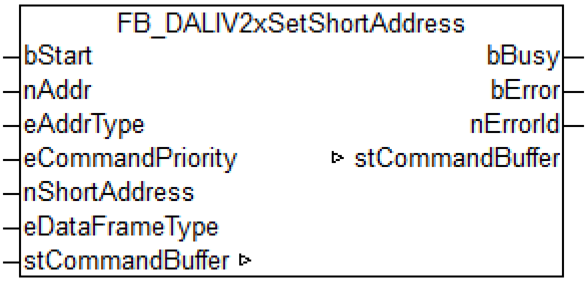
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.134 FB_DALIV2xSetShortAddress



Sets the short address of the control unit.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nShortAddress      : BYTE;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nShortAddress: New short address (0...63, 255)

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

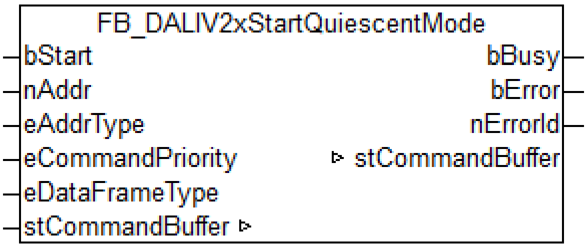
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.135 FB_DALIV2xStartQuiescentMode



The *quiescent mode* of the control unit is started.

In *Quiescent mode* the control unit does not send commands or events. *Quiescent mode* ends after approx. 15 minutes or through FB_DALIV2xStopQuiescentMode() [▶ 181].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType      : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

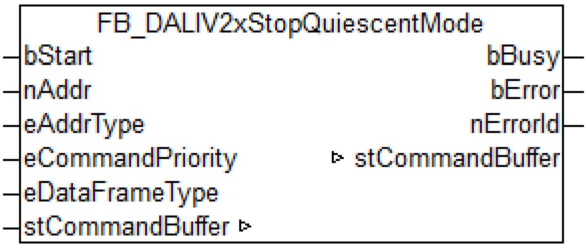
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.136 FB_DALIV2xStopQuiescentMode



The *quiescent mode* of the control unit is stopped.

In *Quiescent mode* the control unit does not send commands or events. *Quiescent mode* is started with FB_DALIV2xStartQuiescentMode() [▶ 180].

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
nErrorId     : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

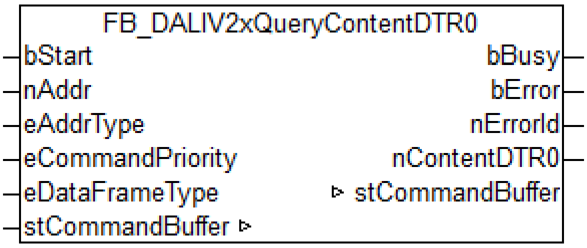
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.137 FB_DALIV2xQueryContentDTR0



The content of DTR0 (Data Transfer Register) is read from the control unit.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nContentDTR0   : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nContentDTR0: Contents of the DTR0 (Data Transfer Register)

VAR_IN_OUT

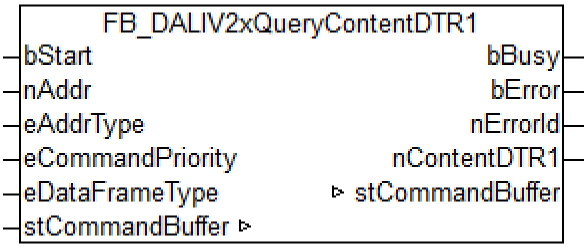
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.138 FB_DALIV2xQueryContentDTR1



The content of DTR1 (Data Transfer Register) is read from the control unit.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▸ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nContentDTR1  : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nContentDTR1: Contents of the DTR1 (Data Transfer Register)

VAR_IN_OUT

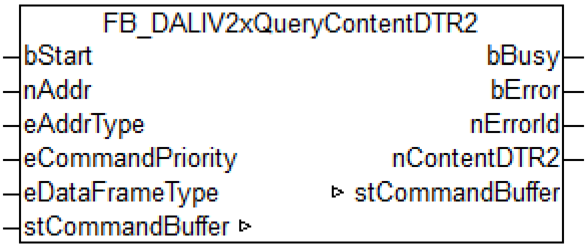
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.139 FB_DALIV2xQueryContentDTR2



The content of DTR2 (Data Transfer Register) is read from the control unit.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType    : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nContentDTR2  : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nContentDTR2: Contents of the DTR2 (Data Transfer Register)

VAR_IN_OUT

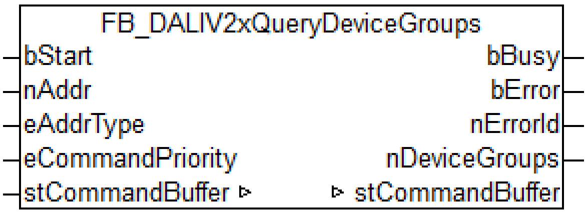
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.140 FB_DALIV2xQueryDeviceGroups



Queries the group allocations of the control unit.

A total of 32 groups are available to which a control unit can be assigned. Each bit of the variable *nDeviceGroups* corresponds to one of these groups. If the bit is set, the control unit is assigned to the respective group. Bit 0 corresponds to group 0, bit 31 to group 31.

The function block FB_DALIV2xAddToDeviceGroups() [▶ 168] can be used to assign a control unit to a group.

The function block FB_DALIV2xRemoveFromDeviceGroups() [▶ 174] can be used to remove a control unit from a group.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nDeviceGroups : DWORD;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nDeviceGroups: 32-bit variable where each bit represents the corresponding group to which the control unit has been assigned.

VAR_IN_OUT

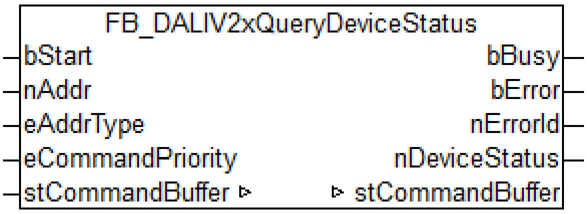
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.141 FB_DALIV2xQueryDeviceStatus



The function block reads the *device status* of the control unit.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nDeviceStatus : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nDeviceStatus: The *device status* of the control unit.

VAR_IN_OUT

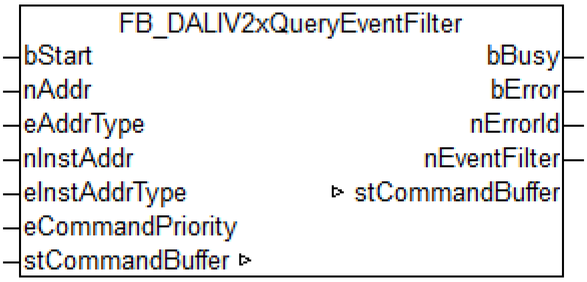
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.142 FB_DALIV2xQueryEventFilter



This function block queries the event filter for the respective control unit instance.

Each bit in *nEventFilter* represents an event. If the bit is set, the associated event is enabled. The event is locked if the bit is not set.

The meaning of the individual bits can be found in the documentation of the respective control unit.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nEventFilter  : DWORD;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nEventFilter: Each bit represents an event that has been enabled or disabled.

VAR_IN_OUT

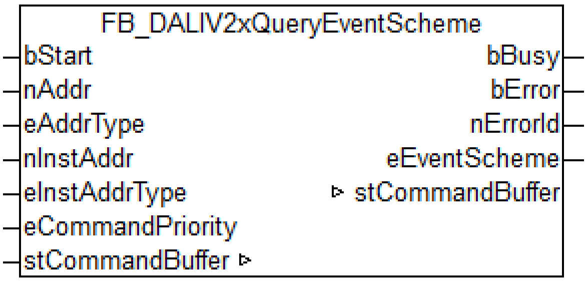
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.143 FB_DALIV2xQueryEventScheme



This function block can be used to query the addressing scheme for the events of the respective control unit instance.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
eEventScheme   : E_DALIV2EventScheme;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

eEventScheme: Addressing scheme [▶ 382] for the events.

VAR_IN_OUT

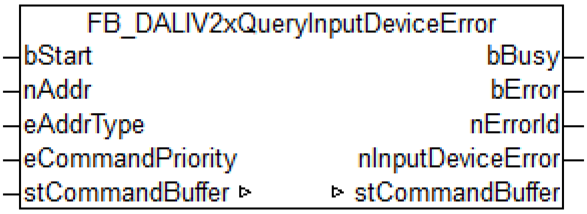
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.144 FB_DALIV2xQueryInputDeviceError



The function block reads the *input device error* of the control unit.

The meaning of the *input device error* depends on the manufacturer of the control unit.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nInputDeviceError  : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nInputDeviceError: The *Input Device Error* of the control unit.

VAR_IN_OUT

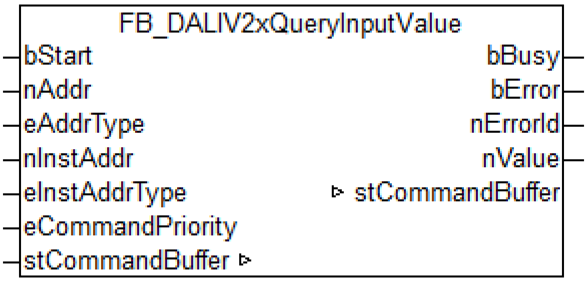
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.145 FB_DALIV2xQueryInputValue



Queries the first byte of the input value of the control unit instance.

The current input value is stored in memory, and the most significant byte (MSB) is returned.

All other bytes can be read with the function block FB_DALIV2xQueryInputValueLatch() [▶ 191].

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nValue        : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nValue: The most significant byte (MSB) of the input value.

VAR_IN_OUT

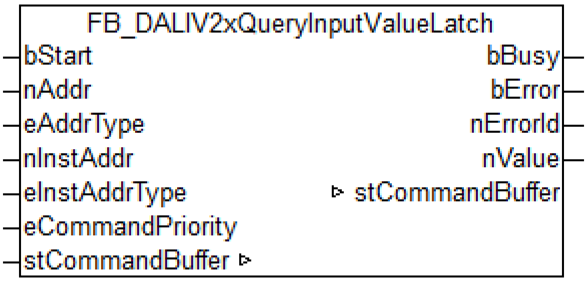
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.146 FB_DALIV2xQueryInputValueLatch



Queries the following byte of the input value of the control unit instance.

The first byte is read with the function block FB_DALIV2xQueryInputValue() [▶ 190].

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nValue         : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nValue: The following byte of the input value.

VAR_IN_OUT

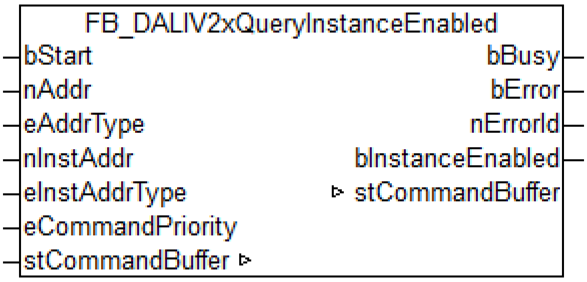
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.147 FB_DALIV2xQueryInstanceEnabled



Queries whether the control unit instance is enabled.

The function block FB_DALIV2xDisableInstance() [▶ 169] can be used to disable the instance.

The function block FB_DALIV2xEnableInstance() [▶ 171] can be used to enable the instance.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
bInstanceEnabled   : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bInstanceEnabled: Is TRUE if the instance has been enabled.

VAR_IN_OUT

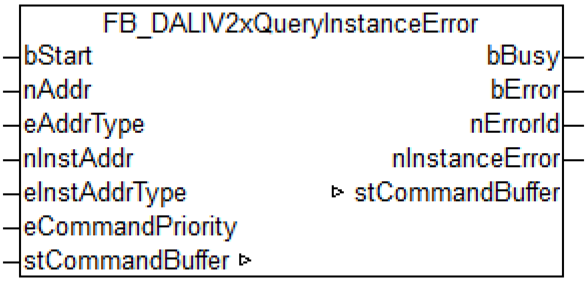
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.148 FB_DALIV2xQueryInstanceError



The function block reads the *Instance Error* of the control unit.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
nInstanceError   : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nInstanceError: The *Instance Error* of the control unit.

VAR_IN_OUT

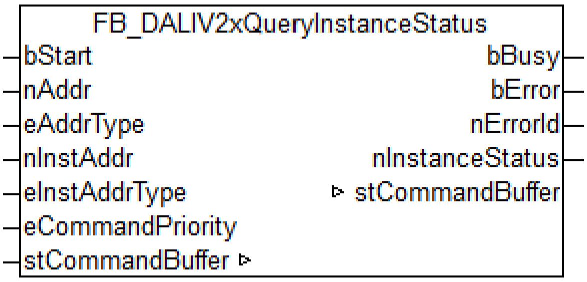
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.149 FB_DALIV2xQueryInstanceStatus



The function block reads the *Instance Status* of the control unit.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
nInstanceStatus  : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nInstanceStatus: The *Instance Status* of the control unit.

VAR_IN_OUT

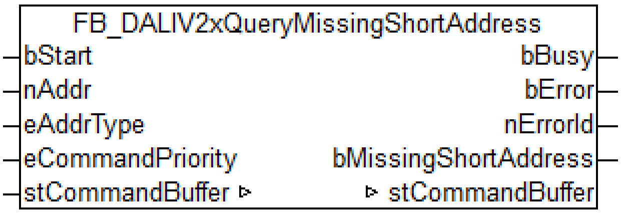
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.150 FB_DALIV2xQueryMissingShortAddress



Queries whether the control unit does not have a valid short address.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy                   : BOOL;
bError                  : BOOL;
nErrorId                : UDINT;
bMissingShortAddress    : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bMissingShortAddress: Is TRUE if the control unit does not have a valid short address.

VAR_IN_OUT

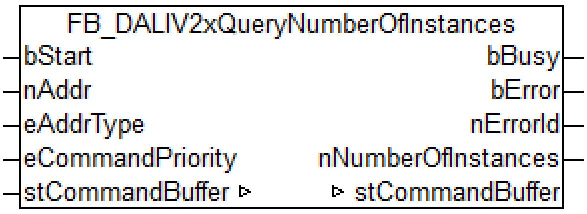
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.151 FB_DALIV2xQueryNumberOfInstances



Queries the number of instances that the control unit has.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy               : BOOL;
bError              : BOOL;
nErrorId            : UDINT;
nNumberOfInstances   : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nNumberOfInstances: Number of instances of the control unit.

VAR_IN_OUT

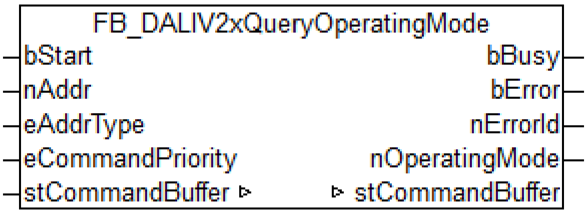
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.152 FB_DALIV2xQueryOperatingMode



Queries the *Operating Mode* of the control unit.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nOperatingMode  : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nOperatingMode: Operating Mode.

VAR_IN_OUT

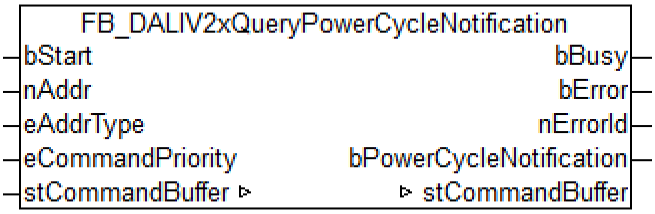
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.153 FB_DALIV2xQueryPowerCycleNotification



Queries whether the *Power Cycle Notification* event is enabled.

The function block FB_DALIV2xDisablePowerCycleNotification() [▸ 170] can be used to lock the event.

The function block FB_DALIV2xEnablePowerCycleNotification() [▸ 172] can be used to enable the event.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy                      : BOOL;
bError                     : BOOL;
nErrorId                   : UDINT;
bPowerCycleNotification    : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bPowerCycleNotification: Is TRUE if the *Power Cycle Notification* event is enabled.

VAR_IN_OUT

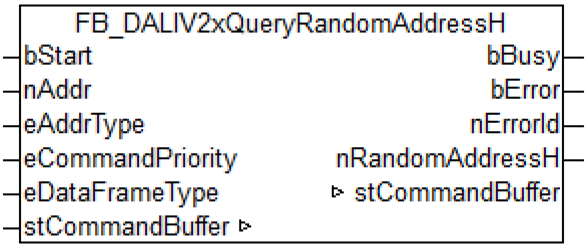
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.154 FB_DALIV2xQueryRandomAddressH



The higher-order byte of the random address is read from the control unit.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType      : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
nRandomAddressH   : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nRandomAddressH: The high-order byte of the random address/long address.

VAR_IN_OUT

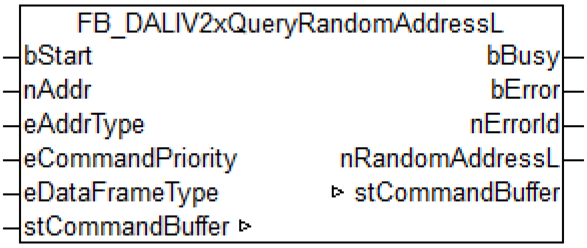
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.155 FB_DALIV2xQueryRandomAddressL



The lower-order byte of the random address is read from the control unit.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nRandomAddressL    : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nRandomAddressL: The low-order byte of the random address/long address.

VAR_IN_OUT

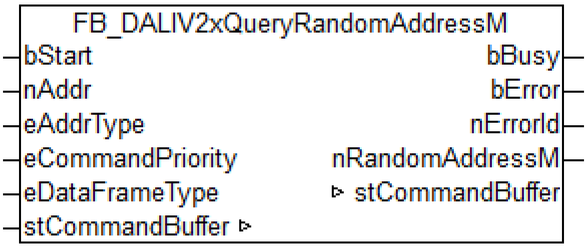
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.156 FB_DALIV2xQueryRandomAddressM



The mean byte of the random address is read from the control unit.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType      : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▸ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nRandomAddressM    : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nRandomAddressM: The medium-order byte of the random address/long address.

VAR_IN_OUT

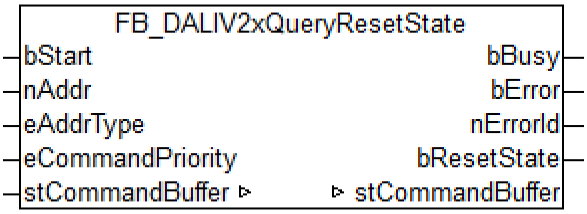
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.157 FB_DALIV2xQueryResetState



Queries whether the parameters of the control unit have their default values.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bResetState    : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

bResetState: Is TRUE if the control unit parameters have the default values.

VAR_IN_OUT

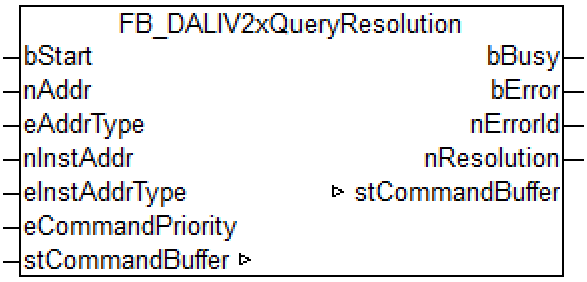
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.158 FB_DALIV2xQueryResolution



Queries the resolution of the input values of the control unit.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nResolution        : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nResolution: Resolution of the input values.

VAR_IN_OUT

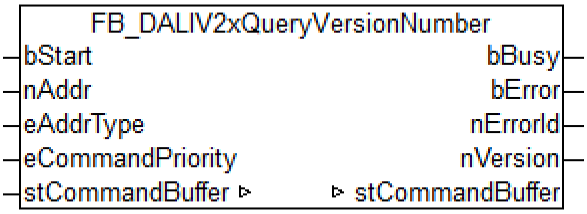
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.159 FB_DALIV2xQueryVersionNumber



Queries the version number of the control unit.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nVersion       : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nVersion: The version number of the control unit.

VAR_IN_OUT

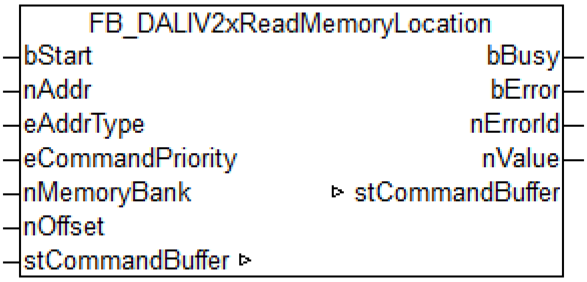
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.160 FB_DALIV2xReadMemoryLocation



A byte is read from the memory of the control unit. The exact memory bank is specified by the parameter *nMemoryBank* and the address within the memory bank by the parameter *nOffset*.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nMemoryBank         : BYTE := 0;
nOffset             : BYTE := 0;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nMemoryBank: The memory bank to be accessed.

nOffset: The address within the memory bank to be accessed.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nValue         : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nValue: Byte read from the memory bank of the control unit.

VAR_IN_OUT

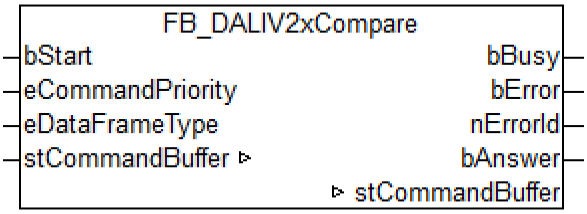
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.161 FB_DALIV2xCompare



The ballast compares its random address with the search address. If the random address is less than or equal to the search address and the control unit is not excluded, then the output *bAnswer* is set to TRUE.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType     : E_DALIV2DataFrameType  := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bAnswer        : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bAnswer: The random address is smaller than or equal to the search address.

VAR_IN_OUT

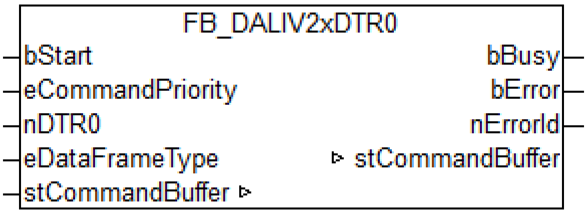
```
stCommandBuffer    : ST_DALIV2CommandBuffer
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.162 FB_DALIV2xDTR0



This command is only available as a broadcast. The DTR0 of all control units is described.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDTR0               : BYTE;
eDataFrameType      : E_DALIV2DataFrameType  := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nDTR0: The value that is to be written into the DTR0.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

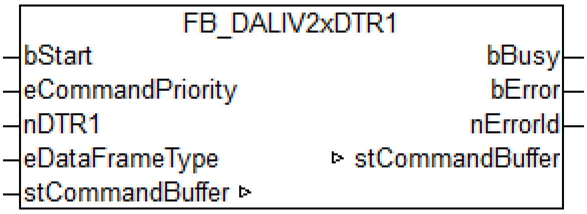
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.163 FB_DALIV2xDTR1



This command is only available as a broadcast. The DTR1 of all control units is described.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDTR1              : BYTE;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nDTR1: The value that is to be written into the DTR1.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

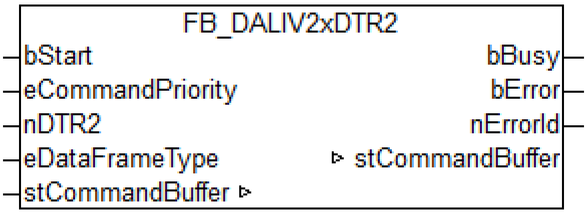
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.164 FB_DALIV2xDTR2



This command is only available as a broadcast. The DTR2 of all control units is described.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDTR2              : BYTE;
eDataFrameType     : E_DALIV2DataFrameType  := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nDTR2: The value that is to be written into the DTR2.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

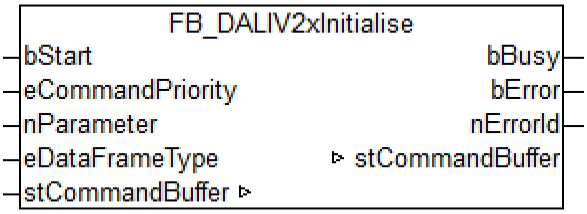
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.165 FB_DALIV2xInitialise



This command starts the addressing procedure for the control units. The addressing has to be halted again with the FB_DALIV2xTerminate() [▸ 217] function block. The maximum duration is limited to 15 minutes. Each control unit stops the addressing independently after 15 minutes. The response of the control units receiving this command depends on the parameter *nParameter*:

eDataFrameType = eDataFrameTypeOsram:

Value (binary)	Description
0000 0000	All control units respond.
0AAA AAA1	Control units with the address AAA AAA respond.
1111 1111	Control units without a short address respond.

eDataFrameType = eDataFrameType24Bit:

Value (binary)	Description
0111 1111	Control units without a short address respond.
00AA AAAA	Control units with the address AAA AAA respond.
1111 1111	All control units respond.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nParameter         : BYTE;
eDataFrameType     : E_DALIV2DataFrameType  := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nParameter: Specifies which control units are to respond to this command (see table above).

eDataFrameType: Output format [▸ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

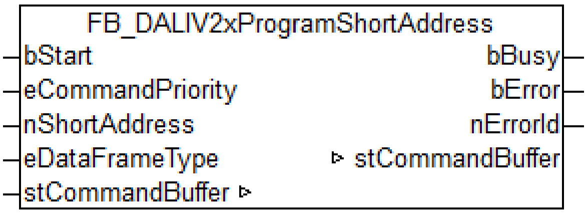
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.166 FB_DALIV2xProgramShortAddress



All selected control units save the value *nShortAddress* as a short address.

Selected means:

- The random address of the control unit matches the search address

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nShortAddress      : BYTE;
eDataFrameType     : E_DALIV2DataFrameType  := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nShortAddress: Short address to be assigned to the selected control units (0 - 63).

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

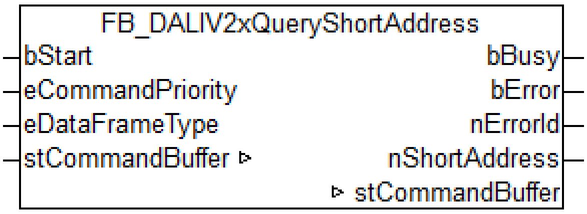
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.167 FB_DALIV2xQueryShortAddress



If the control unit is selected, it sends its short address.

Selected means:

- The random address of the control unit matches the search address

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType     : E_DALIV2DataFrameType  := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nShortAddress : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nShortAddress: Short address of the control unit (0 - 63).

VAR_IN_OUT

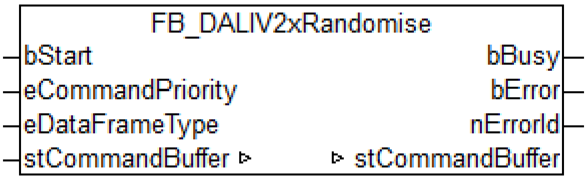
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.168 FB_DALIV2xRandomise



The control units generate a new random address.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

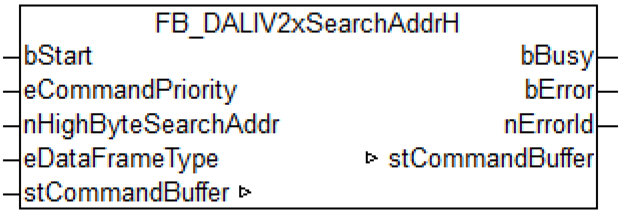
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.169 FB_DALIV2xSearchAddrH



This function block sets the upper 8 bits of the 24-bit search address.

VAR_INPUT

```
bStart               : BOOL;
eCommandPriority     : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nHighByteSearchAddr  : BYTE;
eDataFrameType       : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nHighByteSearchAddr: upper 8 bits of the 24-bit search address.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

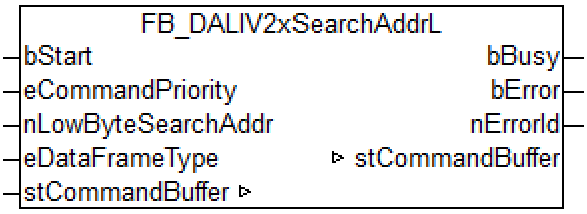
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.170 FB_DALIV2xSearchAddrL



This function block sets the lower 8 bits of the 24-bit search address.

VAR_INPUT

```
bStart               : BOOL;
eCommandPriority     : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nLowByteSearchAddr   : BYTE;
eDataFrameType       : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nLowByteSearchAddr: lower 8 bits of the 24-bit search address.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

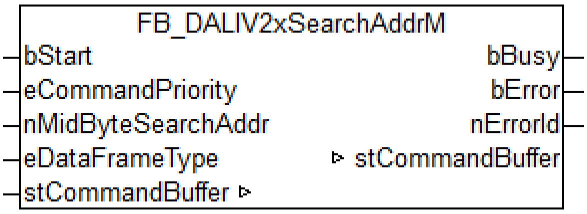
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.171 FB_DALIV2xSearchAddrM



This function block sets the middle 8 bits of the 24-bit search address.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nMidByteSearchAddr  : BYTE;
eDataFrameType      : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nMidByteSearchAddr: middle 8 bits of the 24-bit search address.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

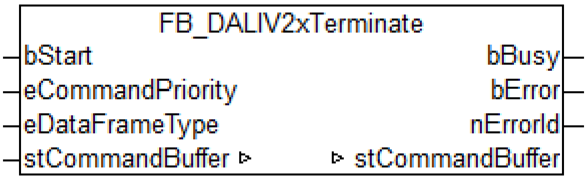
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.172 FB_DALIV2xTerminate



Addressing is terminated for all control units.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

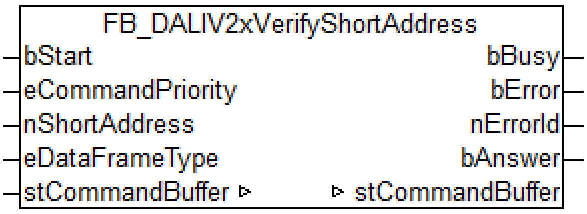
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.173 FB_DALIV2xVerifyShortAddress



If the short address of the control unit is equal to the parameter *nShortAddress*, the output *bAnswer* is set to TRUE.

VAR_INPUT

```
bStart             : BOOL;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nShortAddress      : BYTE;
eDataFrameType     : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nShortAddress: Short address with which the ballast's short address is to be compared.

eDataFrameType: Output format [▸ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bAnswer        : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

bAnswer: The *nShortAddress* parameter is the same as its own short address.

VAR_IN_OUT

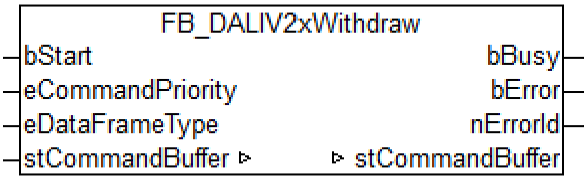
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.174 FB_DALIV2xWithdraw



Control units whose random address is the same as the search address may no longer respond to the FB_DALIV2xCompare() [▸ 206] command.

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDataFrameType      : E_DALIV2DataFrameType := eDALIV2DataFrameType24Bit;
```

bStart: The function block is activated by a positive edge at this input.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

eDataFrameType: Output format [▶ 381] of the DALI command (*eDALIV2DataFrameType24Bit* or *eDALIV2DataFrameTypeOsram*).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

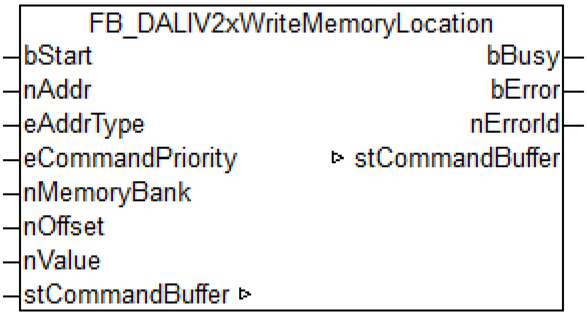
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.175 FB_DALIV2xWriteMemoryLocation



The value *nValue* is written to the memory bank of the control unit. The exact memory bank is specified by *nMemoryBank* and the address within the memory bank by *nOffset*.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

```
nMemoryBank        : BYTE;
nOffset        : BYTE;
nValue         : BYTE;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nMemoryBank: The memory bank to be accessed.

nOffset: The address within the memory bank to be accessed.

nValue: Value to be written to the memory bank of the control unit.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

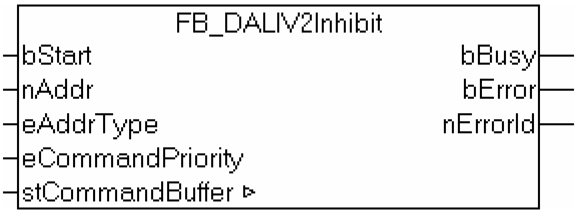
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.176 FB_DALIV2Inhibit



Prevents the control gear from switching to emergency mode for 15 minutes. The function block FB_DALIReLightResetInhibit() [▶ 235] can be used to disable inhibition of emergency mode.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart               : BOOL;
nAddr                : BYTE;
eAddrType            : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority     : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

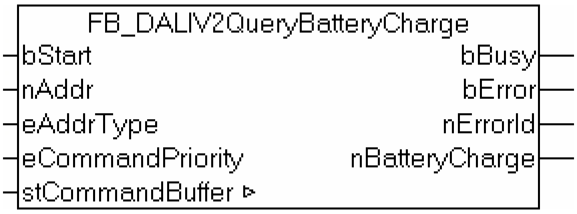
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.177 FB_DALIV2QueryBatteryCharge



The variable BATTERY CHARGE [▸ 370] (state of battery charge) is read from the control gear. 255 is returned if the control gear is unable to determine the value.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
nBatteryCharge   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

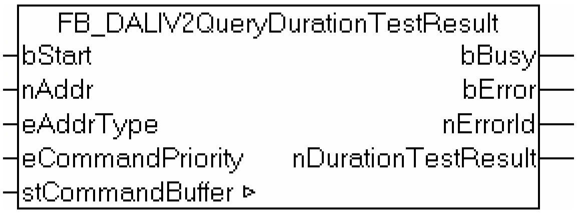
nBatteryCharge: Battery charge status. 0: empty / 254: full.

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.178 FB_DALIV2QueryDurationTestResult



The DURATION TEST RESULT [▸ 370] variable (result of duration test) is read from the control gear. The unit is 2 minutes per step. 255 means a maximum value of 8.5 hours or more. The value is only valid if bit 2 is set in the variable EMERGENCY STATUS [▸ 371].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart           : BOOL;
nAddr            : BYTE;
eAddrType        : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                : BOOL;
bError               : BOOL;
nErrorId             : UDINT;
nDurationTestResult  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

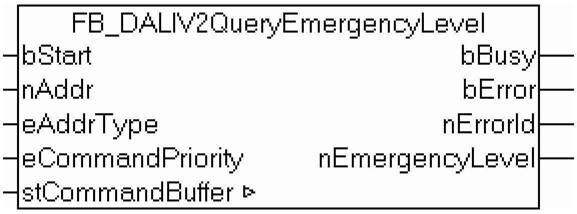
nDurationTestResult: The result of the duration test in steps of 2 minutes.

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.179 FB_DALIV2QueryEmergencyLevel



The variable EMERGENCY LEVEL [▸ 368] (emergency illuminance) is read from the control gear. 255 is returned if the control gear is unable to determine the value.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
nEmergencyLevel  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

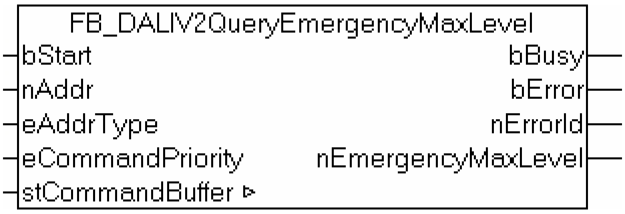
nEmergencyLevel: Emergency illuminance of the ballast.

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.180 FB_DALIV2QueryEmergencyMaxLevel



The variable EMERGENCY MAX LEVEL [▸ 368] (maximum emergency illuminance) is read from the control gear. 255 is returned if the control gear is unable to determine the value.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nEmergencyMaxLevel : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

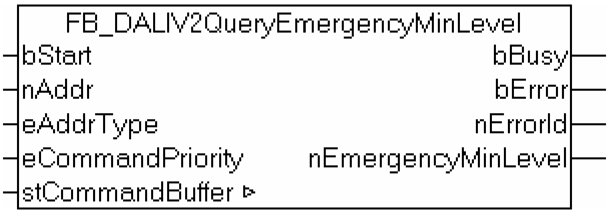
nEmergencyMaxLevel: Emergency illuminance of the ballast.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.181 FB_DALIV2QueryEmergencyMinLevel



The variable EMERGENCY MIN LEVEL [▸ 368] (minimum emergency illuminance) is read from the control gear. 255 is returned if the control gear is unable to determine the value.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nEmergencyMinLevel : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

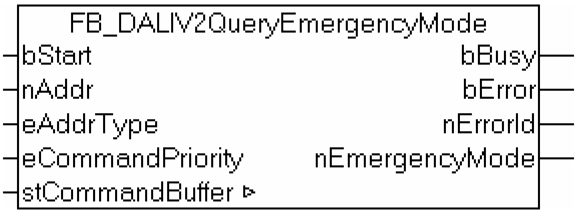
nEmergencyMinLevel: Minimum emergency illuminance of the ballast.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.182 FB_DALIV2QueryEmergencyMode



The EMERGENCY MODE [▶ 370] variable is read from the control gear.

Bit	Description
0	Reset mode. 0: no.
1	Emergency mode readiness (normal operation). 0: no.
2	Emergency mode. 0: no.
3	Extended emergency mode once mains voltage is available again. 0: no.
4	Function test active. 0: no.
5	Duration test active. 0: no.
6	Connected suppress push button is active. 0: not active or not available.
7	Connected mains voltage is active. 0: OFF.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nEmergencyMode      : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

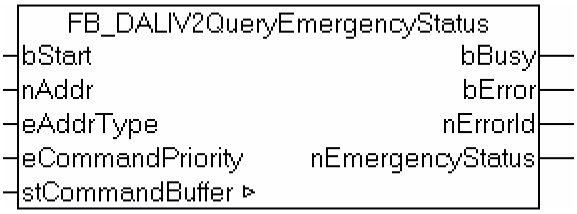
nEmergencyMode: Emergency mode (see table above).

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.183 FB_DALIV2QueryEmergencyStatus



The EMERGENCY STATUS [▶ 370] variable (status of emergency mode) is read from the control gear.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

Bit	Description
0	Inhibit mode. 0: no.
1	Function test is completed and result is valid. 0: no.
2	Duration test is completed and result is valid. 0: no.
3	Battery charger ready for operation. 0: running.
4	Start of function test delayed. 0: no.
5	Start of duration test delayed. 0: no.
6	Identification active. 0: no.
7	Selected. 0: no

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable device Type 1* command, which can be sent with the function block *FB_DALIV2EnableDeviceType*. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
nEmergencyStatus  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

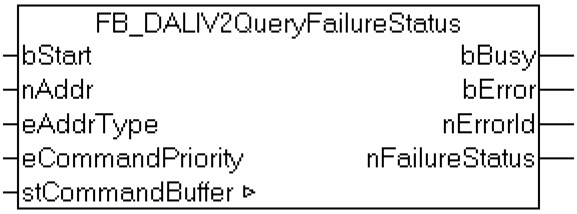
nEmergencyStatus: Status of emergency operation (see table above).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.184 FB_DALIV2QueryFailureStatus



The variable FAILURE STATUS [▶ 371] is read from the control gear.

Bit	Description
0	Error in the control gear circuit. 0: no.
1	Battery operation time fault. 0: no.
2	Battery fault. 0: no.
3	Emergency lamp fault. 0: no.

Bit	Description
4	Timeout during function test. 0: no.
5	Timeout during duration test. 0: no.
6	Function test failed. 0: no.
7	Duration test failed. 0: no

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart                 : BOOL;
nAddr                  : BYTE;
eAddrType              : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority       : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nFailureStatus : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

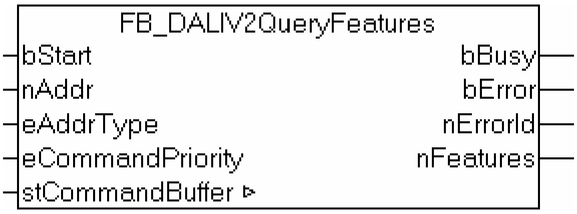
nFailureStatus: Controller failure status (see table above).

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.185 FB_DALIV2QueryFeatures



The variable FEATURES [▸ 371] (performance characteristics) is read from the control gear.

Bit	Description
0	Integrated emergency lighting supply unit. 0: no.
1	Emergency lighting supply unit in continuous mode. 0: no.
2	Switchable emergency lighting supply unit in continuous mode. 0: no.
3	Auto test capability. 0: no.
4	Adjustable emergency lighting illuminance. 0: no.
5	Connected suppress push button is supported. 0: no.
6	Physical selection is supported. 0: no.
7	Reserve

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nFeatures      : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

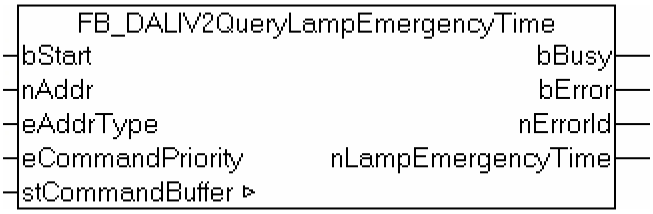
nFeatures: Features of the controller (see table above).

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.186 FB_DALIV2QueryLampEmergencyTime



The variable LAMP EMERGENCY TIME [▸ 370] is read from the control gear. The unit is 1 hour per step. 255 means a maximum value of 254 hours or more. The variable is always incremented at the start of the 1-hour interval. Once the maximum value of 255 is reached, it is not increased further. The variable is reset via the command FB_DALIV2ResetLampTime() [▸ 238].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority      : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                 : BOOL;
bError                : BOOL;
nErrorId              : UDINT;
nLampEmergencyTime    : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

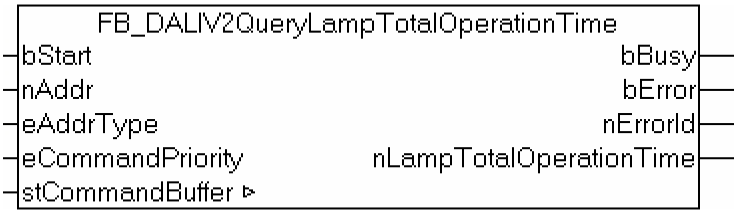
nLampEmergencyTime: Emergency operation time of the lamp from the ballast.

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.187 FB_DALIV2QueryLampTotalOperationTime



The variable LAMP TOTAL OPERATION TIME [▶ 370] is read from the control gear. The unit is 4 hours per step. 255 means a maximum value of 1016 hours or more. The variable is always incremented at the start of the 4-hour interval. Once the maximum value of 255 is reached, it is not increased further. The variable is reset via the command FB_DALIV2ResetLampTime() [▶ 238].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                      : BOOL;
bError                     : BOOL;
nErrorId                   : UDINT;
nLampTotalOperationTime    : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

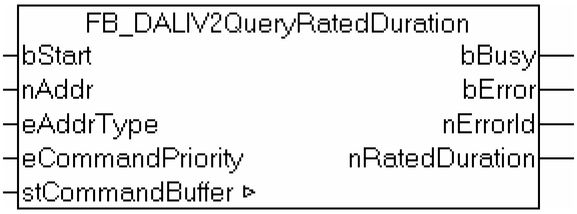
nLampTotalOperationTime: Total operating time of the lamp from the ballast.

VAR_IN_OUT

```
stCommandBuffer       : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.188 FB_DALIV2QueryRatedDuration



The variable RATED DURATION [▸ 370] is read from the control gear. The unit is 2 minutes per step.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
nRatedDuration   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

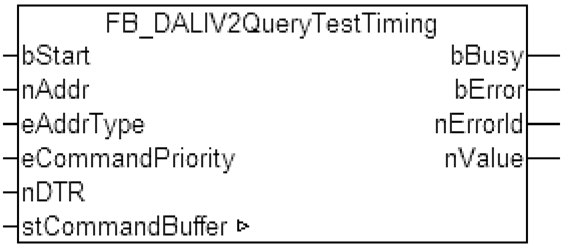
nRatedDuration: Rated operating time of the ballast.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.189 FB_DALIV2QueryTestTiming



This function block reads the test times (intervals, time until next event, etc.) from the control gear. This takes place depending on the value applied at the *nDTR* input. The background to this is that the DALI basic command 242 “Query test timing” works together with the contents of the Data Transfer Register (DTR), which must be written to accordingly beforehand. The function block *FB_DALIV2QueryTestTiming* combines these two actions. The following data are read from the device, depending on the *nDTR* value:

nDTR	read value
0 (2#0000 0000)	If automatic test is activated: time until next **function test** (high-byte) in quarter hours If automatic test is **not** activated: MASK (255)
1 (2#0000 0001)	If automatic test is activated: time until next **function test** (low-byte) in quarter hours If automatic test is **not** activated: MASK (255)
2 (2#0000 0010)	If automatic test is activated: time until next **duration test** (high-byte) in quarter hours If automatic test is **not** activated: MASK (255)
3 (2#0000 0011)	If automatic test is activated: time until next **duration test** (low-byte) in quarter hours If automatic test is **not** activated: MASK (255)
4 (2#0000 0100)	If automatic test is activated: test interval of the **function test** in days If automatic test is **not** activated: MASK (255)
5 (2#0000 0101)	If automatic test is activated: test interval of the **duration test** in weeks If automatic test is **not** activated: MASK (255)
6 (2#0000 0110)	Test execution timeout in days (maximum execution period for one test). Applies only to the automatic test!
7 (2#0000 0111)	Lamp-on extension time after exiting from emergency operating mode, measured in 0.5-min steps

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDTR              : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nDTR: Depending on this input, the specific values will be read out of the device.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nValue         : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

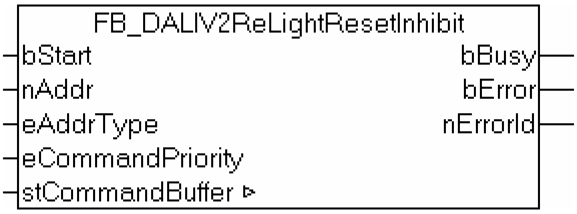
nValue: Read value.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.190 FB_DALIV2ReLightResetInhibit



The control gear is switched back to emergency mode (in the absence of mains voltage). This deactivates the function of the function block FB_DALIV2Inhibit() [▸ 220].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

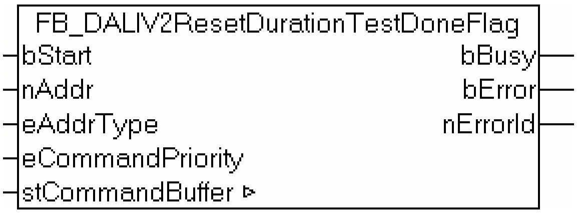
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.191 FB_DALIV2ResetDurationTestDoneFlag



The 'Duration test completed and result is valid' flag is reset. The flag is bit 2 of the variable EMERGENCY STATUS [▶ 371] and can be queried with the function block FB_DALIQueryEmergencyStatus() [▶ 227].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

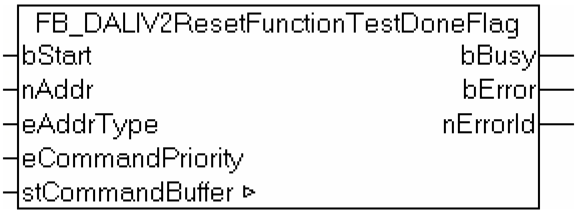
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.192 FB_DALIV2ResetFunctionTestDoneFlag



The 'Function test completed and result is valid' flag is reset. The flag is bit 1 of the variable EMERGENCY STATUS [▸ 371] and can be queried with the function block FB_DALIQueryEmergencyStatus() [▸ 227].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

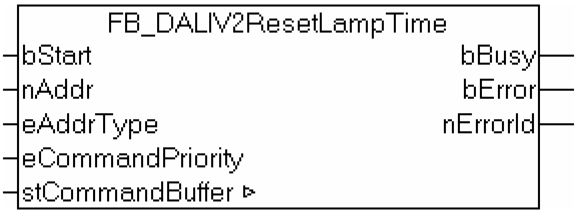
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.193 FB_DALIV2ResetLampTime



Die variables LAMP EMERGENCY TIME [▸ 370] and LAMP TOTAL OPERATION TIME [▸ 370] are reset.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

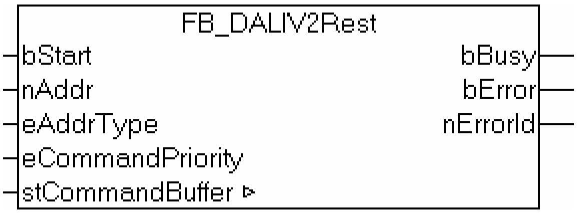
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.194 FB_DALIV2Rest



This function block switches the lamp off if emergency mode is active. Otherwise the system switches to normal operation if mains voltage is available again, or if the function block FB_DALIV2ReLightResetInhibit() [▸ 235] was called.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart                : BOOL;
nAddr                 : BYTE;
eAddrType             : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority      : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

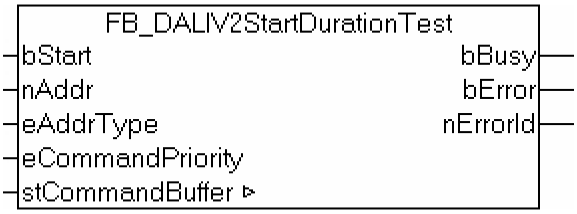
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.195 FB_DALIV2StartDurationTest



Starts the duration test. If the duration test is started with a delay, this is indicated in bit 5 of the variable EMERGENCY STATUS [▶ 371].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

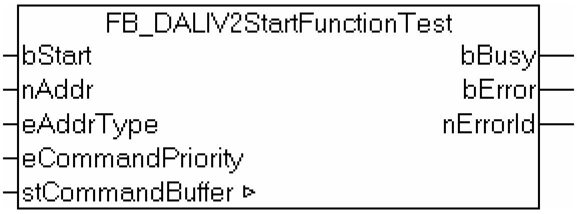
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.196 FB_DALIV2StartFunctionTest



Starts the function test. If the function test is started with a delay, this is indicated in bit 4 of the variable EMERGENCY STATUS [▶ 371].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

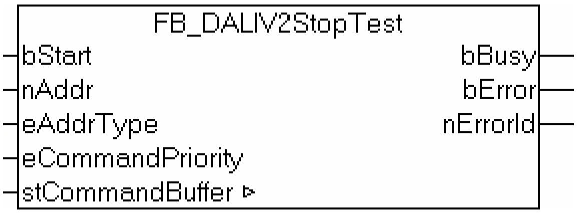
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.197 FB_DALIV2StopTest



Stops any type of function test or duration test.

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

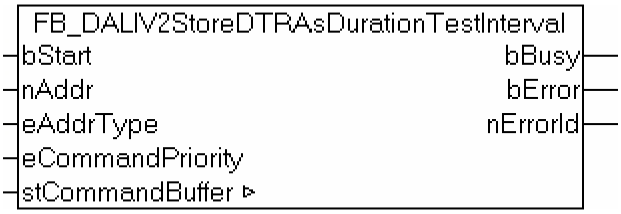
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.198 FB_DALIV2StoreDTRAsDurationTestInterval



Writes the value of the DTR (data transfer register) into the variable DURATION TEST INTERVAL [▸ 369].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

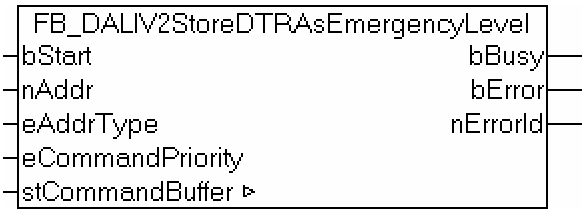
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.199 FB_DALIV2StoreDTRAsEmergencyLevel



Writes the value of the DTR (data transfer register) into the variable EMERGENCY LEVEL [▶ 368] (emergency illuminance).

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

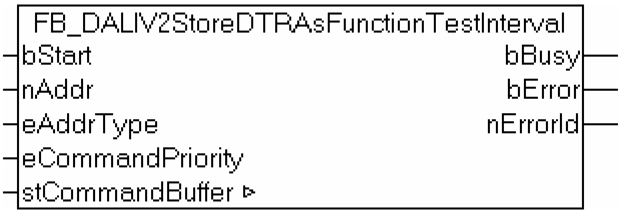
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.200 FB_DALIV2StoreDTRAsFunctionTestInterval



Writes the value of the DTR (data transfer register) into the variable FUNCTION TEST INTERVAL [▸ 369].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

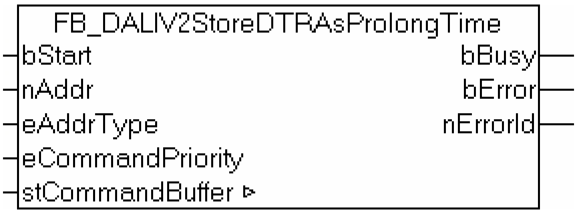
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.201 FB_DALIV2StoreDTRAsProlongTime



Writes the value of the DTR (data transfer register) into the variable PROLONG TIME [▶ 368].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

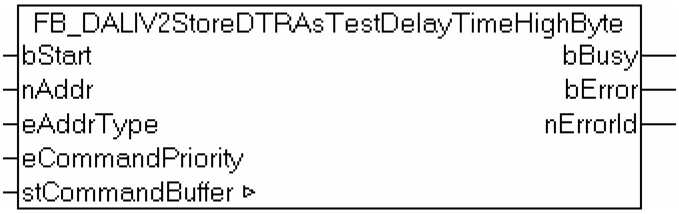
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.202 FB_DALIV2StoreDTRAsTestDelayTimeHighByte



Writes the value of the DTR (data transfer register) into the high-order byte of variable TEST DELAY TIME [▸ 368].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

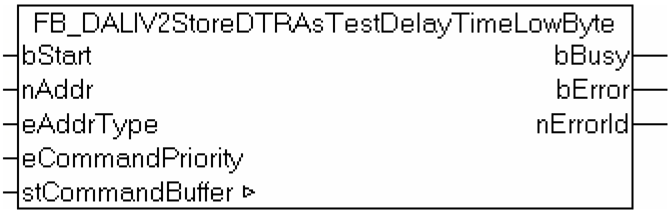
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.203 FB_DALIV2StoreDTRAsTestDelayTimeLowByte



Writes the value of the DTR (data transfer register) into the low-order byte of variable TEST DELAY TIME [▸ 368].

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

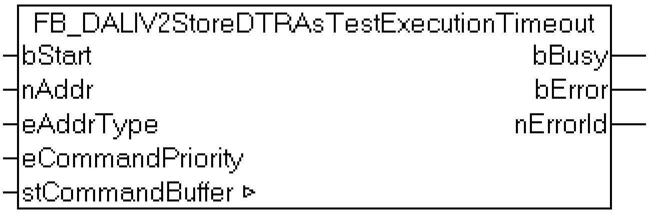
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.204 FB_DALIV2StoreDTRAsTestExecutionTimeout



Writes the value of the DTR (data transfer register) into the variable TEST EXECUTION TIMEOUT [▶ 370] (maximum execution time for a test).

This command belongs to the application extended commands for DALI emergency lighting. These only work if they are preceded by the *Enable Device Type 1* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 1* command is internally placed automatically before all application extended commands for DALI emergency lighting.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

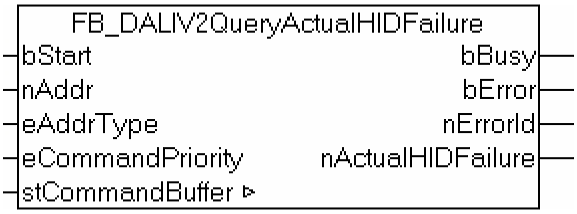
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.205 FB_DALIV2QueryActualHIDFailure



The variable ACTUAL HID FAILURE [▸ 372] (current failure status) is read from the control gear.

Bit	Description
0	Supply voltage too low. 0: no.
1	Supply voltage too high. 0: no.
2	Converter too hot. 0: no.
3	Reserved
4	Time for lamp ignition exceeded. 0: no.
5	Reserved
6	The lamp voltage outside specification. 0: no.
7	Lamp cycle error. 0: no.

This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy               : BOOL;
bError              : BOOL;
nErrorId            : UDINT;
nActualHIDFailure   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

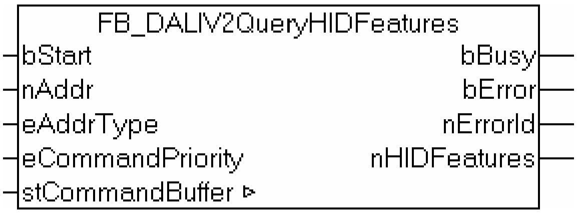
nActualHIDFailure: Content of the ACTUAL HID FAILURE variable.

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

sstCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.206 FB_DALIV2QueryHIDFeatures



The variable HID FEATURES [▸ 373] (performance characteristics) is read from the control gear.

Bit	Description
0	"Supply voltage too low" can be queried. 0: no.
1	"Supply voltage too high" can be queried. 0: no.
2	"Transformer too hot" can be queried. 0: no.
3	Reserve
4	Reserve
5	Reserve
6	"Lamp voltage outside specification" can be queried. 0: no.
7	Physical selection is supported. 0: no.

This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nHIDFeatures    : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

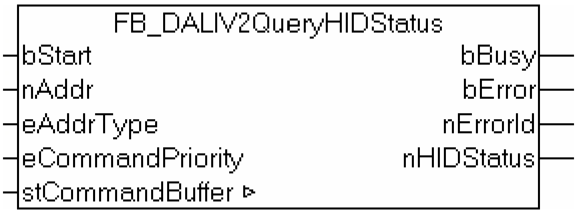
nHIDFeatures: Content of the HID FEATURES variable.

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.207 FB_DALIV2QueryHIDStatus



The variable HID STATUS [▶ 372] is read from the control gear.

Bit	Description
0	Start-up time, ready for operation. 0: no.
1	The lamp power has reached its required setpoint. 0: no.
2	Control gear is waiting for the lamp to ignite. 0: no.
3	Reserve
4	Reserve
5	Reserve
6	Identification active. 0: no.
7	Reserve

This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nHIDStatus      : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

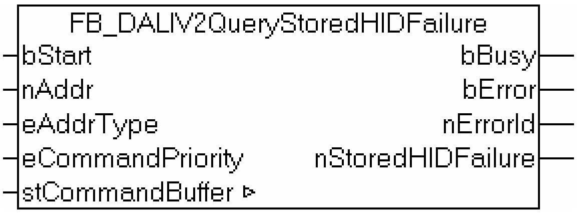
nHIDStatus: Content of the HID STATUS variable.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.208 FB_DALIV2QueryStoredHIDFailure



The variable STORED HID FAILURE [▶ 373] (stored failure status) is read from the control gear.

Bit	Description
0	Supply voltage too low. 0: no.
1	Supply voltage too high. 0: no.
2	Converter too hot. 0: no.
3	Reserve
4	Time for lamp ignition exceeded. 0: no.
5	Reserve
6	The lamp voltage outside specification. 0: no.
7	Lamp cycle error. 0: no.

This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nStoredHIDFailure   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

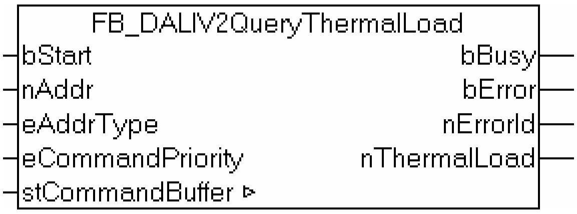
nStoredHIDFailure: Content of the STORED HID FAILURE variable.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.209 FB_DALIV2QueryThermalLoad



The variable THERMAL LOAD [▶ 373] is read. This contains a percentage figure in the range from 0% to 127.5% in steps of 0.5%.

> This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nThermalLoad  : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

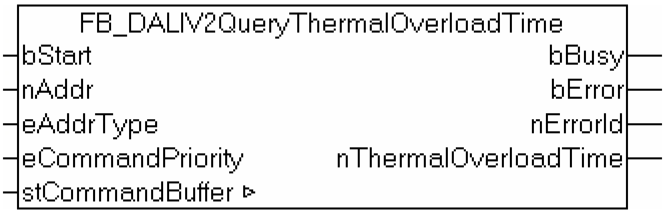
nThermalLoad: Content of the THERMAL LOAD variable.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.210 FB_DALIV2QueryThermalOverloadTime



The 16-bit variable THERMAL OVERLOAD TIME [▶ 373] is read.

> This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                   : BOOL;
bError                  : BOOL;
nErrorId                : UDINT;
nThermalOverloadTime    : WORD;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

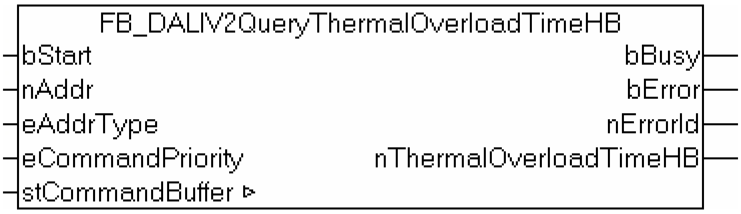
nThermalOverloadTime: The value of the THERMAL OVERLOAD TIME variable.

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.211 FB_DALIV2QueryThermalOverloadTimeHB



The high-order byte of the 16-bit variable THERMAL OVERLOAD TIME [▶ 373] is read.

This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                     : BOOL;
bError                    : BOOL;
nErrorId                  : UDINT;
nThermalOverloadTimeHB    : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

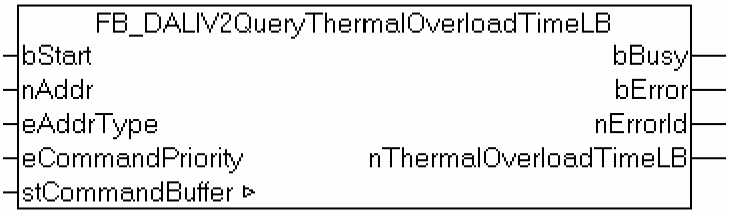
nThermalOverloadTimeHB: The high-order byte of the THERMAL OVERLOAD TIME variable.

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.212 FB_DALIV2QueryThermalOverloadTimeLB



The low-order byte of the 16-bit variable THERMAL OVERLOAD TIME [▶ 373] is read.

This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                    : BOOL;
bError                   : BOOL;
nErrorId                 : UDINT;
nThermalOverloadTimeLB   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

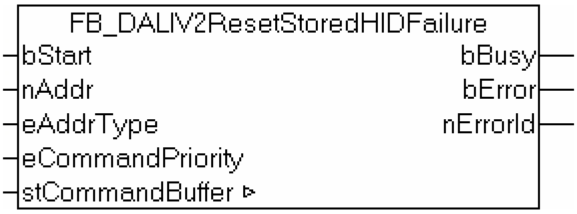
nThermalOverloadTimeLB: The low-order byte of the THERMAL OVERLOAD TIME variable.

VAR_IN_OUT

```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.213 FB_DALIV2ResetStoredHIDFailure



The variable STORED HID FAILURE [▶ 373] (stored lamp faults) is reset.

This command belongs to the application extended commands for DALI discharge lamps. These only work if they are preceded by the *Enable Device Type 2* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 2* command is internally placed automatically before all application extended commands for DALI discharge lamps.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

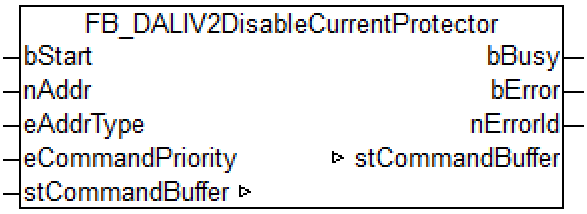
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.214 FB_DALIV2DisableCurrentProtector



The command disables the current protection device of the control gear.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

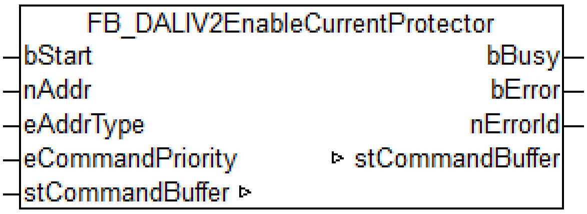
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.215 FB_DALIV2EnableCurrentProtector



The command enables the current protection device of the control gear.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

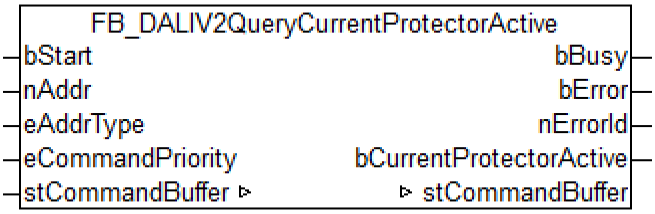
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.216 FB_DALIV2QueryCurrentProtectorActive



The system queries whether the current protection device is active.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *Tc-DALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy                       : BOOL;
bError                      : BOOL;
nErrorId                    : UDINT;
bCurrrentProtectorActive    : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

bCurrrentProtectorActive: Current protection device active.

VAR_IN_OUT

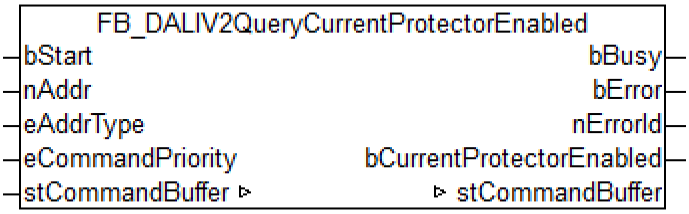
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.217 FB_DALIV2QueryCurrentProtectorEnabled



The system queries whether the current protection device is enabled.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy                       : BOOL;
bError                      : BOOL;
nErrorId                    : UDINT;
bCurrentProtectorEnabled    : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

bCurrentProtectorEnabled: Current protection device enabled.

VAR_IN_OUT

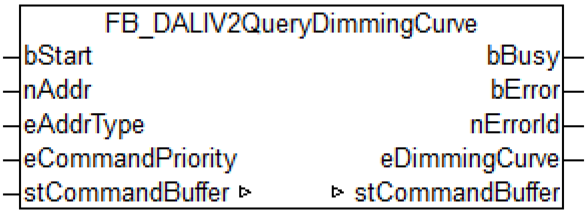
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.218 FB_DALIV2QueryDimmingCurve



The dimming curve of the control gear is read out.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
eDimmingCurve : E_DALIV2DimmingCurve;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

eDimmingCurve: Dimming curve [▸ 381] (linear or logarithmic).

VAR_IN_OUT

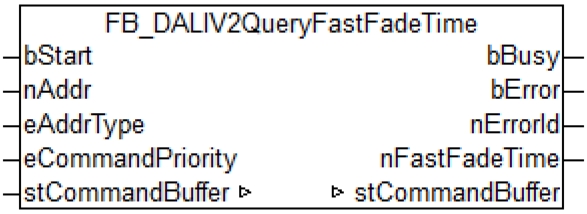
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.219 FB_DALIV2QueryFastFadeTime



Queries the value of FAST FADE TIME.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *Tc-DALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nFastFadeTime : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nFastFadeTime: Value of FAST FADE TIME.

VAR_IN_OUT

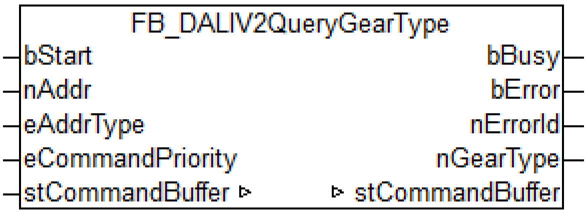
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.220 FB_DALIV2QueryGearType



Queries the value of GEAR TYPE.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nGearType     : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nGearType: Value of GEAR TYPE.

VAR_IN_OUT

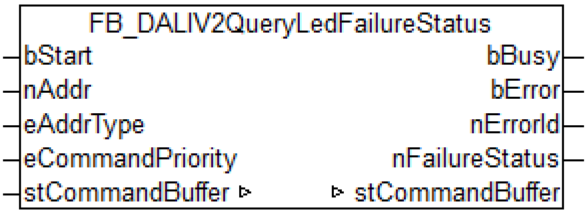
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.221 FB_DALIV2QueryLedFailureStatus



Queries the value of FAILURE STATUS.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nFailureStatus: BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nFailureStatus: Value of FAILURE STATUS.

VAR_IN_OUT

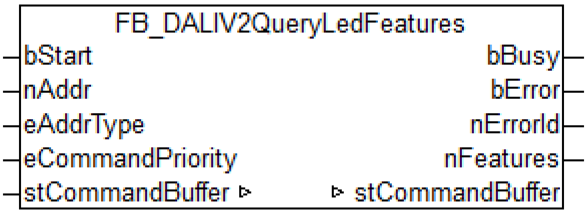
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.222 FB_DALIV2QueryLedFeatures



Queries the value of FEATURES.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address, group address or broadcast.

eCommandPriority: Priority (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nFeatures     : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nFeatures: Value of FEATURES.

VAR_IN_OUT

```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

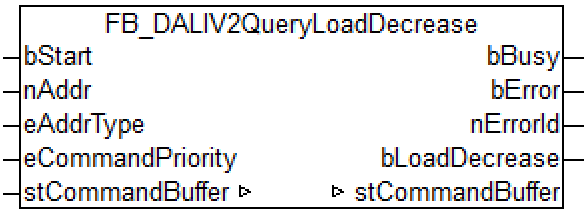
Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

Also see about this

- E_DALIV2AddrType [▸ 380]
- E_DALIV2CommandPriority [▸ 381]

5.1.223 FB_DALIV2QueryLoadDecrease



The system queries whether a significant decrease in load (compared to the reference power of the system) has been detected.

> This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *Tc-DALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
bLoadDecrease : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

bLoadDecrease: Load decrease.

VAR_IN_OUT

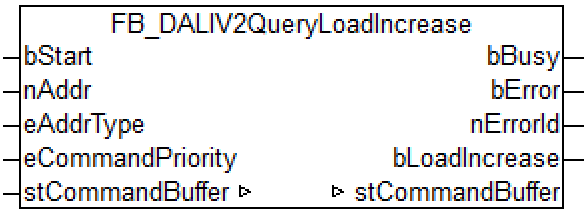
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.224 FB_DALIV2QueryLoadIncrease



The system queries whether a significant increase in load (compared to the reference power of the system) has been detected.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *Tc-DALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
bLoadIncrease : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bLoadIncrease: Load increase.

VAR_IN_OUT

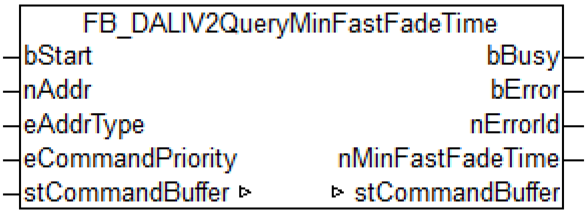
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.225 FB_DALIV2QueryMinFastFadeTime



Queries the value of MIN FAST FADE TIME.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nMinFastFadeTime   : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nMinFastFadeTime: Value of MIN FAST FADE TIME.

VAR_IN_OUT

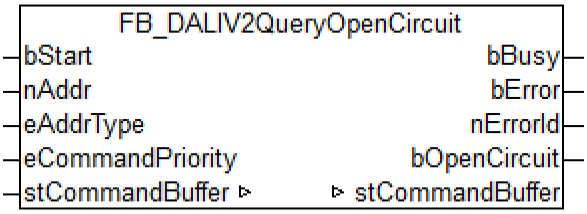
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.226 FB_DALIV2QueryOpenCircuit



The system queries whether an idle mode has been detected.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
bOpenCircuit  : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bOpenCircuit: Idle mode.

VAR_IN_OUT

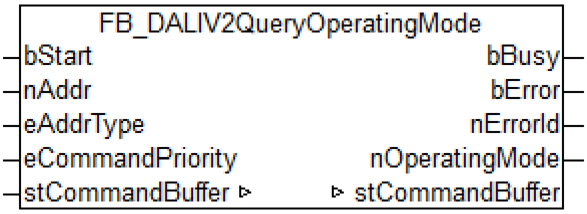
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.227 FB_DALIV2QueryOperatingMode



Queries the value of OPERATING MODE.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nOperatingMode: BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nOperatingMode: Value of OPERATING MODE.

VAR_IN_OUT

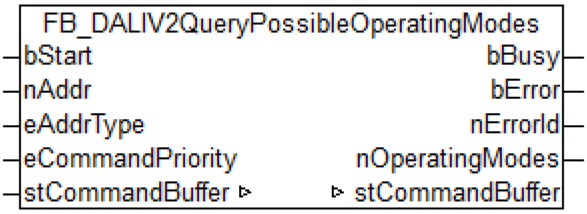
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.228 FB_DALIV2QueryPossibleOperatingModes



Queries the value of OPERATING MODE.

> This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
nOperatingModes   : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nOperatingModes: Value of OPERATING MODE.

VAR_IN_OUT

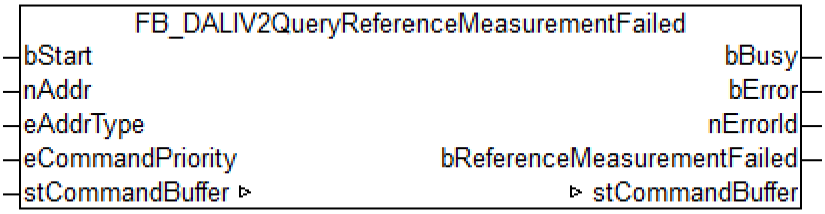
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.229 FB_DALIV2QueryReferenceMeasurementFailed



The system queries whether a started reference measurement has failed.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy                        : BOOL;
bError                       : BOOL;
nErrorId                     : UDINT;
bReferenceMeasurementFailed  : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

bReferenceMeasurementFailed: Reference measurement failed.

VAR_IN_OUT

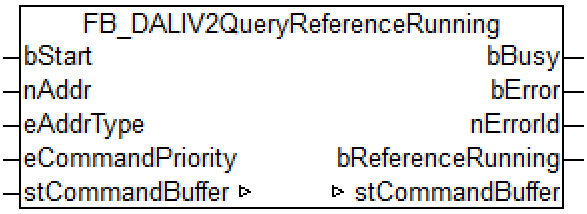
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.230 FB_DALIV2QueryReferenceRunning



The system queries whether a reference measurement of the system performance is active.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy               : BOOL;
bError              : BOOL;
nErrorId            : UDINT;
bReferenceRunning   : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bReferenceRunning: Reference measurement of system performance is active.

VAR_IN_OUT

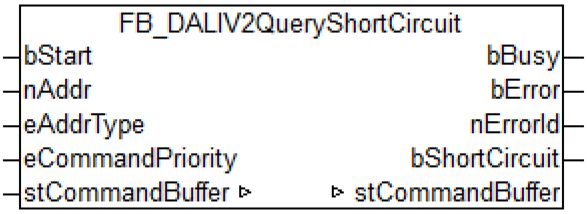
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.231 FB_DALIV2QueryShortCircuit



The system queries whether a short circuit has been detected.

> This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
bShortCircuit : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bShortCircuit: Short circuit.

VAR_IN_OUT

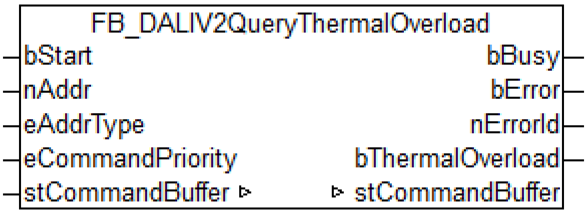
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.232 FB_DALIV2QueryThermalOverload



The system queries whether there is a thermal overload with reduction of the luminous flux.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *Tc-DALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
bThermalOverload   : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bThermalOverload: Thermal overload.

VAR_IN_OUT

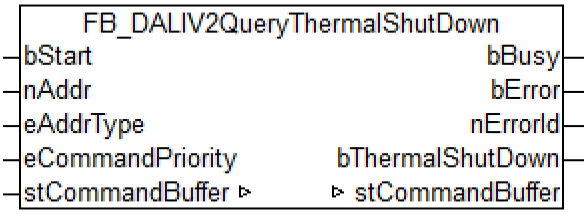
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.233 FB_DALIV2QueryThermalShutDown



The system queries whether a thermal shutdown occurred.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *Tc-DALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
bThermalShutDown   : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bThermalShutDown: Thermal shutdown.

VAR_IN_OUT

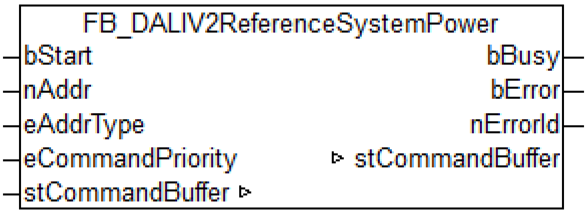
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.234 FB_DALIV2ReferenceSystemPower



The control gear measures and stores the performance level of the system, in order to detect load increase and decrease.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

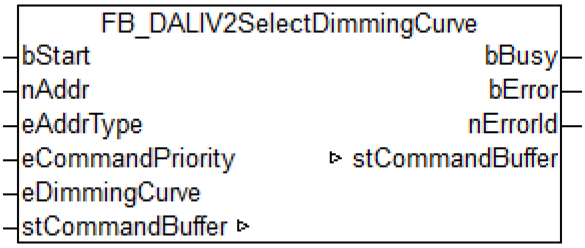
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.235 FB_DALIV2SelectDimmingCurve



The dimming curve of the control gear is selected.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
eDimmingCurve      : E_DALIV2DimmingCurve := eDALIV2DimmingCurveLogarithmic;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

eDimmingCurve: Linear or logarithmic dimming curve [▸ 381].

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

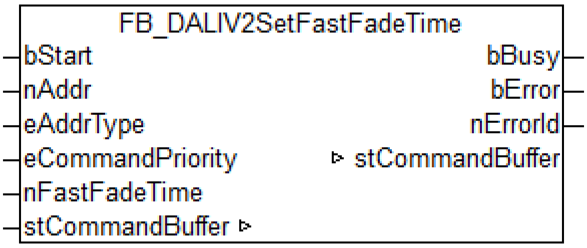
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.236 FB_DALIV2SetFastFadeTime



Sets the FAST FADE TIME in the control gear.

This command is one of the application extended commands for LED modules with DALI interface. These only work if they are preceded by the *Enable Device Type 6* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 6* command is internally placed automatically before all application extended commands for LED modules with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nFastFadeTime       : BYTE;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nFastFadeTime: The new value for the FAST FADE TIME (0 - 27).

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

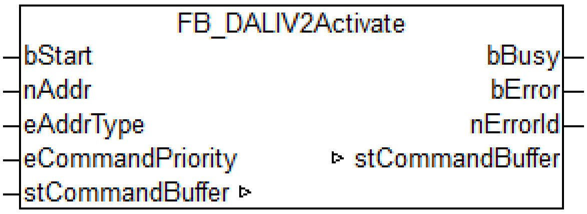
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function blocks FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.237 FB_DALIV2Activate



A current cross-fade is ended and a new cross-fade is started. Only the color is changed here.

The function block supports the following color representations:

- xy coordinates
- Color temperature Tc
- Primary (color) N
- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

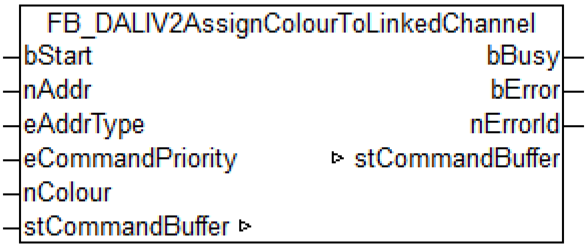
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.238 FB_DALIV2AssignColourToLinkedChannel



Linked output channels are assigned to the defined color (see table). The linked channels are specified by bit 0 to bit 5 of the TEMPORARY RGBWAF CONTROL [▶ 374] variable. The channel assignment is not changed if TEMPORARY RGBWAF CONTROL 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 DTR (Data Transfer Register) of all DALI control gears in the DALI line is changed by the function block.

The function block supports the following color representations:

- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nColour             : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nColour: Colour that is assigned to the output channels (see table).

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

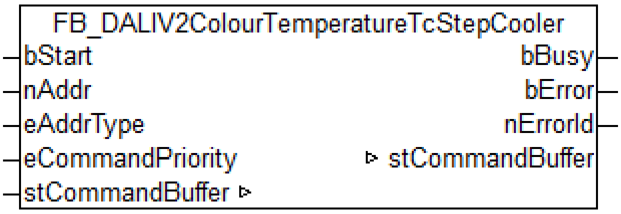
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.239 FB_DALIV2ColourTemperatureTcStepCooler



The COLOUR TEMPERATURE Tc [▶ 374] value is decremented by 1 Mirek without cross-fading. If the value COLOUR TEMPERATURE Tc [▶ 374] already has the value COLOUR TEMPERATURE Tc COOLEST [▶ 374], no change is made.
If the color temperature cannot be reached by the DALI control gear, bit 1 (Colour temperature Tc out of range) is set in COLOUR STATUS [▶ 374]. This command is only executed by the DALI control gear if bit 5 (Color type color temperature Tc active) is set in the variable COLOUR STATUS [▶ 374].
For the conversion from or to Kelvin the functions KELVIN_TO_MIREK [▶ 319] and MIREK_TO_KELVIN [▶ 319] are available.

The function block supports the following color representations:

- Color temperature Tc

> This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

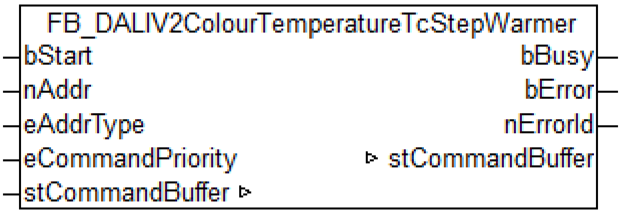
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.240 FB_DALIV2ColourTemperatureTcStepWarmer



The COLOUR TEMPERATURE Tc [▶ 374] value is incremented by 1 Mirek without cross-fading. If the value COLOUR TEMPERATURE Tc [▶ 374] already has the value COLOUR TEMPERATURE Tc WARMEST [▶ 374], no change is made.
If the color temperature cannot be reached by the DALI control gear, bit 1 (Color temperature Tc out of range) is set in COLOUR STATUS [▶ 374]. This command is only executed by the DALI control gear if bit 5 (Color type color temperature Tc active) is set in the variable COLOUR STATUS [▶ 374].
For the conversion from or to Kelvin the functions KELVIN_TO_MIREK [▶ 319] and MIREK_TO_KELVIN [▶ 319] are available.

The function block supports the following color representations:

- Color temperature Tc

> This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

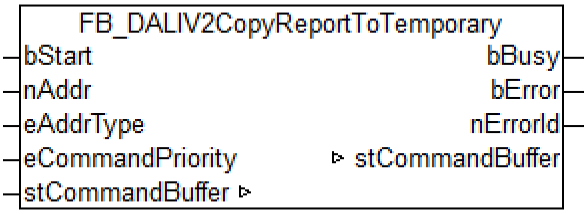
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.241 FB_DALIV2CopyReportToTemporary



The color settings report is copied to the temporary color settings.

The function block supports the following color representations:

- xy coordinates
- Color temperature Tc
- Primary (color) N
- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

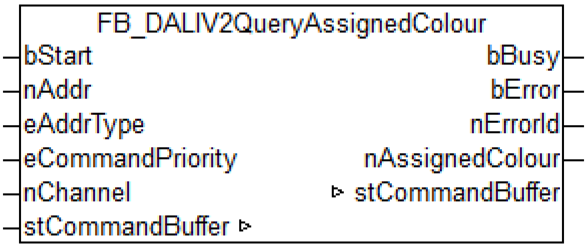
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.242 FB_DALIV2QueryAssignedColour



The variable ASSIGNED COLOUR [▸ 378] is read from the DALI control gear. This contains the color (see table) assigned to the specified output channel (0 - 5). 255 (MASK) is returned if a non-existent channel number is specified.

Value	Description
0	No color assigned
1	Red
2	Green
3	Blue
4	White
5	Amber
6	Freely selectable color

The DTR (Data Transfer Register) of all DALI control gears in the DALI line is changed by the function block.

The function block supports the following color representations:

- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nChannel            : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nChannel: Channel (0 - 5).

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
nAssignedColour    : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

nAssignedColour: Assigned colour of the channel (see table).

VAR_IN_OUT

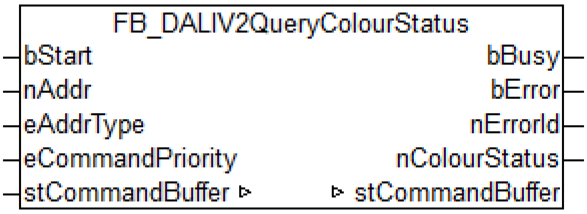
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.243 FB_DALIV2QueryColourStatus



The variable COLOUR STATUS [▶ 378]is read from the DALI control gear.

Bit	Description
0	xy-coordinate color point lies 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 *colour temperature Tc* active.
6	Color representation *primary N* active.
7	Color representation *RGBWAF* active.

The function block supports the following color representations:

- xy coordinates
- Color temperature Tc
- Primary (color) N
- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nColourStatus   : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

nColourStatus: Colour status (see table above).

VAR_IN_OUT

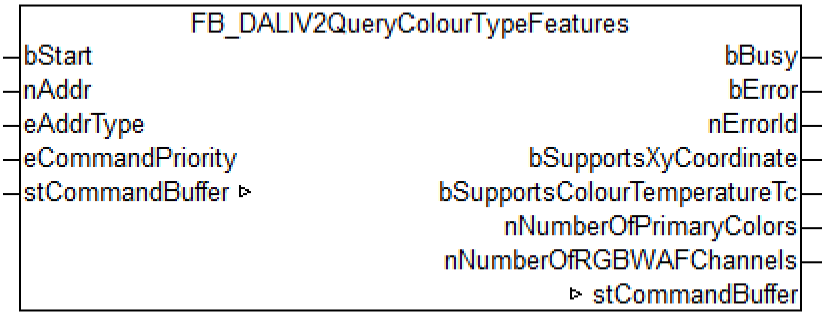
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.244 FB_DALIV2QueryColourTypeFeatures



The variable COLOUR TYPE FEATURES [▶ 379] is read from the DALI control gear. This contains the color representations supported by the DALI control gear.

The function block supports the following color representations:

- xy coordinates
- Color temperature Tc
- Primary (color) N
- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy                           : BOOL;
bError                          : BOOL;
nErrorId                        : UDINT;
bSupportsXyCoordinate           : BOOL;
bSupportsColourTemperatureTc    : BOOL;
nNumberOfPrimaryColors          : BYTE;
nNumberOfRGBWAFChannels         : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

bSupportsXyCoordinate: The ballast is *xy-coordinate*-capable.

bSupportsColourTemperatureTc: The ballast is *Colour temperature Tc*-capable.

nNumberOfPrimaryColors: The number of primary colours supported by the ballast. A value of 0 means that this colour representation is not supported.

nNumberOfRGBWAFChannels: The number of RGBWAF channels supported by the ballast. A value of 0 means that this colour representation is not supported.

VAR_IN_OUT

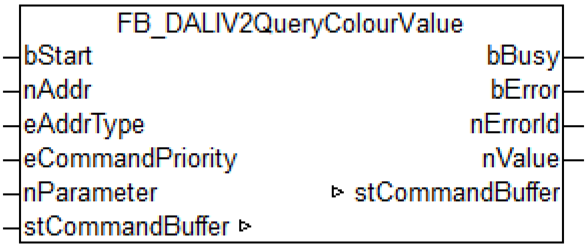
```
stCommandBuffer             : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.245 FB_DALIV2QueryColourValue



The specified variable (color value) is read from the DALI control gear. The value to be read is defined by *nParameter* (see table below).

Certain variables can be read out directly via DALI commands (e.g. FB_QueryColourStatus() [▸ 290] or FB_QueryRGBWAFControl() [▸ 297]). Further details on the variables can be found here [▸ 374].

Value	Description
0	x-COORDINATE
1	y-COORDINATE
2	COLOUR TEMPERATURE Tc
3	PRIMARY N DIMLEVEL 0
4	PRIMARY N DIMLEVEL 1
5	PRIMARY N DIMLEVEL 2
6	PRIMARY N DIMLEVEL 3
7	PRIMARY N DIMLEVEL 4
8	PRIMARY N DIMLEVEL 5
9	RED DIMLEVEL
10	GREEN DIMLEVEL
11	BLUE DIMLEVEL
12	WHITE DIMLEVEL
13	AMBER DIMLEVEL
14	FREECOLOUR DIMLEVEL
15	RGBWAF CONTROL
64	x-COORDINATE PRIMARY N 0
65	y-COORDINATE PRIMARY N 0
66	TY PRIMARY N 0
67	x-COORDINATE PRIMARY N 1
68	y-COORDINATE PRIMARY N 1
69	TY PRIMARY N 1
70	x-COORDINATE PRIMARY N 2
71	y-COORDINATE PRIMARY N 2
72	TY PRIMARY N 2
73	x-COORDINATE PRIMARY N 3
74	y-COORDINATE PRIMARY N 3
75	TY PRIMARY N 3
76	x-COORDINATE PRIMARY N 4
77	y-COORDINATE PRIMARY N 4
78	TY PRIMARY N 4
79	x-COORDINATE PRIMARY N 5
80	y-COORDINATE PRIMARY N 5
81	TY PRIMARY N 5
82	NUMBER OF PRIMARIES

Value	Description
128	COLOUR TEMPERATURE Tc COOLEST
129	COLOUR TEMPERATURE Tc PHYSICAL COOLEST
130	COLOUR TEMPERATURE Tc WARMEST
131	COLOUR TEMPERATURE Tc PHYSICAL WARMEST
192	TEMPORARY x-COORDINATE
193	TEMPORARY y-COORDINATE
194	TEMPORARY COLOUR TEMPERATURE Tc
195	TEMPORARY PRIMARY N DIMLEVEL 0
196	TEMPORARY PRIMARY N DIMLEVEL 1
197	TEMPORARY PRIMARY N DIMLEVEL 2
198	TEMPORARY PRIMARY N DIMLEVEL 3
199	TEMPORARY PRIMARY N DIMLEVEL 4
200	TEMPORARY PRIMARY N DIMLEVEL 5
201	TEMPORARY RED DIMLEVEL
202	TEMPORARY GREEN DIMLEVEL
203	TEMPORARY BLUE DIMLEVEL
204	TEMPORARY WHITE DIMLEVEL
205	TEMPORARY AMBER DIMLEVEL
206	TEMPORARY FREECOLOUR DIMLEVEL
207	TEMPORARY RGBWAF CONTROL
208	TEMPORARY COLOUR TYPE
224	REPORT x-COORDINATE
225	REPORT y-COORDINATE
226	REPORT COLOUR TEMPERATURE Tc
227	REPORT PRIMARY N DIMLEVEL 0
228	REPORT PRIMARY N DIMLEVEL 1
229	REPORT PRIMARY N DIMLEVEL 2
230	REPORT PRIMARY N DIMLEVEL 3
231	REPORT PRIMARY N DIMLEVEL 4
232	REPORT PRIMARY N DIMLEVEL 5
233	REPORT RED DIMLEVEL
234	REPORT GREEN DIMLEVEL
235	REPORT BLUE DIMLEVEL
236	REPORT WHITE DIMLEVEL
237	REPORT AMBER DIMLEVEL
238	REPORT FREECOLOUR DIMLEVEL
239	REPORT RGBWAF CONTROL
240	REPORT COLOUR TYPE

Responses that concern an active color representation are valid only if the color representation of the requested color value is active (see FB_DALIV2QueryColourStatus() [▶ 290]) or if the control gear is capable or reconverting the demanded color value from the active color representation to a color value of another color representation.

The response must be 255 (MASK) if the control gear does not know the coordinates or if the primary color is not present.

The DTR (Data Transfer Register) and DTR1 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- xy coordinates

- Color temperature Tc
- Primary (color) N
- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart           : BOOL;
nAddr            : BYTE;
eAddrType        : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nParameter       : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nParameter: Colour value to be read out (see table above).

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
nErrorId     : UDINT;
nValue       : UINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

nValue: Contains the value read out.

VAR_IN_OUT

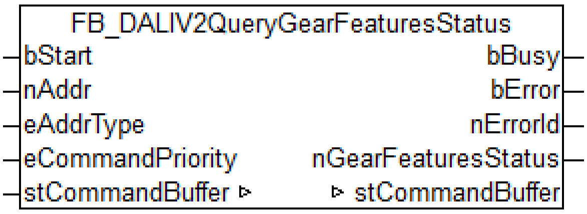
```
stCommandBuffer  : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.246 FB_DALIV2QueryGearFeaturesStatus



The variable GEAR FEATURES/STATUS [▸ 379] is read from the DALI control gear.

Bit	Description
0	Automatic activation.
1 - 5	reserved.
6	Automatic calibration is supported.
7	Restoration of the automatic calibration is supported.

The function block supports the following color representations:

- xy coordinates
- Color temperature Tc
- Primary (color) N
- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart           : BOOL;
nAddr            : BYTE;
eAddrType        : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy               : BOOL;
bError              : BOOL;
nErrorId            : UDINT;
nGearFeaturesStatus : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

nGearFeaturesStatus: Status information (see table above).

VAR_IN_OUT

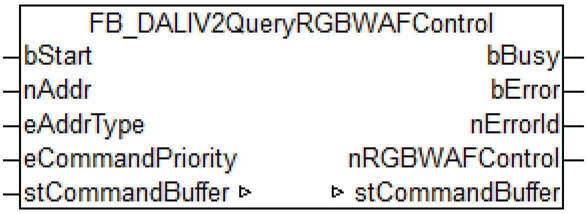
```
stCommandBuffer       : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.247 FB_DALIV2QueryRGBWAFControl



The variable RGBWAF CONTROL [▶ 379] is read from the DALI control gear.

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

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

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nRGBWAFControl: BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

nRGBWAFControl: Information about the channel assignment (see table above).

VAR_IN_OUT

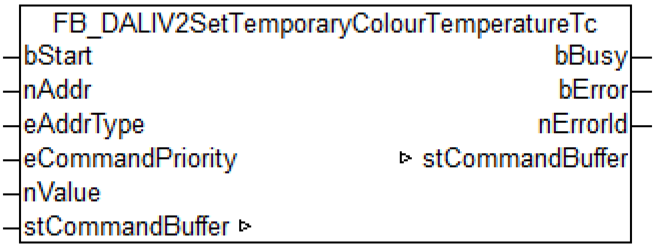
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.248 FB_DALIV2SetTemporaryColourTemperatureTc



Saves the value in the variable TEMPORARY COLOUR TEMPERATURE Tc [▸ 374] of the DALI control gear. The value can be read out using the function block FB_DALIV2QueryColourValue() [▸ 293].

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).
For the conversion from or to Kelvin the functions KELVIN_TO_MIREK [▸ 319] and MIREK_TO_KELVIN [▸ 319] are available.

The DTR (Data Transfer Register) and DTR1 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- Color temperature Tc

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nValue              : UINT;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nValue: The value that is written into the TEMPORARY COLOUR TEMPERATURE Tc variable.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

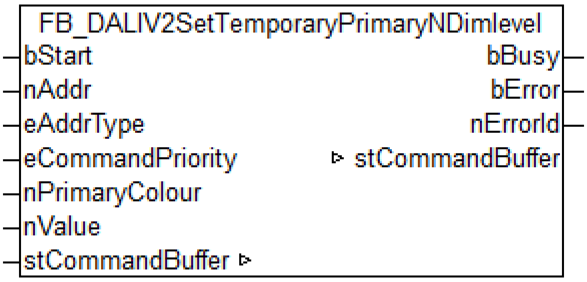
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.249 FB_DALIV2SetTemporaryPrimaryNDimlevel



Saves the value in the variable TEMPORARY PRIMARY N DIMLEVEL [▶ 374] of the control gear. The value can be read out using the function block FB_DALIV2QueryColourValue() [▶ 293].

The value is expressed in units of 1 / 65536. The maximum value of the PRIMARY N DIMLEVEL is 0.99997 and is interpreted on a linear scale.

The DTR (Data Transfer Register), DTR1 and DTR2 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- Primary (color) N

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nPrimaryColour     : BYTE;
nValue             : UINT;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nPrimaryColour: Primary colour (0 - 5).

nValue: The value that is written into the TEMPORARY PRIMARY N DIMLEVEL variable.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

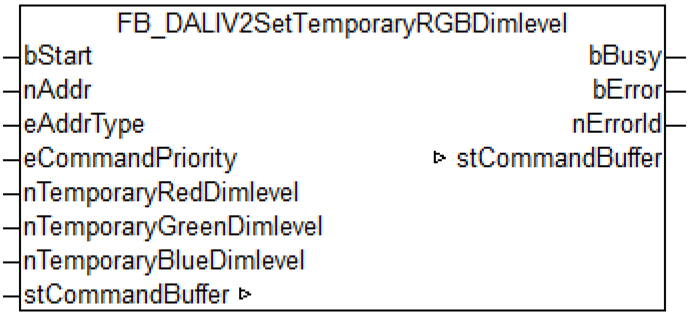
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.250 FB_DALIV2SetTemporaryRGBDimlevel



Saves the values in the variables TEMPORARY RED DIMLEVEL [▸ 374], TEMPORARY GREEN DIMLEVEL [▸ 374] and TEMPORARY BLUE DIMLEVEL [▸ 374] of the DALI control gear. The values can be read out using the function block FB_DALIV2QueryColourValue() [▸ 293].

The DTR (Data Transfer Register), DTR1 and DTR2 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart                   : BOOL;
nAddr                    : BYTE;
eAddrType                : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority         : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nTemporaryRedDimlevel    : BYTE;
nTemporaryGreenDimlevel  : BYTE;
nTemporaryBlueDimlevel   : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nTemporaryRedDimlevel: The new value for TEMPORARY RED DIMLEVEL.

nTemporaryGreenDimlevel: The new value for TEMPORARY GREEN DIMLEVEL.

nTemporaryBlueDimlevel: The new value for TEMPORARY BLUE DIMLEVEL.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

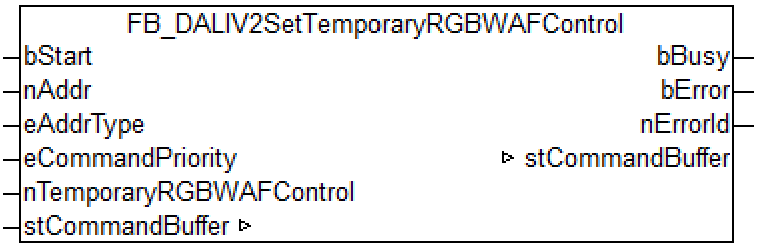
```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.251 FB_DALIV2SetTemporaryRGBWAFControl



Saves the value in the variable TEMPORARY RGBWAF CONTROL [▶ 374] of the DALI control gear. The value can be read out using the function block FB_DALIV2QueryColourValue() [▶ 293]. The *nTemporaryRGBWAFControl* input thereby contains the new assignment (see table).

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

Bit	Description
6 - 7	00 = channel control 01 = color control 10 = standardized color control 11 = reserved

The DTR (Data Transfer Register) of all DALI control gears in the DALI line is changed by the function block.

The function block supports the following color representations:

- RGBWAF

> This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nTemporaryRGBWAFControl  : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nTemporaryRGBWAFControl: Contains the assignment (see table above).

VAR_OUTPUT

```
bBusy             : BOOL;
bError            : BOOL;
nErrorId          : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

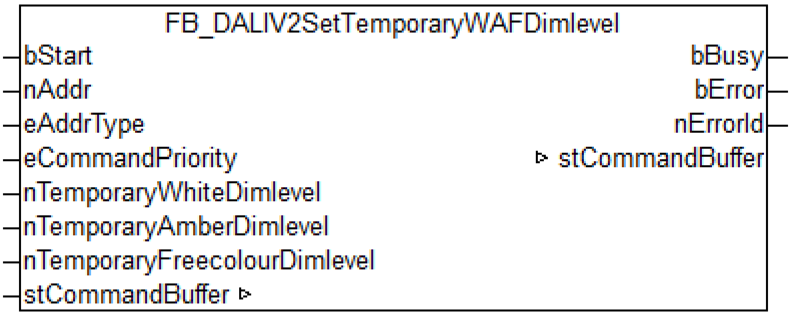
```
stCommandBuffer       : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.252 FB_DALIV2SetTemporaryWAFDimlevel



Saves the values in the variables TEMPORARY WHITE DIMLEVEL [▶ 374], TEMPORARY AMBER DIMLEVEL [▶ 374] and TEMPORARY FREECOLOUR DIMLEVEL [▶ 374] of the DALI control gear. The values can be read out using the function block FB_DALIV2QueryColourValue() [▶ 293].

The DTR (Data Transfer Register), DTR1 and DTR2 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart                        : BOOL;
nAddr                         : BYTE;
eAddrType                     : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority              : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nTemporaryWhiteDimlevel       : BYTE;
nTemporaryAmberDimlevel       : BYTE;
nTemporaryFreecolourDimlevel  : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nTemporaryWhiteDimlevel: The new value for TEMPORARY WHITE DIMLEVEL.

nTemporaryAmberDimlevel: The new value for TEMPORARY AMBER DIMLEVEL.

nTemporaryFreecolourDimlevel: The new value for TEMPORARY FREECOLOUR DIMLEVEL.

VAR_OUTPUT

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

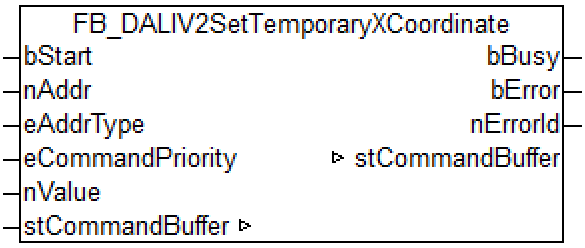
```
stCommandBuffer              : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.253 FB_DALIV2SetTemporaryXCoordinate



Saves the value in the variable TEMPORARY x-COORDINATE [▸ 374] of the DALI control gear. The value can be read out using the function block FB_DALIV2QueryColourValue() [▸ 293].

The DTR (Data Transfer Register) and DTR1 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- xy coordinates
- Primary (color) N

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nValue          : UINT;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nValue: The value that is written into the TEMPORARY x-COORDINATE variable.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

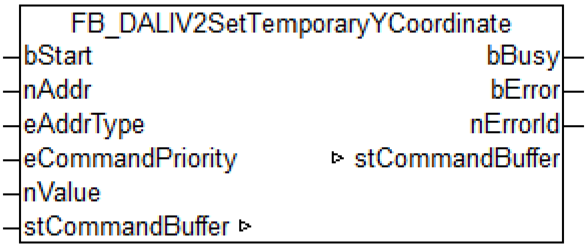
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.254 FB_DALIV2SetTemporaryYCoordinate



Saves the value in the variable TEMPORARY y-COORDINATE [▸ 374] of the DALI control gear. The value can be read out using the function block FB_DALIV2QueryColourValue() [▸ 293].

The DTR (Data Transfer Register) and DTR1 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- xy coordinates
- Primary (color) N

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nValue              : UINT;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nValue: The value that is written into the TEMPORARY y-COORDINATE variable.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

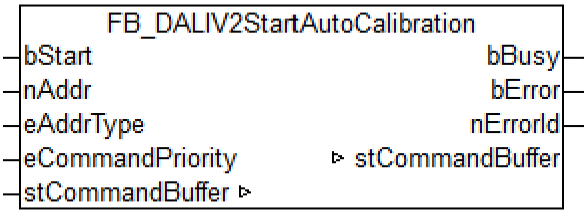
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.255 FB_DALIV2StartAutoCalibration



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 COLOUR STATUS [▸ 378] variable is 1 as long as the timer is active (see FB_DALIV2QueryColourStatus() [▸ 290]). 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 [▶ 164], QUERY COLOUR STATUS [▶ 290] and START AUTO CALIBRATION. In addition, bit 3 in the variable COLOUR STATUS [▶ 378] is set to 0 at the start of the calibration. The TERMINATE [▶ 164] command ends the procedure and stops the timer.

If the calibration was successful, bit 3 in COLOUR STATUS [▶ 378] 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. Subsequently, bit 3 of COLOUR STATUS is set to 1. The ability to restore the last successful calibration data is a feature of the operating device (see QUERY GEAR FEATURES/STATUS [▶ 296] 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 COLOUR STATUS [▶ 290] 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

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

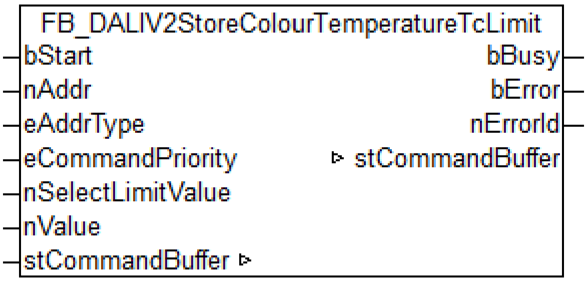
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.256 FB_DALIV2StoreColourTemperatureTcLimit



Saves the value in the variable COLOUR TEMPERATURE Tc COOLEST [▶ 374], COLOUR TEMPERATURE Tc WARMEST [▶ 374], COLOUR TEMPERATURE Tc PHYSICAL COOLEST [▶ 374] or COLOUR TEMPERATURE Tc PHYSICAL WARMEST [▶ 374] of the DALI control gear. The values can be read out using the function block FB_DALIV2QueryColourValue() [▶ 293]. The *nSelectLimitValue* input defines the new limit value to be set:

Value	Limit value	Description
0	COLOUR TEMPERATURE Tc COOLEST	lowest possible value, but always equal to or warmer than the lowest possible physical value.
1	COLOUR TEMPERATURE Tc WARMEST	highest possible value, but always equal to or cooler than the highest possible physical value.
2	COLOUR TEMPERATURE Tc PHYSICAL COOLEST	lowest possible physical value.
3	COLOUR TEMPERATURE Tc PHYSICAL WARMEST	highest possible physical value.

The functions KELVIN_TO_MIREK [▶ 319] and MIREK_TO_KELVIN [▶ 319] are available for converting from or to Kelvin.

The DTR (Data Transfer Register), DTR1 and DTR2 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- Color temperature Tc

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nSelectLimitValue   : BYTE;
nValue              : UINT;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nSelectLimitValue: Specifies the new limit value to be set (see table above).

nValue: The value that is written into the selected variable.

VAR_OUTPUT

```
bBusy            : BOOL;
bError           : BOOL;
nErrorId         : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

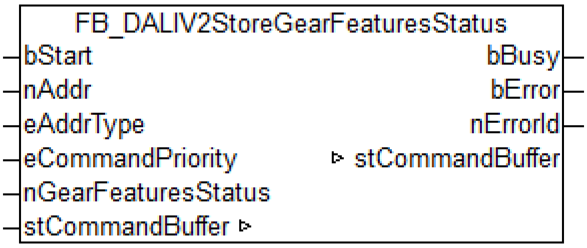
```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.257 FB_DALIV2StoreGearFeaturesStatus



Saves the value in the variable GEAR FEATURES/STATUS [▸ 379] of the DALI control gear. The values can be read with the function block FB_DALIV2QueryGearFeaturesStatus() [▸ 296].

If bit 0 is set to 1, all commands for the control of the lamp power, with the exception of ENABLE DAPC SEQUENCE [▸ 121], must automatically trigger a color transition.

Bit	Description
0	Automatic activation.
1 - 7	reserved.

The DTR (Data Transfer Register) of all DALI control gears in the DALI line is changed by the function block.

The function block supports the following color representations:

- xy coordinates
- Color temperature Tc
- Primary (color) N
- RGBWAF

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart                    : BOOL;
nAddr                     : BYTE;
eAddrType                 : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority          : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nGearFeaturesStatus       : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nGearFeaturesStatus: Value that is written into the GEAR FEATURES/STATUS variable.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

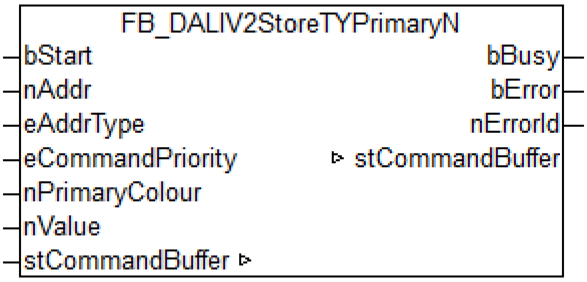
```
stCommandBuffer        : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.258 FB_DALIV2StoreTYPrimaryN



Saves the value in the variable TY PRIMARY N [▸ 374] of the DALI control gear.

The value is expressed in units of 0.5 lm, which results in a possible range of TYmin = 0 lm to TYmax = 32767 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 DTR (Data Transfer Register), DTR1 and DTR2 of all DALI control gears in the DALI line are changed by the function block.

The function block supports the following color representations:

- Primary (color) N

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nPrimaryColour     : BYTE;
nValue         : UINT;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nPrimaryColour: Primary colour (0 - 5).

nValue: The value that is written into the TY PRIMARY N variable.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

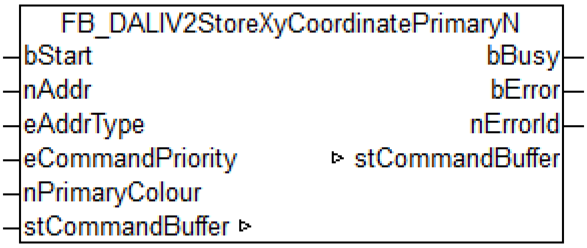
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.259 FB_DALIV2StoreXyCoordinatePrimaryN



Copies the value from the variables TEMPORARY x-COORDINATE [▶ 374] and TEMPORARY y-COORDINATE [▶ 374] to the variables x-COORDINATE PRIMARY N [▶ 374] and y-COORDINATE PRIMARY N [▶ 374].

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 DTR2 (Data Transfer Register) of all DALI control gears in the DALI line is changed by the function block.

The function block supports the following color representations:

- Primary (color) N

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nPrimaryColour      : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

nPrimaryColour: Primary colour (0 - 5).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

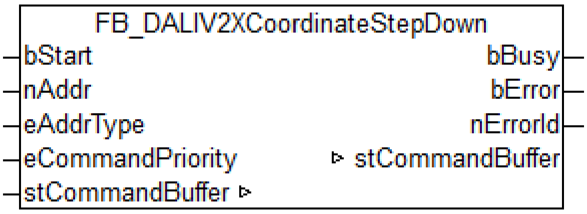
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.260 FB_DALIV2XCoordinateStepDown



The variable x-COORDINATE is reduced 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 COLOUR STATUS [▸ 378] (xy-coordinate color point lies outside the valid range). The command is executed only if bit 4 of COLOUR STATUS [▸ 378] (color representation xy-coordinate active) is set.

The function block supports the following color representations:

- xy coordinates

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

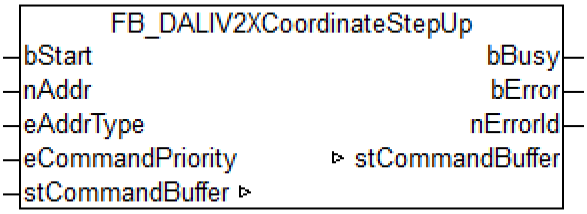
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.261 FB_DALIV2XCoordinateStepUp



The variable x-COORDINATE is increased 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 COLOUR STATUS [▸ 378] (xy-coordinate color point lies outside the valid range). The command is executed only if bit 4 of COLOUR STATUS [▸ 378] (color representation xy-coordinate active) is set.

The function block supports the following color representations:

- xy coordinates

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

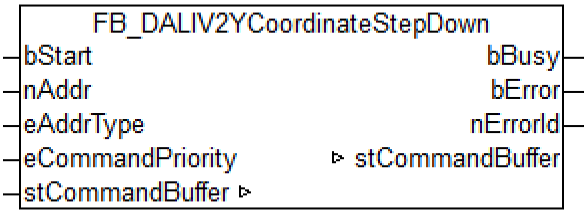
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.262 FB_DALIV2YCoordinateStepDown



The variable y-COORDINATE is reduced 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 COLOUR STATUS [▸ 378] (xy-coordinate color point lies outside the valid range). The command is executed only if bit 4 of COLOUR STATUS [▸ 378] (color representation xy-coordinate active) is set.

The function block supports the following color representations:

- xy coordinates

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

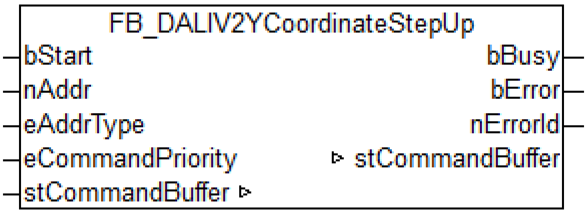
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.263 FB_DALIV2YCoordinateStepUp



The variable y-COORDINATE is increased 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 COLOUR STATUS [▸ 378] (xy-coordinate color point lies outside the valid range). The command is executed only if bit 4 of COLOUR STATUS [▸ 378] (color representation xy-coordinate active) is set.

The function block supports the following color representations:

- xy coordinates

This command belongs to the application extended commands for lamps for the color/color temperature control with DALI interface. These only work if they are preceded by the *Enable Device Type 8* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▸ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type 8* command is internally placed automatically before all application extended commands for lamps for color/color temperature control with DALI interface.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

eCommandPriority: The priority [▸ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

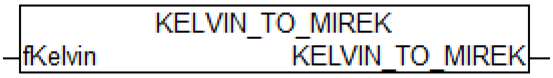
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from Build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.264 KELVIN_TO_MIREK



Conversion of the color temperature from Kelvin to Mirek.

Mirek is the unit that is used with most DALI commands. The return value is limited and lies within the range from 0 to 65535 (see table).

Mirek = 1,000,000 / (color temperature in Kelvin).

Kelvin	Mirek
0	65535
15	65535
16	62500
1000	1000
10000	100
1000000	1
1000001	0

VAR_INPUT

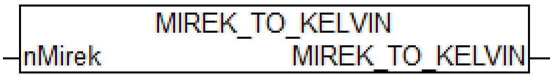
```
fKelvin          : LREAL;
```

fKelvin: Color temperature in Kelvin.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.265 MIREK_TO_KELVIN



Conversion of the color temperature from Mirek to Kelvin.

Mirek is the unit that is used with most DALI commands. The return value is limited and lies within the range from approx. 15.259 to 1000001 (see table).

Mirek = 1,000,000 / (color temperature in Kelvin).

Mirek	Kelvin
0	1000001
1	1000000
100	10000
1000	1000
10000	100

Mirek	Kelvin
65534	15.259
65535	15.259

VAR_INPUT

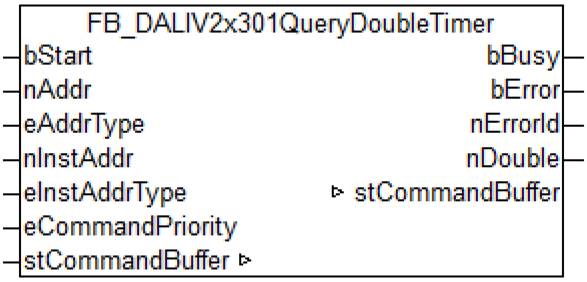
```
nMirek        : UINT;
```

nMirek: Color temperature in Mirek.

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2244	PC/CX, BX or BC	TcDALIV2 library from V2.10.0

5.1.266 FB_DALIV2x301QueryDoubleTimer



Queries the value of the DOUBLE TIMER.

The unit is 20 ms. The maximum permissible value is 2000 ms. The value 0 disables the timer. The minimum allowed value can be queried with FB_DALIV2x301QueryDoubleTimerMin() [▶ 321].

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType  := eDALIV2AddrTypeShort;
nInstAddr          : BYTE  := 0;
eInstAddrType      : E_DALIV2InstAddrType  := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority  := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nDouble       : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nDouble: Value of the DOUBLE TIMER [20 ms].

VAR_IN_OUT

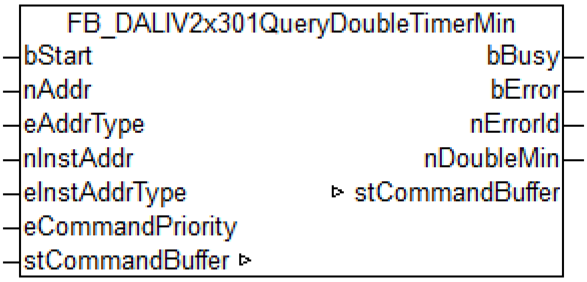
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.267 FB_DALIV2x301QueryDoubleTimerMin



Queries the minimum value of the DOUBLE TIMER.

The unit is 20 ms.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nDoubleMin    : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nDoubleMin: Minimum value of the DOUBLE TIMER [20 ms].

VAR_IN_OUT

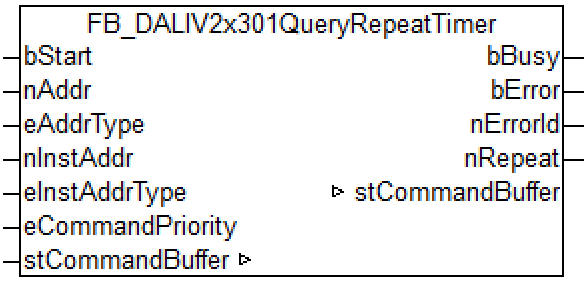
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.268 FB_DALIV2x301QueryRepeatTimer



Queries the value of the REPEAT TIMER.

The unit is 20 ms. The permissible value range is 100 ms to 2000 ms, i.e. from 5 to 100.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nRepeat        : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nRepeat: Value of the REPEAT TIMER [20 ms].

VAR_IN_OUT

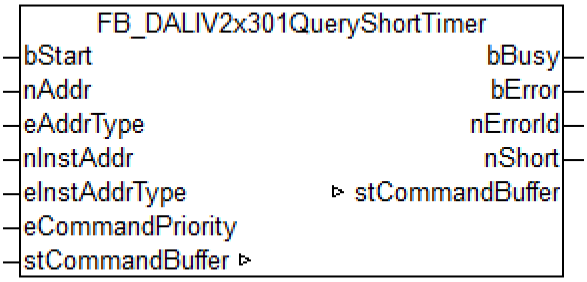
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.269 FB_DALIV2x301QueryShortTimer



Queries the value of the SHORT TIMER.

The unit is 20 ms. The maximum permissible value is 5100 ms. The minimum allowed value can be queried with FB_DALIV2x301QueryShortTimerMin() [▶ 324].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_E_DALIV2InstAddrType:= eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nShort          : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nShort: Value of the SHORT TIMER [20 ms].

VAR_IN_OUT

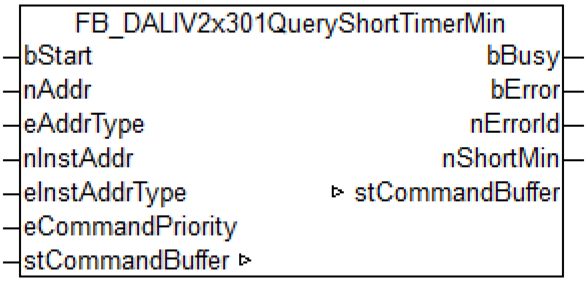
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.270 FB_DALIV2x301QueryShortTimerMin



Queries the minimum value of the SHORT TIMER.

The unit is 20 ms.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nShortMin      : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nShortMin: Minimum value of the SHORT TIMER [20 ms].

VAR_IN_OUT

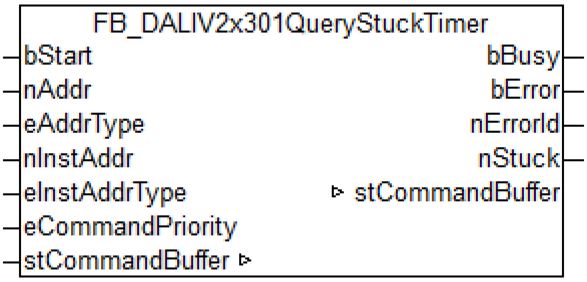
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.271 FB_DALIV2x301QueryStuckTimer



Queries the value of the STUCK TIMER.

The unit is 1 s. The permissible value range is 5 s to 255 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nStuck          : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nStuck: Value of the STUCK TIMER [s].

VAR_IN_OUT

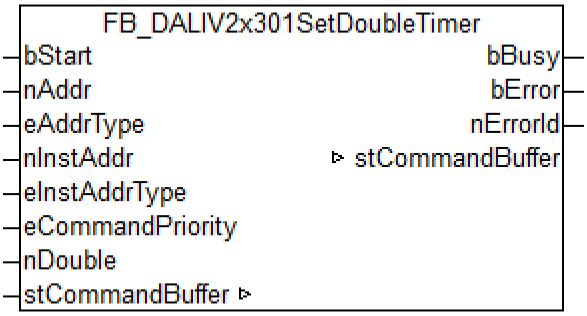
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.272 FB_DALIV2x301SetDoubleTimer



Sets the value of the DOUBLE TIMER.

The unit is 20 ms. The maximum permissible value is 2000 ms. The value 0 disables the timer. The minimum allowed value can be queried with FB_DALIV2x301QueryDoubleTimerMin() [▶ 321].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDouble         : BYTE := 0;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address, group address or broadcast.

nInstAddr: Address of the instance within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority (high, medium or low) with which the command is processed by the library.

nDouble: Value of the DOUBLE TIMER [20 ms].

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

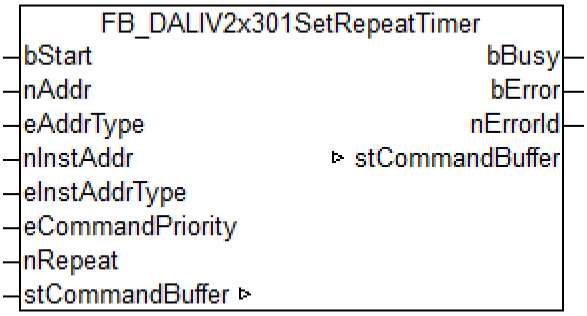
Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

Also see about this

- E_DALIV2AddrType [▸ 380]
- E_DALIV2InstAddrType [▸ 382]
- E_DALIV2CommandPriority [▸ 381]

5.1.273 FB_DALIV2x301SetRepeatTimer



Sets the value of the REPEAT TIMER.

The unit is 20 ms. The permissible value range is 100 ms to 2000 ms, i.e. from 5 to 100.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nRepeat             : BYTE := 8;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nRepeat: Value of the REPEAT TIMER [20 ms].

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

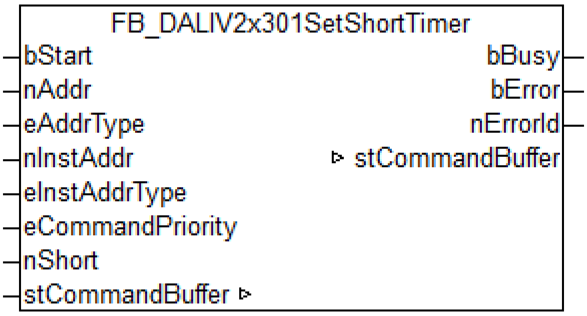
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.274 FB_DALIV2x301SetShortTimer



Sets the value of the SHORT TIMER.

The unit is 20 ms. The maximum permissible value is 5100 ms. The minimum allowed value can be queried with FB_DALIV2x301QueryShortTimerMin() [▸ 324].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nShort              : BYTE := 25;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nShort: Value of the SHORT TIMER [20 ms].

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

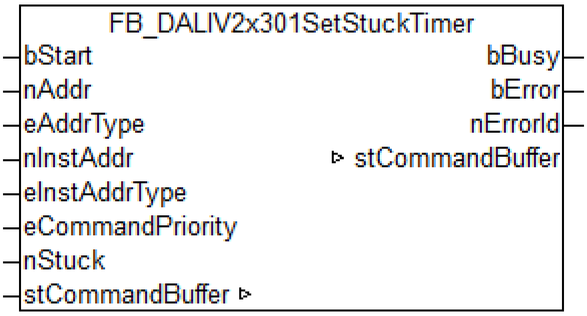
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.275 FB_DALIV2x301SetStuckTimer



Sets the value of the STUCK TIMER.

The unit is 1 s. The permissible value range is 5 s to 255 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nStuck              : BYTE := 20;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nStuck: Value of the STUCK TIMER [s].

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

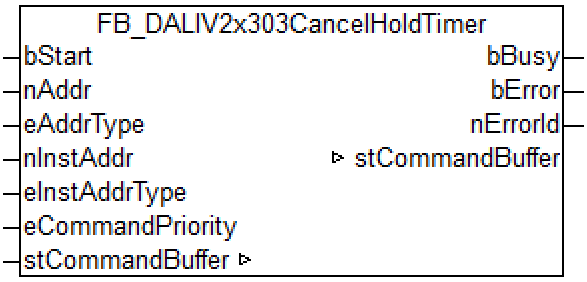
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.276 FB_DALIV2x303CancelHoldTimer



Terminates the HOLD TIMER prematurely.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

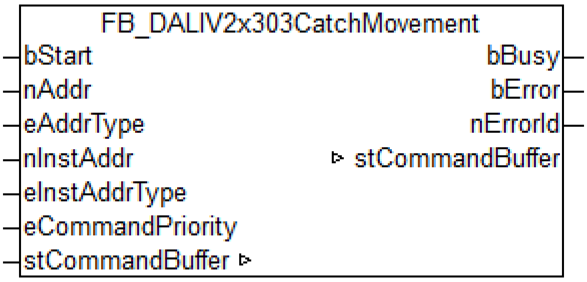
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.277 FB_DALIV2x303CatchMovement



After calling this command, an event is only sent once if a movement is detected.

For this function the event filter must be configured accordingly.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

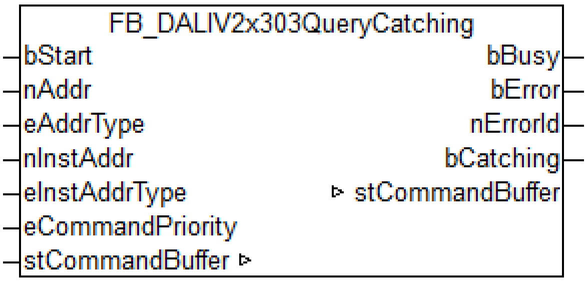
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.278 FB_DALIV2x303QueryCatching



Queries whether the system is waiting for the detection of movement. This function can be activated with FB_DALIV2x303CatchMovement() [▶ 332].

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
bCatching       : BOOL;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

bCatching: Catching active.

VAR_IN_OUT

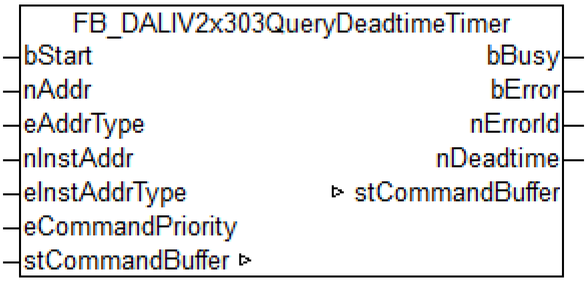
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.279 FB_DALIV2x303QueryDeadtimeTimer



Queries the value of the DEADTIME TIMER.

The unit is 50 ms. The permissible value range is 0 s to 12.75 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nDeadtime      : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nDeadtime: Value of the DEADTIME TIMER [50 ms].

VAR_IN_OUT

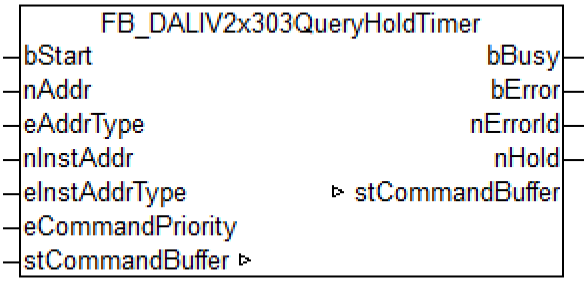
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.280 FB_DALIV2x303QueryHoldTimer



Queries the value of the HOLD TIMER.

The unit is 10 s. The permissible value range is 1 s to 42.3 min.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
```

```
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nHold          : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nHold: Value of the HOLD TIMER [10 s].

VAR_IN_OUT

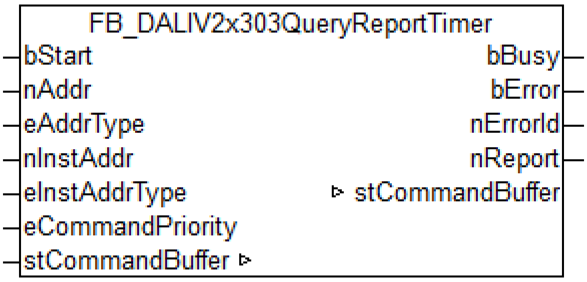
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.281 FB_DALIV2x303QueryReportTimer



Queries the value of the REPORT TIMER.

The unit is 1 s. The permissible value range is 1 s to 4 min 15 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr            : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
nReport         : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nReport: Value of the REPORT TIMER [s].

VAR_IN_OUT

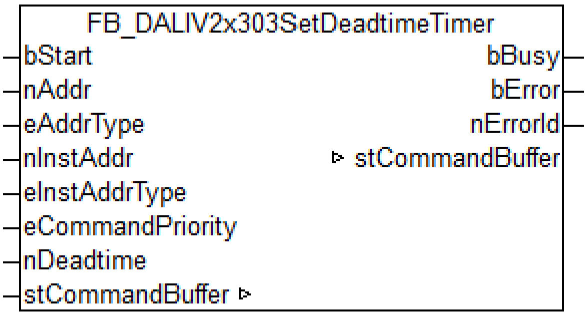
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.282 FB_DALIV2x303SetDeadtimeTimer



Sets the value of the DEADTIME TIMER.

The unit is 50 ms. The permissible value range is 0 s to 12.75 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDeadtime       : BYTE := 2;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nDeadtime: Value of the DEADTIME TIMER [50 ms].

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

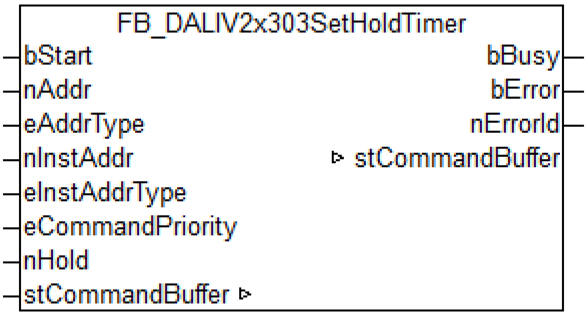
```
stCommandBuffer     : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.283 FB_DALIV2x303SetHoldTimer



Sets the value of the HOLD TIMER.
If the value 0 is passed, HOLD TIME is set to 1 sThe unit is 10 s. The permissible value range is 1 s to 42,3 min.

VAR_INPUT

```
bStart           : BOOL;
nAddr            : BYTE;
eAddrType        : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr        : BYTE := 0;
eInstAddrType    : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nHold            : BYTE := 90;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nHold: Value of the HOLD TIMER [10 s]. The value 0 corresponds to 1 s.

VAR_OUTPUT

```
bBusy        : BOOL;
bError       : BOOL;
nErrorId     : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

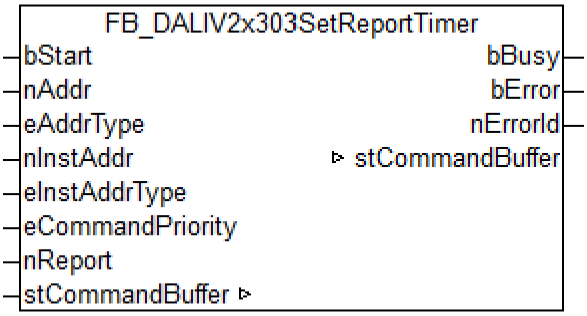
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.284 FB_DALIV2x303SetReportTimer



Sets the value of the REPORT TIMER.

The unit is 1 s. The permissible value range is 1 s to 4 min 15 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nReport             : BYTE := 20;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nReport: Value of the REPORT TIMER [s].

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

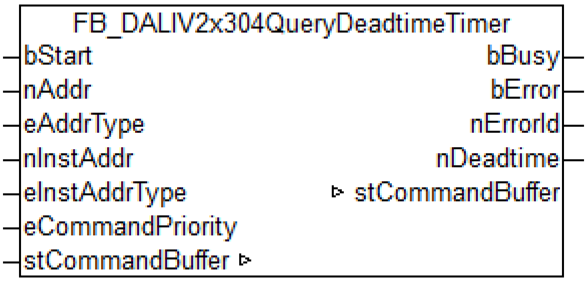
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.285 FB_DALIV2x304QueryDeadtimeTimer



Queries the value of the DEADTIME TIMER.

The unit is 50 ms. The permissible value range is 0 s to 12.75 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nDeadtime     : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nDeadtime: Value of the DEADTIME TIMER [50 ms].

VAR_IN_OUT

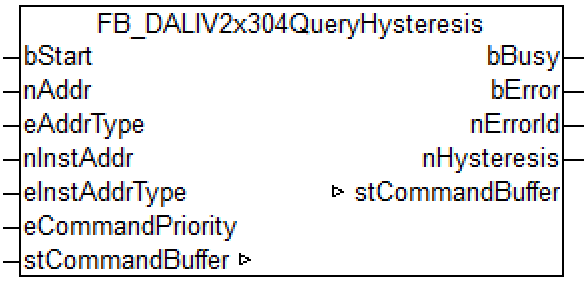
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.286 FB_DALIV2x304QueryHysteresis



Queries the hysteresis value.

The hysteresis is given in % and is within the range 0% to 25%.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy         : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nHysteresis   : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nHysteresis: Hysteresis in %.

VAR_IN_OUT

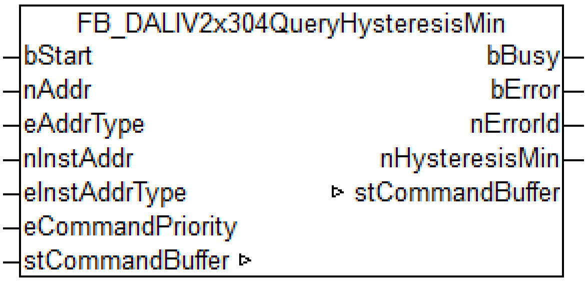
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.287 FB_DALIV2x304QueryHysteresisMin



Queries the value for the minimum possible hysteresis.

VAR_INPUT

```
bStart            : BOOL;
nAddr             : BYTE;
eAddrType         : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr         : BYTE := 0;
eInstAddrType     : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority  : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nHysteresisMin: BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

nHysteresisMin: Minimum possible hysteresis.

VAR_IN_OUT

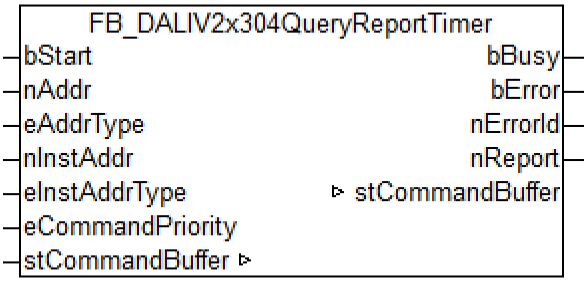
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.288 FB_DALIV2x304QueryReportTimer



Queries the value of the REPORT TIMER.

The unit is 1 s. The permissible value range is 1 s to 4 min 15 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
nReport        : BYTE;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

nReport: Value of the REPORT TIMER [s].

VAR_IN_OUT

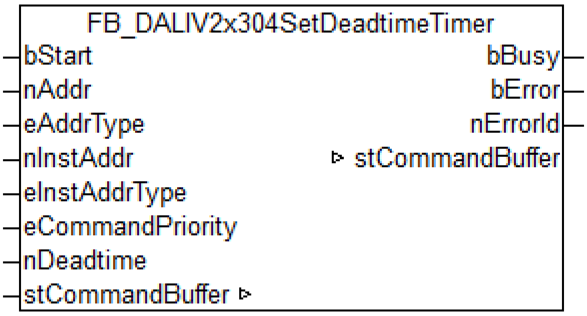
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.289 FB_DALIV2x304SetDeadtimeTimer



Sets the value of the DEADTIME TIMER.

The unit is 50 ms. The permissible value range is 0 s to 12.75 s.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDeadtime          : BYTE := 30;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nDeadtime: Value of the DEADTIME TIMER [50 ms].

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

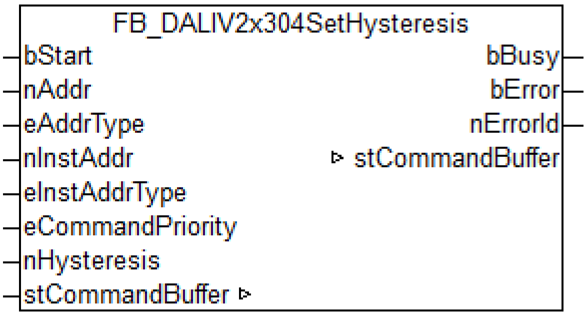
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.290 FB_DALIV2x304SetHysteresis



Sets the hysteresis value.

The hysteresis is given in % and is within the range 0% to 25%.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nHysteresis         : BYTE := 5;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nHysteresis: Hysteresis in %.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

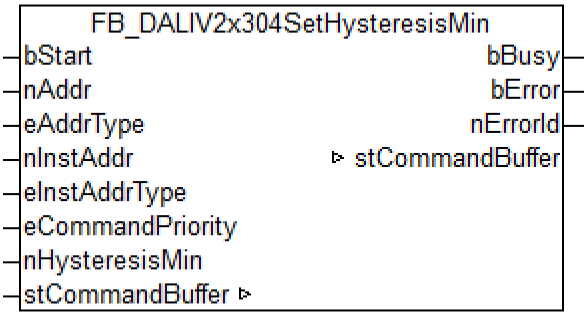
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.291 FB_DALIV2x304SetHysteresisMin



Sets the value for the minimum possible hysteresis.

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.

VAR_INPUT

```
bStart             : BOOL;
nAddr              : BYTE;
eAddrType          : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr          : BYTE := 0;
eInstAddrType      : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority   : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nHysteresisMin     : BYTE := 0;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▸ 380], group address or broadcast.

nInstAddr: Address of the instance [▸ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▸ 381] (high, medium or low) with which the command is processed by the library.

nHysteresisMin: Minimum possible hysteresis.

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▸ 385].

VAR_IN_OUT

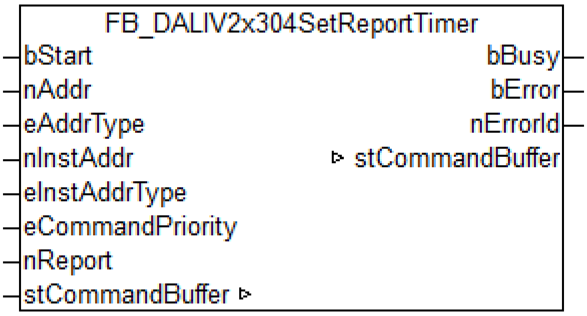
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.292 FB_DALIV2x304SetReportTimer



Sets the value of the REPORT TIMER.

The unit is 1 s. The permissible value range is 1 s to 4 min 15 s.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
nInstAddr           : BYTE := 0;
eInstAddrType       : E_DALIV2InstAddrType := eDALIV2InstAddrTypeNumber;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nReport             : BYTE := 30;
```

bStart: The function block is activated by a positive edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

nInstAddr: Address of the instance [▶ 382] within the DALI control unit.

eInstAddrType: Defines the access mode to the desired instance within the DALI control unit.

eCommandPriority: Priority [▶ 381] (high, medium or low) with which the command is processed by the library.

nReport: Value of the REPORT TIMER [s].

VAR_OUTPUT

```
bBusy           : BOOL;
bError          : BOOL;
nErrorId        : UDINT;
```

bBusy: When the function block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of a command at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of a command at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

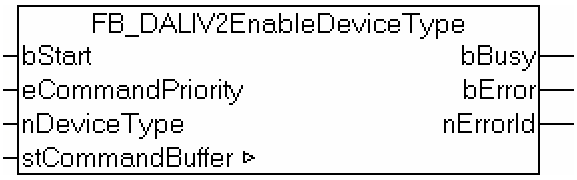
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.293 FB_DALIV2EnableDeviceType



This command must be sent every time prior to an application-specific extension command. Only the ballasts that belong to the corresponding device type (DEVICE TYPE [▶ 366]) react to this. The command does not have to be used for device type 0. The following device types are defined according to the IEC 62386 standard:

Value	Description
0	Standard device.
1	Device for emergency lighting [▶ 32].
2	Device for discharge lamps [▶ 36].
3	Device for low-voltage halogen lamps.
4	Device for dimming bulbs.
5	Device for converting digital signals into DC signals.
6	Device for light emitting diodes (LEDs) [▶ 36].
7	Switching function.
8	Device for colour/colour temperature control [▶ 34].

VAR_INPUT

```
bStart              : BOOL;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDeviceType         : BYTE;
```

bStart: The block is activated by a rising edge at this input.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nDeviceType: Identifier for the device type (see table above).

VAR_OUTPUT

```
bBusy          : BOOL;
bError         : BOOL;
nErrorId       : UDINT;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

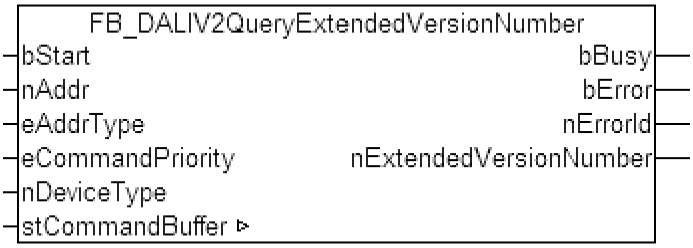
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▶ 385].

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

5.1.294 FB_DALIV2QueryExtendedVersionNumber



The variable EXTENDED VERSION NUMBER [▶ 371] is read from the control gear.

This command belongs to the application extended commands. These only work if they are preceded by the *Enable Device Type* command, which can be sent with the function block FB_DALIV2EnableDeviceType() [▶ 350]. From version 2.6.0 of the libraries *TcDALIV2* and *TcDALIV2AppExtCmds*, however, the *Enable Device Type* command is internally placed automatically before them. The *Device Type* to be activated can be set at the nDeviceType [▶ 350] input.

VAR_INPUT

```
bStart              : BOOL;
nAddr               : BYTE;
eAddrType           : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority    : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
nDeviceType         : BYTE;
```

bStart: The block is activated by a rising edge at this input.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [▶ 380], group address or broadcast.

eCommandPriority: The priority [▶ 381] (high, middle, low) this command has when executed by the library.

nDeviceType: Identifier for the device type (see table at FB_DALIV2EnableDeviceType() [▶ 350]).

VAR_OUTPUT

```
bBusy                     : BOOL;
bError                    : BOOL;
nErrorId                  : UDINT;
nExtendedVersionNumber    : BYTE;
```

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [▸ 385].

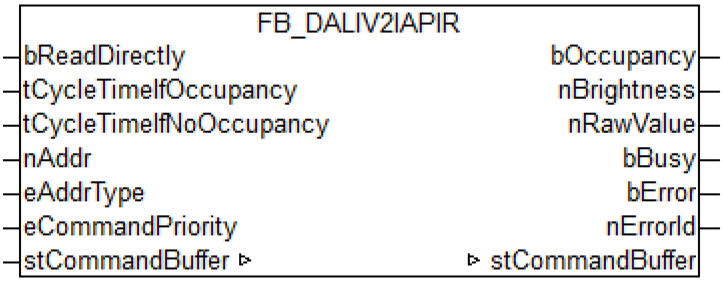
nExtendedVersionNumber: extended version number of the ballast.

VAR_IN_OUT

```
stCommandBuffer      : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▸ 93] (KL6811) or FB_KL6821Communication() [▸ 101] (KL6821) block.

5.1.295 FB_DALIV2IAPIR



This function cyclically reads the status of an IA PIR-sensor and scales the measured light intensity and the recognized presence of a person from the received value.

There are two different cycle times allowed. One cycle time which is used if no presence of a person is recognized and another, which is used if the presence of a person is recognized.

Thus it is possible to minimize the accesses to the DALI-Bus. If the presence of a person is recognized, a slow cycle time (f. e. 20s) can be used, because the light regulation and the switching off of the lighting is not time-critical. If no presence of a person is recognized, a smaller cycle time (f. e. 2s) should be used. Thus the switching on of the lighting can be done with a minimized reaction time.

You can find more informations and a description of the electrical and physical properties in the product description of the IA PIR-sensor.

VAR_INPUT

```
bReadDirectly             : BOOL := FALSE;
tCycleTimeIfOccupancy     : TIME := t#20s;
tCycleTimeIfNoOccupancy   : TIME := t#2s;
nAddr                     : BYTE := 0;
eAddrType                 : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority          : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bReadDirectly: with a positive edge at this input the module is activated.

tCycleTimeIfOccupancy: cycle time if a presence is recognized.

tCycleTimeIfNoOccupancy: cycle time if no presence is recognized.

nAddr: adress of a member or a group.

eAddrType: short adress [▸ 380], group adress or broadcast.

eCommandPriority: priority [▸ 381] (high, medium or low), with which the command is executed by the library.

VAR_OUTPUT

```
bOccupancy       : BOOL;
nBrightness      : INT;
nRawValue        : BYTE;
```

```
bBusy              : BOOL;
bError             : BOOL;
nErrorId           : UDINT;
```

bOccupancy: this output is set, if the sensor recognizes presence.

nBrightness: measured light intensity in Lux.

nRawValue: the value the sonsor read before conversion.

bBusy: this output is set, when the module is activated and stays active until the command is executed.

bError: this output is set TRUE, if an error occurs while a command is executed.

nErrorId: contains the command-specific error code of the command executed last. Is set back to 0, if a new command is executed. See Error codes [▶ 385].

VAR_IN_OUT

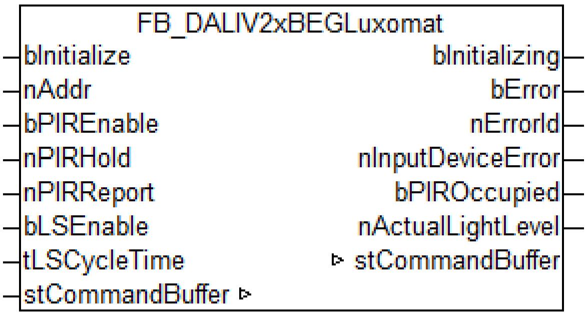
```
stCommandBuffer        : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [▶ 93] (KL6811) or FB_KL6821Communication() [▶ 101] (KL6821) block.

Requirements

Development environment	Target system type	PLC library to include
TwinCAT 2.11 R3/x64 Build 2234 or higher	PC/CX, BX or BC	TcDALIV2-Library V2.9.0 or higher

5.1.296 FB_DALIV2xBEGLuxomat



This function block is used as an example and is therefore not included in the library. The function block can be downloaded as an export file and imported into the desired project. Adjustments can be made as required.

This function block evaluates the measured brightness and presence of the B.E.G. Luxomat DALI control device. This function block can also be used to initialize the DALI control units.

The DALI control unit occupies a short address and contains two instances. Instance 0 is the movement sensor and complies with the IEC 62386-303 standard. The light sensor is stored in Instance 1 and complies with the IEC 62386-304 standard.

A detected movement is sent as an event while the brightness is read out cyclically. Each instance can be decoded individually as required.

Download the TwinCAT 2 PLC Control export file: https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019296651/.zip

VAR_INPUT

```
bInitialize       : BOOL := FALSE;
nAddr             : BYTE;
(* Occupancy Sensor Parameters *)
bPIREnable        : BOOL := TRUE;
nPIRHold          : BYTE := 1;    (* 10 s *)
nPIRReport        : BYTE := 30;   (* 30 s *)
(* Light Sensor Parameters *)
bLSEnable         : BOOL := TRUE;
tLSCycleTime      : TIME := t#1m;
```

bInitialize: The DALI controller is initialized by a positive edge at this input. The parameters are written to the DALI control unit, which must be accessible via the short address *nAddr*. The output *bInitializing* is TRUE during initialization.

nAddr: Short address of the DALI control unit.

bPIREnable: Parameter: If this input is TRUE, the motion sensor (instance 0) is enabled. The occupancy sensor is disabled by FALSE.

nPIRHold: Parameter: Sets the *Hold Timer* value. If no movement is detected, the status of the motion sensor is only changed after the *Hold Timer* has expired. The unit is 10 s. This means that times of up to 42 min 20 s (value 254) are possible. The value 0 corresponds to 1 s while the value 255 is ignored.

nPIRReport: Parameter: Sets the value for the *Report Timer*. The motion sensor status is retransmitted after the *Report Timer* has expired, even if the status has not changed. The unit is 1 s. The value 0 disables the *Report Timer*. This means that times of up to 4 min 15 s (value 255) are possible.

bLSEnable: Parameter: If this input is TRUE, the brightness sensor (instance 1) is enabled. The brightness sensor is disabled by FALSE.

tLSCycleTime: Cycle time in which the current actual value of the brightness sensor is read out.

Further details on the parameters can be found in the IEC 62386 standard and the manufacturer's documentation.

VAR_OUTPUT

```
bInitializing         : BOOL;
bError                : BOOL;
nErrorId              : UDINT;
nInputDeviceError     : BYTE;
(* Occupancy Sensor *)
bPIROccupied          : BOOL;
(* Light Sensor *)
nActualLightLevel     : UINT;
```

bInitializing: This output is TRUE during initialization.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*.

nErrorId: Contains the command-specific error code of the most recently executed command. See Error codes [▶ 385].

nInputDeviceError: The status of the DALI control unit (INPUT DEVICE ERROR) is queried before the initialization. 0 means no error. The individual error numbers are manufacturer-specific.

bPIROccupied: This output indicates the status of the motion sensor.

nActualLightLevel: This output indicates the status of the brightness sensor.

VAR_IN_OUT

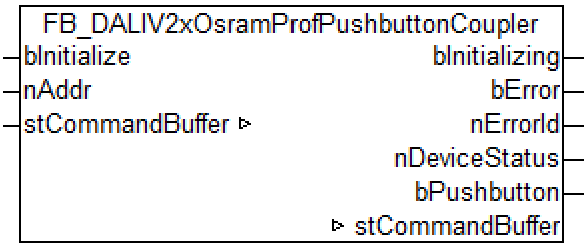
```
stCommandBuffer   : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.297 FB_DALIV2xOsramProfPushbuttonCoupler



This function block is used as an example and is therefore not included in the library. The function block can be downloaded as an export file and imported into the desired project. Adjustments can be made as required.

This function block evaluates the status of the digital input of the Osram DALI Professional Pushbutton Coupler. This function block can also be used to initialize the DALI control units.

The DALI control unit has a short address. The device does not fully comply with the IEC 62386 standard and uses a company-specific protocol instead.

Download the TwinCAT 2 PLC Control export file: https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019298059/.zip

VAR_INPUT

```
bInitialize        : BOOL := FALSE;
nAddr              : BYTE;
```

bInitialize: The DALI controller is initialized by a positive edge at this input. The parameters are written to the DALI control unit which must be accessible via the short addresses *nPIRAddr* and *nLSAddr*. The output *bInitializing* is TRUE during initialization.

nAddr: Short address of the DALI control unit.

Further details about the parameters can be found in the manufacturer's documentation.

VAR_OUTPUT

```
bInitializing : BOOL;
bError        : BOOL;
nErrorId      : UDINT;
nDeviceStatus : BYTE;
bPushbutton   : BOOL;
```

bInitializing: This output is TRUE during initialization.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*.

nErrorId: Contains the command-specific error code of the most recently executed command. See Error codes [▶ 385].

nDeviceStatus: Before the initialization, the status of the DALI control unit is queried and output to this output.

bPushbutton: This output indicates the actual value of the digital input.

VAR_IN_OUT

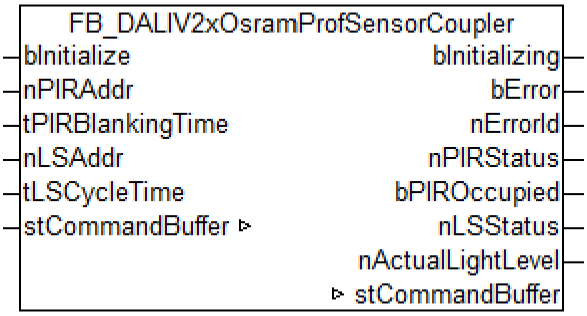
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.298 FB_DALIV2xOsramProfSensorCoupler



This function block is used as an example and is therefore not included in the library. The function block can be downloaded as an export file and imported into the desired project. Adjustments can be made as required.

This function block evaluates the measured brightness and presence of the Osram DALI Professional Sensor Coupler. This function block can also be used to initialize the DALI control units.

The movement sensor and the light sensor have separate short addresses. The DALI control unit thus occupies two short addresses. The device does not fully comply with the IEC 62386 standard and uses a company-specific protocol instead.

A detected movement is sent as an event while the brightness is read out cyclically.

Download the TwinCAT 2 PLC Control export file: https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019299467/.zip

VAR_INPUT

```
bInitialize          : BOOL := FALSE;
(* Occupancy Sensor Parameters *)
nPIRAddr             : BYTE;
tPIRBlankingTime     : TIME := T#1M;
(* Light Sensor Parameters *)
nLSAddr              : BYTE;
tLSCycleTime         : TIME := T#1M;
```

bInitialize: The DALI controller is initialized by a positive edge at this input. The parameters are written to the DALI control unit which must be accessible via the short addresses *nPIRAddr* and *nLSAddr*. The output *bInitializing* is TRUE during initialization.

nPIRAddr: Short address of the DALI control unit for the motion sensor.

tPIRBlankingTime: Parameter: Once the status of the motion sensor was sent, no further changes are sent for this time. The unit is 1 s. This means that times of up to 4 min 15 s (value 255) are possible.

nLSAddr: Short address of the DALI control unit for the brightness sensor.

tLSCycleTime: Cycle time in which the current actual value of the brightness sensor is read out.

Further details about the parameters can be found in the manufacturer's documentation.

VAR_OUTPUT

```
bInitializing          : BOOL;
bError                 : BOOL;
nErrorId               : UDINT;
(* Occupancy Sensor *)
nPIRStatus             : BYTE;
bPIROccupied           : BOOL;
(* Light Level *)
nLSStatus              : BYTE;
nActualLightLevel      : WORD;
```

bInitializing: This output is TRUE during initialization.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*.

nErrorId: Contains the command-specific error code of the most recently executed command. See Error codes [▶ 385].

nPIRStatus: Before the initialization, the status of the DALI control unit for the motion sensor is queried and output at this output.

bPIROccupied: This output indicates the actual value of the motion sensor.

nLSStatus: Before the initialization, the status of the DALI controller for the brightness sensor is queried and output at this output.

nActualLightLevel: This output indicates the actual value of the brightness sensor.

VAR_IN_OUT

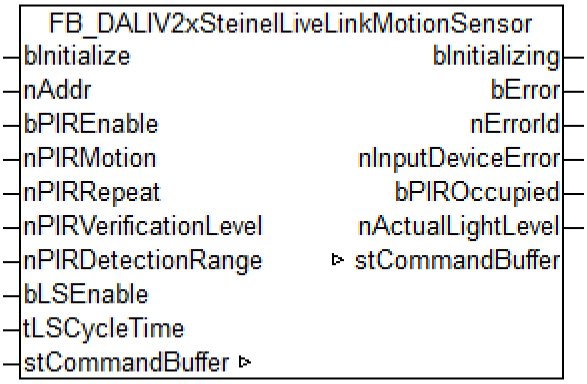
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.299 FB_DALIV2xSteinelLiveLinkMotionSensor



This function block is used as an example and is therefore not included in the library. The function block can be downloaded as an export file and imported into the desired project. Adjustments can be made as required.

This function block evaluates the measured brightness and presence of the Steinel LiveLink DALI control unit. This function block can also be used to initialize the DALI control units.

The DALI control unit occupies a short address and contains 2 instances. Instance 1 contains the movement sensor, instance 0 the light sensor. The device does not fully comply with the IEC 62386 standard and uses a company-specific protocol instead.

A detected movement is sent as an event while the brightness is read out cyclically. Each instance can be decoded individually as required.

Download TwinCAT 2 PLC Control export file: https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019300875/.zip

VAR_INPUT

```
bInitialize             : BOOL := FALSE;
nAddr                   : BYTE;
(* Occupancy Sensor Parameters *)
bPIREnable              : BOOL := TRUE;
nPIRMotion              : BYTE := 4;     (* 5 s + (nPIRMotion * 5 s) *)
nPIRRepeat              : BYTE := 6;     (* 5 s + (nPIRRepeat * 5 s) *)
nPIRVerificationLevel   : BYTE := 1;     (* standard *)
nPIRDetectionRange      : BYTE := 255;   (* (only for HF sensors) 100 % *)
(* Light Sensor Parameters *)
bLSEnable               : BOOL := TRUE;
tLSCycleTime            : TIME := T#1M;
```

bInitialize: The DALI controller is initialized by a positive edge at this input. The parameters are written to the DALI control unit, which must be accessible via the short address *nAddr*. The output *bInitializing* is TRUE during initialization.

nAddr: Short address of the DALI control unit.

bPIREnable: Parameter: If this input is TRUE, the motion sensor (instance 1) is enabled. The occupancy sensor is disabled by FALSE.

nPIRMotion: Parameter: Sets the value for the *Motion Timer*. If no movement is detected, the status of the motion sensor is not changed until after the *Motion Timer* has expired. The time is calculated based on the formula 5 s + nPIRMotion * 5 s. This means that times of 5 s to 21 min 20 s are possible.

nPIRRepeat: Parameter: Sets the value for the *Repeat Timer*. The motion sensor status is retransmitted after the *Report Timer* has expired, even if the status has not changed. The time is calculated based on the formula 5 s + nPIRRepeat * 5 s. This means that times of 5 s to 21 min 20 s are possible.

nPIRVerificationLevel: Parameter: The detection sensitivity can be adjusted (0: motion detection disabled, 1: very sensitive to 15: very insensitive).

nPIRDetectionRange: Parameter: If an HF sensor is used, the detection range can be set (0: 0% to 255: 100%).

bLSEnable: Parameter: If this input is TRUE, the brightness sensor (instance 0) is enabled. The brightness sensor is disabled by FALSE.

tLSCycleTime: Cycle time in which the current actual value of the brightness sensor is read out.

Further details on the parameters can be found in the IEC 62386 standard and the manufacturer's documentation.

VAR_OUTPUT

```
bInitializing       : BOOL;
bError              : BOOL;
nErrorId            : UDINT;
nInputDeviceError   : BYTE;
(* Occupancy Sensor *)
bPIROccupied        : BOOL;
(* Light Sensor *)
nActualLightLevel   : UINT;
```

bInitializing: This output is TRUE during initialization.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*.

nErrorId: Contains the command-specific error code of the most recently executed command. See Error codes [▸ 385].

nInputDeviceError: The status of the DALI control unit (INPUT DEVICE ERROR) is queried before the initialization. 0 means no error. The individual error numbers are manufacturer-specific.

bPIROccupied: This output indicates the status of the motion sensor.

nActualLightLevel: This output indicates the status of the brightness sensor.

VAR_IN_OUT

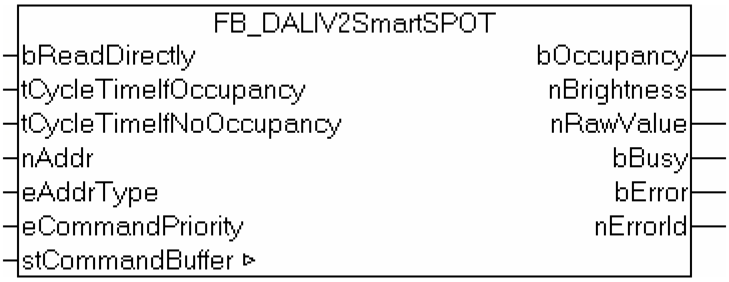
```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▸ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.1.300 FB_DALIV2SmartSPOT



This function block cyclically reads the status of a smartSPOT sensor or MSensor 02 and scales the measured brightness and detected presence based on the received value.

If the function block with MSensor 02 is to be used, it must be in slave mode.

Two different cycle times can be specified. One cycle time that is used if no presence is detected and one that is used in case of active presence. In this way access to the DALI bus can be minimized. If presence is detected a slower cycle time (e.g. 20 s) can usually be selected, because lighting control and switching off the lighting is not time-critical. If there is no presence, then a shorter cycle time (e.g. 2 s) should be selected. As a result, the lighting is switched on with the shortest possible reaction time when the room is entered.

The measured brightness of the sensor depends on the reflection of the furniture in the room. The measured value is only 20 to 40% of the actual brightness of the respective surface. Example: 500 lux on the table with a reflectivity of 30 % results in a brightness value of 150 lux. The measuring range of the sensor is designed so that the usual brightness at the workplace can be controlled from 200 lux to 1000 lux.

Further information and a description of the electrical and physical properties can be found in the product description of the smartSPOT sensor or the MSensor 02.

VAR_INPUT

```
bReadDirectly              : BOOL := FALSE;
tCycleTimeIfOccupancy      : TIME := t#20s;
tCycleTimeIfNoOccupancy    : TIME := t#2s;
nAddr                      : BYTE := 0;
eAddrType                  : E_DALIV2AddrType := eDALIV2AddrTypeShort;
eCommandPriority           : E_DALIV2CommandPriority := eDALIV2CommandPriorityMiddle;
```

bReadDirectly: The actual values are read immediately by a rising edge at this input.

tCycleTimeIfOccupancy: Cycle time for the read of the actual values during active presence.

tCycleTimeIfNoOccupancy: Cycle time for the read of the actual values during inactive presence.

nAddr: The address of a participating device or of a group.

eAddrType: Short address [⏵ 380], group address or broadcast.

eCommandPriority: The priority [⏵ 381] (high, middle, low) this command has when executed by the library.

VAR_OUTPUT

```
bOccupancy          : BOOL;
nBrightness         : INT;
nRawValue           : BYTE;
bBusy               : BOOL;
bError              : BOOL;
nErrorId            : UDINT;
```

bOccupancy: The output is set if the sensor detects presence.

nBrightness: Measured brightness in lux.

nRawValue: Raw value before conversion.

bBusy: When the block is activated the output is set, and it remains active until execution of the command has been completed.

bError: This output is switched to TRUE if an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*. Is reset to FALSE by the execution of an instruction at the inputs.

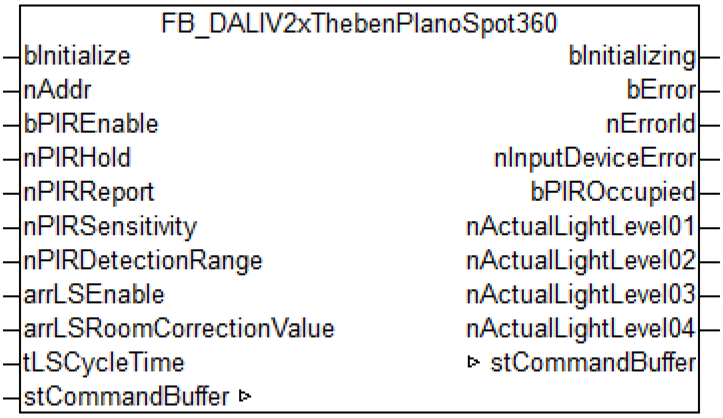
nErrorId: Contains the command-specific error code of the most recently executed command. Is reset to 0 by the execution of an instruction at the inputs. See Error codes [⏵ 385].

VAR_IN_OUT

```
stCommandBuffer       : ST_DALIV2CommandBuffer;
```

stCommandBuffer: A reference to the structure for communication with the FB_DALIV2Communication() [⏵ 93] (KL6811) or FB_KL6821Communication() [⏵ 101] (KL6821) block.

5.1.301 FB_DALIV2xThebenPlanoSpot360



This function block is used as an example and is therefore not included in the library. The function block can be downloaded as an export file and imported into the desired project. Adjustments can be made as required.

This function block evaluates the 3 measured brightness values and the presence of the ThebenHTS PlanoSpot DALI control unit. This function block can also be used to initialize the DALI control units.

The DALI control unit occupies a short address and contains 5 instances. Instance 0 is the movement sensor and complies with the IEC 62386-303 standard. The light sensors are stored in instances 1 to 4 and comply with the IEC 62386-304 standard.

A detected movement is sent as an event, while the brightness values are read out cyclically. Each instance can be decoded individually as required.

Download the TwinCAT 2 PLC Control export file: https://infosys.beckhoff.com/content/1033/tcplclibdali/Resources/12019302283/.zip

VAR_INPUT

```
bInitialize                 : BOOL := FALSE;
nAddr                       : BYTE;
(* Occupancy Sensor Parameters *)
bPIREnable                  : BOOL := TRUE;
nPIRHold                    : BYTE := 1;     (* 10 s *)
nPIRReport                  : BYTE := 30;    (* 30 s *)
nPIRSensitivity             : BYTE := 3;     (* standard *)
nPIRDetectionRange          : BYTE := 0;     (* standard *)
(* Light Sensor Parameters *)
arrLSEnable                 : ARRAY [1..4] OF BOOL := TRUE, TRUE, TRUE, TRUE;
arrLSRoomCorrectionValue    : ARRAY [1..4] OF BYTE := 30, 30, 30, 30; (* 0.3 *)
tLSCycleTime                : TIME := t#1m;
```

bInitialize: The DALI controller is initialized by a positive edge at this input. The parameters are written to the DALI control unit, which must be accessible via the short address *nAddr*. The output *bInitializing* is TRUE during initialization.

nAddr: Short address of the DALI control unit.

bPIREnable: Parameter: If this input is TRUE, the motion sensor (instance 0) is enabled. The occupancy sensor is disabled by FALSE.

nPIRHold: Parameter: Sets the *Hold Timer* value. If no movement is detected, the status of the motion sensor is only changed after the *Hold Timer* has expired. The unit is 10 s. This means that times of up to 42 min 20 s (value 254) are possible. The value 0 corresponds to 1 s while the value 255 is ignored.

nPIRReport: Parameter: Sets the value for the *Report Timer*. The motion sensor status is retransmitted after the *Report Timer* has expired, even if the status has not changed. The unit is 1 s. The value 0 disables the *Report Timer*. This means that times of up to 4 min 15 s (value 255) are possible.

nPIRSensitivity: Parameter: The detection sensitivity can be adjusted in five steps (1: very insensitive to 5: very sensitive).

nPIRDetectionRange: Parameter: The sensor supports two different values for the detection range(0: standard and 1: reduced).

arrLSEnable: Parameter: If this input is TRUE, the brightness sensor (instance 1) is enabled. The brightness sensor is disabled by FALSE.

arrLSRoomCorrectionValue: Parameter: The room correction factor can be used to calibrate the value measured by the detector with the value measured with a reference device (luxmeter).

tLSCycleTime: Cycle time in which the current actual values of the brightness sensors are read out.

Further details on the parameters can be found in the IEC 62386 standard and the manufacturer's documentation.

VAR_OUTPUT

```
bInitializing            : BOOL;
bError                   : BOOL;
nErrorId                 : UDINT;
nInputDeviceError        : BYTE;
(* Occupancy Sensor *)
bPIROccupied             : BOOL;
(* Light Sensor *)
nActualLightLevel01      : UINT;  (* Light sensor 1 (integral) *)
nActualLightLevel02      : UINT;  (* Light sensor 2 (inner) *)
nActualLightLevel03      : UINT;  (* Light sensor 3 (middle) *)
nActualLightLevel04      : UINT;  (* Light sensor 4 (window) *)
```

bInitializing: This output is TRUE during initialization.

bError: This output is switched to TRUE as soon as an error occurs during the execution of a command. The command-specific error code is contained in *nErrorId*.

nErrorId: Contains the command-specific error code of the most recently executed command. See Error codes [▶ 385].

nInputDeviceError: The status of the DALI control unit (INPUT DEVICE ERROR) is queried before the initialization. 0 means no error. The individual error numbers are manufacturer-specific.

bPIROccupied: This output indicates the status of the motion sensor.

nActualLightLevel01 ... nActualLightLevel04: These outputs indicate the status of the brightness sensors.

VAR_IN_OUT

```
stCommandBuffer    : ST_DALIV2CommandBuffer;
```

stCommandBuffer: Reference to the internal structure for communication with the function block FB_KL6821Communication() [▶ 101] (KL6821).

Requirements

Development environment	Target system	Required libraries
TwinCAT 2.11 R3/x64 from build 2260	PC/CX, BX or BC	TcDALIV2 library from V2.12.0

5.2 Variables

5.2.1 DALI ballast variables

Every DALI ballast has a certain number of variables (parameters) from which it is possible to read a variety of information or to modify individual parameters.

Name	Default value	Reset value	Valid range	Size	Comment
ACTUAL DIM LEVEL [▶ 364]	?	254	0, MIN LEVEL ... MAX LEVEL	1 byte	
POWER ON LEVEL [▶ 364]	254	254	1 ... 254	1 byte	
SYSTEM FAILURE LEVEL [▶ 364]	254	254	0 ... 255	1 byte	
MIN LEVEL [▶ 364]	PHYSICAL MIN LEVEL	PHYSICAL MIN LEVEL	PHYSICAL MIN LEVEL ... MAX LEVEL	1 byte	
MAX LEVEL [▶ 364]	254	254	MIN LEVEL ... 254	1 byte	
FADE RATE [▶ 364]	7	7	1 ... 15	1 byte	
FADE TIME [▶ 365]	0	0	0 ... 15	1 byte	
SHORT ADDRESS [▶ 365]	255	No change	0 ... 63, 255	1 byte	
SEARCH ADDRESS [▶ 365]	FF FF FF	FF FF FF	00 00 00 ... FF FF FF	3 bytes	
RANDOM ADDRESS [▶ 365]	FF FF FF	FF FF FF	00 00 00 ... FF FF FF	3 bytes	
GROUP 0-7 [▶ 366]	0	0	0 ... 255	1 byte	
GROUP 8-15 [▶ 366]	0	0	0 ... 255	1 byte	
SCENE 0 [▶ 366]	255	255	0 ... 255	1 byte	
...	...	...	...	...	
SCENE 15 [▶ 366]	255	255	0 ... 255	1 byte	
STATUS INFORMATION [▶ 366]	???? ????	0?10 0???	0 ... 255	1 byte	read only
VERSION NUMBER [▶ 366]	Manufacturer-dependent	Manufacturer-dependent	0 ... 255	1 byte	read only
DEVICE TYPE [▶ 366]	Manufacturer-dependent	Manufacturer-dependent	0 ... 255	1 byte	read only
PHYSICAL MIN LEVEL [▶ 367]	Manufacturer-dependent	Manufacturer-dependent	1 ... 254	1 byte	read only

?: Not specified

ACTUAL DIM LEVEL

This variable contains the power currently applying to the lamp.
The value can be read with the FB_DALIV2QueryActualLevel() [▸ 129] block.

POWER ON LEVEL

When power is supplied to the ballast the lamp is driven to the level of power specified in the variable POWER ON LEVEL. This assumes that the DALI bus has already been supplied with power and is idle. The range of values available to POWER ON LEVEL is restricted by the two variables MIN LEVEL and MAX LEVEL.
The variable can be read with the FB_DALIV2QueryPowerOnLevel() [▸ 145] block, and written with FB_DALIV2StoreDTRAsPowerOnLevel() [▸ 116].

SYSTEM FAILURE LEVEL

If a fault occurs on the DALI bus (the idle voltage remains below the specified level for longer than 500 ms) then the lamp is driven to the power specified by the SYSTEM FAILURE LEVEL variable. If the variable contains 255 (mask) the lamp power will not change. The possible range is limited by MIN LEVEL and MAX LEVEL.
The variable can be read with the FB_DALIV2QuerySystemFailureLevel() [▸ 152] block, and written with FB_DALIV2StoreDTRAsSystemFailureLevel() [▸ 118].

MIN LEVEL / MAX LEVEL

The ballast internally restricts the value of the output power to the lamp by means of the MIN LEVEL and MAX LEVEL variables. The exceptions to this are power values of 0 (off) and 255 (mask).

FADE RATE

The FADE RATE specifies the rate at which changes are made (in steps per second) in the value of the lamp's power. This variable has an effect on the FB_DALIV2Up() [▸ 128] and FB_DALIV2Down() [▸ 120] commands. The absolute fade rate is not entered directly, but it is calculated according to the following formula:

$$T = \frac{506}{\sqrt{2^n}}$$

T = absolute fade rate
n = value that is stored in the FADE RATE variable

The following values result:

n	absolute fade rate
0	Not permitted
1	357,796 steps/s
2	253.000 steps/s
3	178,898 steps/s
4	126,500 steps/s
5	89,449 steps/s
6	63,250 steps/s
7	44,725 steps/s
8	31,625 steps/s
9	22,362 steps/s
10	15,813 steps/s
11	11,181 steps/s
12	7,906 steps/s
13	5,591 steps/s

n	absolute fade rate
14	3,953 steps/s
15	2.795 steps/s

FADE TIME

The FADE TIME specifies the time allowed for the current lamp power to be changed to the requested value. In the case of a lamp that is switched off, the pre-heating and ignition time is not included in the fade time. The FB_DALIV2DirectArcPowerControl() [▶ 119] and FB_DALIV2GoToScene() [▶ 122] blocks are affected. The absolute fade time is not entered directly, but it is calculated according to the following formula:

$$T = \frac{1}{2}\sqrt{2^n}$$

T = absolute fade time

n = value that is stored in the FADE TIME variable

The following values result:

n	absolute fade time
0	< 0.707 s
1	0,707 s
2	1.000 s
3	1,414 s
4	2.000 s
5	2,828 s
6	4.000 s
7	5,657 s
8	8.000 s
9	11,314 s
10	16.000 s
11	22,627 s
12	32.000 s
13	45,255 s
14	64.000 s
15	90.510 s

SHORT ADDRESS

The short address is stored in this variable. A valid short address lies in the range between 0 and 63. The short address is regarded as having been deleted if 255 is written to the variable. The short address can be set with the FB_DALIV2SetShortAddress() [▶ 111] and FB_DALIV2StoreDTRAsShortAddress() [▶ 117] blocks. Calling the FB_DALIV2QueryMissingShortAddress() [▶ 142] queries whether a ballast still does not have a short address.

SEARCH ADDRESS

The search address is only needed when assigning short addresses.

RANDOM ADDRESS

The random address, also called the long address, is specified by the manufacturer when the ballasts are supplied. The FB_DALIV2QueryRandomAddress() [▶ 146] block can be used to read out the 3 bytes of the random address.

GROUP 0-7 / GROUP 8-15

16 groups exist within a DALI network. Any ballast can belong to one group, to several, or indeed to none. Commands that are to be sent to a group have an effect on all the ballasts that belong to that particular group. The FB_DALIV2QueryGroups() [▶ 136] block reads both 8-bit variables and assembles them into a single 16 bit value. Each bit indicates whether the ballast belongs to a particular group.

SCENE 0-15

Each DALI ballast can store lamp power values for 16 different scenes. There is a value of the lamp power for every scene. If the command for calling up a scene (FB_DALIV2GoToScene() [▶ 122]) is sent to one device, a group, or to all the devices (broadcast), then each of the affected lamps is set to the saved value. The output is limited by the values of MAX LEVEL, MIN LEVEL and PHYSICAL MIN LEVEL.

STATUS INFORMATION

The status information contains the most important items describing the status of a ballast. The 8-bit value can be read with the FB_DALIV2QueryStatus() [▶ 151] block. The significance of the individual bits is defined as follows:

Bit	Description
0	Status of the ballast. 0: OK.
1	Lamp failure. 0: OK.
2	Lamp power on. 0: OFF.
3	Limit value error. 0: the most recently requested lamp power was either between MIN LEVEL and MAX LEVEL or was OFF.
4	Fading completed: 0: fading finished. 1: fading active.
5	Reset status. 0: No.
6	Missing short address. 0: No.
7	Power supply fault. 0: No, A reset, or a lamp power control command has been received since the most recent power up.

VERSION NUMBER

VERSION NUMBER

The version number corresponds to the version number of the IEC standard in accordance with which the ballast was developed and manufactured. The version number can only be read, and is specified by the manufacturer of the ballasts. The major version (*nMajorVersion*) and the minor version (*nMinorVersion*) can each have a value in the range from 0 to 15 (4 bits).

DEVICE TYPE

DEVICE TYPE

The value can be read, for which the FB_DALIV2QueryDeviceType() [▶ 134] block is used. The following device types are defined according to the IEC 62386 standard:

Value	Description
0	Standard device
1	Device for emergency lighting [▶ 32].
2	Device for discharge lamps [▶ 36].
3	Device for low-voltage halogen lamps.
4	Device for dimming bulbs.
5	Device for converting digital signals into DC signals.
6	Device for light emitting diodes (LEDs).
7	Switching function.

Value	Description
8	Device for colour/colour temperature control [▶ 34].

PHYSICAL MIN LEVEL

PHYSICAL MIN LEVEL

The lowest physically possible lamp power level is stored by the manufacturer in the PHYSICAL MIN LEVEL variable. The value can only be read, for which the FB_DALIV2QueryPhysicalMinimumLevel() [▶ 143] block is used.

Also see about this

- LED modules - function blocks [▶ 36]

5.2.2 Emergency lighting variables

Each DALI emergency lighting supply unit has a certain number of variables (parameters) from which different information is read, or through which individual parameters can be modified.

Name	Default value	Reset value	Valid range	Size	Comment
EMERGENCY LEVEL [▶ 368]	EMERGENCY MAX LEVEL	No change	EMERGENCY MIN LEVEL ... EMERGENCY MAX LEVEL or MASK	1 byte	
EMERGENCY MIN LEVEL [▶ 368]	Manufacturer-dependent	No change	1 ... EMERGENCY MAX LEVEL or MASK	1 byte	read only
EMERGENCY MAX LEVEL [▶ 368]	Manufacturer-dependent	No change	EMERGENCY MIN LEVEL ... 254 or MASK	1 byte	read only
PROLONG TIME [▶ 368]	4	No change	0 ... 255	1 byte	
TEST DELAY TIME [▶ 368]	0	0	00 00 ... FF FF	2 bytes	
FUNCTION TEST DELAY TIME [▶ 369]	0	No change	00 00 ... FF FF	2 bytes	
DURATION TEST DELAY TIME [▶ 369]	0	No change	00 00 ... FF FF	2 bytes	
FUNCTION TEST INTERVAL [▶ 369]	7	No change	0, 1 ... 255	1 byte	
DURATION TEST INTERVAL [▶ 369]	13	No change	0, 1 ... 97	1 byte	
TEST EXECUTION TIMEOUT [▶ 370]	7	No change	0 ... 255	1 byte	
BATTERY CHARGE [▶ 370]	???? ????	No change	0 ... 255	1 byte	

Name	Default value	Reset value	Valid range	Size	Comment
DURATION TEST RESULT [▶ 370]	0	No change	0 ... 255	1 byte	
LAMP EMERGENCY TIME [▶ 370]	0	No change	0 ... 255	1 byte	
LAMP TOTAL OPERATION TIME [▶ 370]	0	No change	0 ... 255	1 byte	
RATED DURATION [▶ 370]	Manufacturer-dependent	No change	0 ... 255	1 byte	read only
EMERGENCY MODE [▶ 370]	???? ????	No change	0 ... 255	1 byte	
FEATURES [▶ 371]	Manufacturer-dependent	No change	0 ... 255	1 byte	read only
FAILURE STATUS [▶ 371]	???? ????	No change	0 ... 255	1 byte	
EMERGENCY STATUS [▶ 371]	???? ????	No change	0 ... 255	1 byte	
EXTENDED VERSION NUMBER [▶ 371]	1	No change	0 ... 255	1 byte	read only

?: Not specified

EMERGENCY LEVEL

This variable contains the emergency illuminance of the lamp. This value is limited via the variables EMERGENCY MAX LEVEL and EMERGENCY MIN LEVEL. The value can be read via the FB_DALIV2QueryEmergencyLevel() [▶ 223] block.

EMERGENCY MIN LEVEL / EMERGENCY MAX LEVEL

The emergency illuminance is limited via the variables EMERGENCY MIN LEVEL and EMERGENCY MAX LEVEL within the ballast. The exceptions to this are power values of 0 (off) and 255 (mask). The EMERGENCY MIN LEVEL and EMERGENCY MAX LEVEL values are specified by the manufacturer of the ballast. The FB_DALIV2QueryEmergencyMinLevel() [▶ 225] and FB_DALIV2QueryEmergencyMaxLevel() [▶ 224] blocks can be used to read the two variables from the ballast.

PROLONG TIME

The prolong time defines how long emergency mode is extended after mains voltage is available again. The unit is 30 seconds per step. The value can be read from the ballast via the FB_DALIV2QueryTestTiming() [▶ 234] block. The FB_DALIV2StoreDTRAsProlongTime() [▶ 246] block is used to write to this variable.

TEST DELAY TIME

The variables FUNCTION TEST DELAY TIME, DURATION TEST DELAY TIME FUNCTION TEST INTERVAL and DURATION TEST INTERVAL are set by means of the TEST DELAY TIME variable. This specifies the time behaviour associated with the functional test and the duration test as follows:

Function test:
Use FB_DALIV2SetDTR() [▶ 162] to write the high-order byte for the variable FUNCTION TEST DELAY TIME into the DTR (data transfer register).
Use FB_DALIV2StoreDTRAsTestDelayTimeHighByte() [▶ 247] to write the content of the DTR into the high-order byte of the 16-bit variable TEST DELAY TIME.
Use FB_DALIV2SetDTR() [▶ 162] to write the low-order byte for the variable FUNCTION TEST DELAY TIME into the DTR (data transfer register).
Use FB_DALIV2StoreDTRAsTestDelayTimeLowByte() [▶ 248] to write the content the DTR into the low-order byte of the 16-bit-variable TEST DELAY TIME.
Use FB_DALIV2SetDTR() [▶ 162] to write the value for the variable FUNCTION TEST INTERVAL into the DTR (data transfer register).
Use FB_DALIV2StoreDTRAsFunctionTestInterval() [▶ 245] to write the content of the DTR into the variable FUNCTION TEST INTERVAL. The content of the variable TEST DELAY TIME is copied into the variable FUNCTION TEST DELAY TIME.

Duration test:
Use FB_DALIV2SetDTR() [▶ 162] to write the high-order byte for the variable DURATION TEST DELAY TIME into the DTR (data transfer register).
Use FB_DALIV2StoreDTRAsTestDelayTimeHighByte() [▶ 247] to write the content of the DTR into the high-order byte of the 16-bit variable TEST DELAY TIME.
Use FB_DALIV2SetDTR() [▶ 162] to write the low-order byte for the variable DURATION TEST DELAY TIME into the DTR (data transfer register).
Use FB_DALIV2StoreDTRAsTestDelayTimeLowByte() [▶ 248] to write the content the DTR into the low-order byte of the 16-bit-variable TEST DELAY TIME.
Use FB_DALIV2SetDTR() [▶ 162] to write the value for the variable FUNCTION TEST INTERVAL into the DTR (data transfer register).
Use FB_DALIV2StoreDTRAsDurationTestInterval() [▶ 243] to write the content of the DTR into the variable DURATION TEST INTERVAL. The content of the variable TEST DELAY TIME is copied into the variable DURATION TEST DELAY TIME.

FUNCTION TEST DELAY TIME

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 unit of this variable is 15 minutes per step. This variable can be queried via the FB_DALIV2QueryTestTiming() [▶ 234] block. TEST DELAY TIME is used to write to this variable (see above).

DURATION TEST DELAY TIME

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 unit of this variable is 15 minutes per step. This variable can be queried via the FB_DALIV2QueryTestTiming() [▶ 234] block. TEST DELAY TIME is used to write to this variable (see above).

FUNCTION TEST INTERVAL

This variable is used to specify the interval time for the function test. The function test is executed periodically at these intervals. The unit of this variable is 1 day per step. This variable can be queried via the FB_DALIV2QueryTestTiming() [▶ 234] block. TEST DELAY TIME is used to write to this variable (see above).

DURATION TEST INTERVAL

This variable is used to specify the interval time for the duration test. The duration test is executed periodically at these intervals. The unit of this variable is 1 week per step. This variable can be queried via the FB_DALIV2QueryTestTiming() [▶ 234] block. TEST DELAY TIME is used to write to this variable (see above).

TEST EXECUTION TIMEOUT

The function test or duration test can be interrupted through various events. The variable TEST EXECUTION TIMEOUT can be used to specify the maximum execution time within which the respective test must be completed. The unit of this variable is 1 day per step. This variable can be queried via the FB_DALIV2QueryTestTiming() [▸ 234] block. The FB_DALIV2StoreDTRAsTestExecutionTimeout() [▸ 249] block is used to write to this variable.

BATTERY CHARGE

The current state of battery charge can be retrieved via this variable. 0 means minimum load, 254 means maximum load. If the ballast is unable to determine the state of charge, this variable contains the value 255. The FB_DALIV2QueryBatteryCharge() [▸ 221] block can be used to read this variable.

DURATION TEST RESULT

The result of a duration test is stored in this variable. The unit is 2 minutes per step. The value is only valid if bit 2 is set in the variable EMERGENCY STATUS (see below). The variable DURATION TEST RESULT can be queried with the function FB_DALIV2QueryDurationTestResult() [▸ 222].

LAMP EMERGENCY TIME

The emergency mode operating period of the lamp (supply via rechargeable batteries) is stored in this variable. The unit is 1 hour per step. This variable can be queried with the FB_DALIV2QueryLampEmergencyTime() [▸ 231] block and reset with the FB_DALIV2ResetLampTime() block [▸ 238].

LAMP TOTAL OPERATION TIME

The total lamp operating period is stored in this variable. The unit is 4 hours per step. This variable can be queried with the FB_DALIV2QueryLampTotalOperationTime() [▸ 232] block and reset with the FB_DALIV2ResetLampTime() [▸ 238] block.

RATED DURATION

The rated service time of the rechargeable battery is stored in this variable. The unit is 2 minutes per step. The value is specified by the manufacturer of the ballast and can be read with the FB_DALIV2QueryRatedDuration() [▸ 233] block.

EMERGENCY MODE

In EMERGENCY MODE the ballast stores the current mode. The FB_DALIV2QueryEmergencyMode() [▸ 226] block can be used to read the variable.

Bit	Description
0	Reset mode. 0: No.
1	Emergency mode readiness (normal operation). 0: No.
2	Emergency mode. 0: No.
3	Extended emergency mode once mains voltage is available again. 0: No.
4	Function test active. 0: No.
5	Duration test active. 0: No.
6	Connected suppress push button is active. 0: Not active or not available.
7	Connected mains voltage is active. 0: Off.

FEATURES

The performance characteristics supported by the ballast can be read from this variable via the FB_DALIV2QueryFeatures() [▸ 229] block. The content of this variable is specified by the manufacturer of the ballast and cannot be modified.

Bit	Description
0	Integrated emergency lighting supply unit. 0: No.
1	Emergency lighting supply unit in continuous mode. 0: No.
2	Switchable emergency lighting supply unit in continuous mode. 0: No.
3	Auto test capability. 0: No.
4	Adjustable emergency lighting illuminance. 0: No.
5	Connected suppress push button is supported. 0: No.
6	Physical selection is supported. 0: No.
7	Reserved

FAILURE STATUS

Possible fault states are displayed in this variable and can be read with the FB_DALIV2QueryFailureStatus() [▸ 228] block.

Bit	Description
0	Fault in the ballast circuit. 0: No.
1	Battery operation time fault. 0: No.
2	Battery fault. 0: No.
3	Emergency lamp fault. 0: No.
4	Timeout during function test. 0: No.
5	Timeout during duration test. 0: No.
6	Function test failed. 0: No.
7	Duration test failed. 0: No

EMERGENCY STATUS

The current state is displayed by the ballast. The FB_DALIV2QueryEmergencyStatus() [▸ 227] block can be used to read the content of the variable.

Bit	Description
0	Suppress mode. 0: No.
1	Function test is completed, and result is valid. 0: No.
2	Duration test is completed, and result is valid. 0: No.
3	Battery charger ready for operation. 0: running.
4	Start of function test delayed. 0: No.
5	Start of duration test delayed. 0: No.
6	Identification active. 0: No.
7	Selected. 0: No

EXTENDED VERSION NUMBER

The extended version number can be read with the FB_DALIV2QueryExtendedVersionNumber() [▸ 351] block. The version number can only be read, and is specified by the manufacturer of the ballasts.

5.2.3 Discharge lamps variables

Every DALI ballast for discharge lamps has a certain number of variables (parameters) from which it is possible to read a variety of information or to modify individual parameters.

Name	Default value	Reset value	Valid range	Size	Comment
HID STATUS [▸ 372]	0	0	0 ... 255	1 byte	
ACTUAL HID FAILURE [▸ 372]	???? ????	No change	0 ... 255	1 byte	
STORED HID FAILURE [▸ 373]	???? ????	No change	0 ... 255	1 byte	
HID FEATURES [▸ 373]	Manufacturer-dependent	No change	0 ... 255	1 byte	read only
THERMAL OVERLOAD TIME [▸ 373]	0	No change	00 00 ... FF FF	2 bytes	
THERMAL LOAD [▸ 373]	???? ????	No change	0 ... 255	1 byte	
EXTENDED VERSION NUMBER [▸ 374]	1	No change	0 ... 255	1 byte	read only

?: Not specified

HID STATUS

The current status of the ballast is stored in HID STATUS. The value can be read with the FB_DALIV2QueryHIDStatus() [▸ 252] block.

Bit	Description
0	Start-up time, ready for operation. 0: No.
1	The lamp power has reached its required set value. 0: No.
2	Ballast is waiting for the lamp to ignite. 0: No.
3	Reserved
4	Reserved
5	Reserved
6	Identification active. 0: No.
7	Reserved

ACTUAL HID FAILURE

This variable contains all the information about the fault status of the ballast. The variable is read with the FB_DALIV2QueryActualHIDFailure() [▸ 250] block. Each of the relevant bits is set as soon as a fault occurs and is automatically reset again as soon as the fault is rectified.

If either bit 4 or bit 7 is set, bit 1 in the STATUS INFORMATION [▸ 366] variable will also be set. In that case, the FB_DALIV2QueryLampeFailure() [▸ 138] block will return TRUE at the *bLampFailure* output.

Bit	Description
0	Supply voltage too low. 0: No.
1	Supply voltage too high. 0: No.
2	Converter too hot. 0: No.

Bit	Description
3	Reserved
4	Time for lamp ignition exceeded. 0: No.
5	Reserved
6	The lamp voltage outside specification. 0: No.
7	Lamp cycle error. 0: No.

STORED HID FAILURE

STORED HID FAILURE

This variable contains all the information about the fault status of the ballast. The variable is read with the FB_DALIV2QueryStoredHIDFailure() [▶ 253] block. The error messages are reset by switching off the ballast, or by the FB_DALIV2ResetStoredHIDFailure() [▶ 258] block.

Bit	Description
0	Supply voltage too low. 0: No.
1	Supply voltage too high. 0: No.
2	Converter too hot. 0: No.
3	Reserved
4	Time for lamp ignition exceeded. 0: No.
5	Reserved
6	The lamp voltage outside specification. 0: No.
7	Lamp cycle error. 0: No.

HID FEATURES

The performance characteristics supported by the ballast can be read from this variable via the FB_DALIV2QueryHIDFeatures() [▶ 251] block. The content of this variable is specified by the manufacturer of the ballast and cannot be modified.

Bit	Description
0	"Supply voltage too low" can be queried. 0: No.
1	"Supply voltage too high" can be queried. 0: No.
2	"Transformer too hot" can be queried. 0: No.
3	Reserved
4	Reserved
5	Reserved
6	"Lamp voltage outside specification" can be queried. 0: No.
7	Physical selection is supported. 0: No.

THERMAL OVERLOAD TIME

As soon as excess temperature is detected, the variable THERMAL OVERLOAD TIME is incremented in steps of 15 minutes. This variable cannot be reset. Reaching 65535 (0xFFFF) here corresponds to a time of 16,383 hours and 45 minutes or more. The THERMAL OVERLOAD TIME variable can be read by the FB_DALIV2QueryThermalOverloadTimeHB() [▶ 256] and FB_DALIV2QueryThermalOverloadTimeLB() [▶ 257] blocks.
The threshold value for detecting excess temperature is specified by the THERMAL LOAD variable (see below).

THERMAL LOAD

A percentage value in the range between 0% and 127.5% is specified in steps of 0.5% in the THERMAL LOAD variable. 255 corresponds here to a value of 127.5%. The variable can be read with the FB_DALIV2QueryThermalLoad() [▶ 254] block.

EXTENDED VERSION NUMBER

The extended version number can be read with the FB_QueryV2ExtendedVersionNumber() [▸ 351] block. The version number can only be read, and is specified by the manufacturer of the ballasts.

5.2.4 Colour/colour temperature control variables

Every DALI ballast for colour/colour 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.

Certain variables can be read out directly via DALI commands (e.g. FB_QueryColourStatus() [▸ 290] or FB_QueryRGBWAFControl () [▸ 297]). The FB_QueryColourValue() [▸ 293]function block can be used to read out further variables.

Name	Default value	Reset value	Valid range	Size	necessary colour representation (1) [▸ 374]	Comment
TEMPORARY x-COORDINATE	65535	65535	0 ...65535	2 bytes	0, 2	
REPORT x-COORDINATE	65535	65535	0 ... 65535	2 bytes	0	
x-COORDINATE	?	No change	0 ... 65535	2 bytes	0	
TEMPORARY x-COORDINATE	65535	65535	0 ... 65535	2 bytes	0, 2	
REPORT x-COORDINATE	65535	65535	0 ... 65535	2 bytes	0	
x-COORDINATE	?	No change	0 ... 65535	2 bytes	0	
TEMPORARY COLOUR TEMPERATURE Tc	65535	65535	1 ... 65535	2 bytes	1	
REPORT COLOUR TEMPERATURE Tc	65535	65535	1 ... 65535	2 bytes	1	
COLOUR TEMPERATURE Tc	?	No change	1 ... 65535	2 bytes	1	

Name	Default value	Reset value	Valid range	Size	necessary colour representation (1) [▶ 374]	Comment
COLOUR TEMPERATURE Tc COOLEST	?	COLOUR TEMPERATURE Tc PHYSICAL COOLEST	COLOUR TEMPERATURE Tc PHYSICAL COOLEST ... COLOUR TEMPERATURE Tc WARMEST, 65535	2 bytes	1	read only
COLOUR TEMPERATURE Tc WARMEST	?	COLOUR TEMPERATURE Tc PHYSICAL WARMEST	COLOUR TEMPERATURE Tc COOLEST ... COLOUR TEMPERATURE Tc PHYSICAL WARMEST, 65535	2 bytes	1	read only
COLOUR TEMPERATURE Tc PHYSICAL COOLEST	?	No change	1 - COLOUR TEMPERATURE Tc PHYSICAL WARMEST, 65535	2 bytes	1	read only
COLOUR TEMPERATURE Tc PHYSICAL WARMEST	?	No change	COLOUR TEMPERATURE Tc PHYSICAL COOLEST - 65534, 65535	2 bytes	1	read only
TEMPORARY PRIMARY N DIMLEVEL	65535	65535	0 ... 65535	up to 12 bytes	2	
REPORT PRIMARY N DIMLEVEL	65535	65535	0 ... 65535	up to 12 bytes	2	
PRIMARY N DIMLEVEL	?	No change	0 ... 65535	up to 12 bytes	2	
x-COORDINATE PRIMARY N	?	No change	0 ... 65535	up to 12 bytes	0, 2	read only
y-COORDINATE PRIMARY N	?	No change	0 ... 65535	up to 12 bytes	0, 2	read only
TY PRIMARY N	?	No change	0 ... 65535	up to 12 bytes	0, 2	read only

Name	Default value	Reset value	Valid range	Size	necessary colour representation (1) [▸ 374]	Comment
TEMPORARY RED DIMLEVEL	255	255	0 ... 255	1 byte	3	
REPORT RED DIMLEVEL	255	255	0 ... 255	1 byte	3	
RED DIMLEVEL	?	No change	0 ... 255	1 byte	3	
TEMPORARY GREEN DIMLEVEL	255	255	0 ... 255	1 byte	3	
REPORT GREEN DIMLEVEL	255	255	0 ... 255	1 byte	3	
GREEN DIMLEVEL	?	No change	0 ... 255	1 byte	3	
TEMPORARY BLUE DIMLEVEL	255	255	0 ... 255	1 byte	3	
REPORT BLUE DIMLEVEL	255	255	0 ... 255	1 byte	3	
BLUE DIMLEVEL	?	No change	0 ... 255	1 byte	3	
TEMPORARY WHITE DIMLEVEL	255	255	0 ... 255	1 byte	3	
REPORT WHITE DIMLEVEL	255	255	0 ... 255	1 byte	3	
WHITE DIMLEVEL	?	No change	0 ... 255	1 byte	3	
TEMPORARY AMBER DIMLEVEL	255	255	0 ... 255	1 byte	3	
REPORT AMBER DIMLEVEL	255	255	0 ... 255	1 byte	3	
AMBER DIMLEVEL	?	No change	0 ... 255	1 byte	3	
TEMPORARY FREECOLOUR DIMLEVEL	255	255	0 ... 255	1 byte	3	
REPORT FREECOLOUR DIMLEVEL	255	255	0 ... 255	1 byte	3	
FREECOLOUR DIMLEVEL	?	No change	0 ... 255	1 byte	3	

Name	Default value	Reset value	Valid range	Size	necessary colour representation (1) [▸ 374]	Comment
TEMPORARY RGBWAF CONTROL	255	255	0 ... 255	1 byte	3	
REPORT RGBWAF CONTROL	255	255	0 ... 255	1 byte	3	
RGBWAF CONTROL [▸ 379]	63	No change	0 ... 255	1 byte	3	
ASSIGNED COLOUR [▸ 378]	0x0102 0304 0506	0x0102 0304 0506	0x0000 0000 0000 ... 0x0606 0606 0606	6 bytes	3	read only MSB: Channel 0 LSB: Channel 5
TEMPORARY COLOUR TYPE [▸ 378]	255	255	0x10, 0x20, 0x40, 0x80, 0xFF	1 byte	0, 1, 2, 3	
REPORT COLOUR TYPE [▸ 378]	255	255	0x10, 0x20, 0x40, 0x80, 0xFF	1 byte	0, 1, 2, 3	
SCENE 0–15 COLOUR TYPE [▸ 378]	65535	65535	0x10, 0x20, 0x40, 0x80, 0xFF	16 bytes	0, 1, 2, 3	read only
SCENE 0–15 COLOUR VALUE	65535	65535	0 ... 65535	32 bytes ... 192 bytes	0, 1, 2, 3	read only
POWER ON COLOUR TYPE [▸ 378]	Manufacturer-dependent	Manufacturer-dependent	0x10, 0x20, 0x40, 0x80, 0xFF	1 byte	0, 1, 2, 3	read only
POWER ON COLOUR VALUE	Manufacturer-dependent	Manufacturer-dependent	0 ... 65535	2 bytes ... 12 bytes	0, 1, 2, 3	read only
SYSTEM FAILURE COLOUR TYPE [▸ 378]	Manufacturer-dependent	Manufacturer-dependent	0x10, 0x20, 0x40, 0x80, 0xFF	1 byte	0, 1, 2, 3	read only
SYSTEM FAILURE COLOUR VALUE	Manufacturer-dependent	Manufacturer-dependent	0 ... 65535	2 bytes ... 12 bytes	0, 1, 2, 3	read only
GEAR FEATURES/ STATUS [▸ 379]	??00 0001b	??00 0001b	??00 0000b, ??00 0001b	1 byte	0, 1, 2, 3	
COLOUR STATUS [▸ 378]	?	No change	0 ... 255	1 byte	0, 1, 2, 3	

Name	Default value	Reset value	Valid range	Size	necessary colour representation (1) [▸ 374]	Comment
COLOUR TYPE FEATURES [▸ 379]	?	No change	0 ... 255	1 byte	0, 1, 2, 3	read only

?: Not specified

In the case of 1-byte values the value 255 is also called *MASK*.

In the case of 2-byte values the value 65535 is also called *MASK*.

(1): Specifies the colour representation that the DALI ballast must support so that it contains the appropriate variable:

Value	Description
0	xy coordinates
1	Colour temperature Tc
2	Primary (colour) N
3	RGBWAF

ASSIGNED COLOUR

The assignment between output channel and colour is defined in ASSIGNED COLOUR. Each byte contains the colour of the corresponding channel.

The value can be read with the FB_DALIV2QueryAssignedColour() [▸ 288] block.

Value	Description
0	No colour assigned
1	Red
2	Green
3	Blue
4	White
5	Amber
6	Free colour

COLOUR TYPE

COLOUR TYPE defines the colour representations supported by the DALI ballast.

The values can be read out using the FB_DALIV2QueryColourValue() [▸ 293] function block.

Value	Description
0x10	xy coordinates
0x20	Colour temperature Tc
0x40	Primary (colour) N
0x80	RGBWAF
0xFF	No colour change

COLOUR STATUS

COLOUR TYPE contains information about the current status of the DALI ballast.

The values can be read out using the FB_DALIV2QueryColourStatus() [▸ 290] function block.

Bit	Description
0	xy-coordinate colour point lies outside the valid range.
1	Colour temperature Tc lies outside the valid range.
2	Automatic calibration is active.
3	Automatic calibration was successful.
4	Colour representation *xy-coordinate* active.
5	Colour representation *colour temperature Tc* active.
6	Colour representation *primary N* active.
7	Colour representation *RGBWAF* active.

GEAR FEATURES/STATUS

GEAR FEATURES/STATUS contains information about the current status of the DALI ballast.

The values can be read out using the FB_DALIV2QueryGearFeaturesStatus() [▸ 296] function block and written using the FB_DALIV2StoreGearFeaturesStatus() [▸ 310] function block.

Bit	Description
0	Automatic activation.
1 - 5	Reserved.
6	Automatic calibration is supported.
7	Restoration of the automatic calibration is supported.

RGBWAF CONTROL

RGBWAF CONTROL contains further information about the assignment between output channel and colour.

The values can be read out using the FB_DALIV2QueryColourValue() [▸ 293] function block.

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 colour
6 - 7	00 = channel control 01 = colour control 10 = standardised colour control 11 = reserved

COLOUR TYPE FEATURES

COLOUR TYPE FEATURES defines the colour representations supported by the DALI ballast.

The values can be read out using the FB_DALIV2QueryColourTypeFeatures() [▸ 291] function block.

Bit	Description
0	The ballast supports the colour representation by xy coordinates.
1	The ballast supports the colour representation by colour temperature Tc.
2 - 4	The number of primary colours supported by the ballast. A value of 0 means that this colour representation by primary colours is not supported.
5 - 7	The number of RGBWAF channels supported by the ballast. A value of 0 means that this colour representation by RGBWAF is not supported.

5.2.5 Philips proprietary discharge lamps variables

Name	Default value	Reset value	Valid range	Size	Comment
PAEC_ENABLED	0	No change	0 ... 1	1 Byte	
CONTROL GEAR SEGMENT ADDRESS	0	No change	0 ... 255	1 Byte	
SEGMENT ADDRESS	0	0	0 ... 255	1 Byte	
UIC	Manufacturer-dependent	No change	00 00 00 00 00 00 00 00 ... FF FF FF FF FF FF FF FF	8 Bytes	read only
LAMPTYPE	0	No change	0 ... 255	1 Byte	
HID LAMP LEVEL	???? ????	0	0 ... 255	1 Byte	
FADE UP	0	No change	0 ... 255	1 Byte	
FADE DOWN	0	No change	0 ... 255	1 Byte	
FAILURE STATUS	0	0	0 ... 255	1 Byte	
OPERATION TIME	0	No change	0 ... 255	1 Byte	
OVERTEMPERATURE TIME	0	No change	0 ... 255	1 Byte	
CONTROL GEAR TEMPERATURE	0	0	0 ... 255	1 Byte	
OVERTEMPERATURE LEVEL	255	No change	0 ... 255	1 Byte	
TIMEOUT	0	No change	0 ... 255	1 Byte	
MAINS VOLTAGE	0	0	0 ... 255	1 Byte	
LAMP VOLTAGE	0	0	0 ... 255	1 Byte	

?: Not specified

5.3 Data types

5.3.1 E_DALIV2AddrType

```
TYPE E_DALIV2AddrType :
(
  eDALIV2AddrTypeShort       := 0,
  eDALIV2AddrTypeGroup       := 1,
  eDALIV2AddrTypeBroadcast   := 2
);
END_TYPE
```

5.3.2 E_DALIV2CommandPriority

```
TYPE E_DALIV2CommandPriority :
(
  eDALIV2CommandPriorityHigh     := 0,
  eDALIV2CommandPriorityMiddle   := 1,
  eDALIV2CommandPriorityLow      := 2
);
END_TYPE
```

5.3.3 E_DALIV2ConfigurationCommands

```
TYPE E_DALIV2ConfigurationCommands :
(
  eDALIV2CommandDoNothing       := 0,
  eDALIV2CommandOff             := 1,
  eDALIV2CommandRecallMaxLevel  := 2,
  eDALIV2CommandRecallMinLevel  := 3,
  eDALIV2CommandGoToScene0      := 4,
  eDALIV2CommandGoToScene1      := 5,
  eDALIV2CommandGoToScene2      := 6,
  eDALIV2CommandGoToScene3      := 7,
  eDALIV2CommandGoToScene4      := 8,
  eDALIV2CommandGoToScene5      := 9,
  eDALIV2CommandGoToScene6      := 10,
  eDALIV2CommandGoToScene7      := 11,
  eDALIV2CommandGoToScene8      := 12,
  eDALIV2CommandGoToScene9      := 13,
  eDALIV2CommandGoToScene10     := 14,
  eDALIV2CommandGoToScene11     := 15,
  eDALIV2CommandGoToScene12     := 16,
  eDALIV2CommandGoToScene13     := 17,
  eDALIV2CommandGoToScene14     := 18,
  eDALIV2CommandGoToScene15     := 19
);
END_TYPE
```

5.3.4 E_DALIV2CurrentAddressingState

```
TYPE E_DALIV2CurrentAddressingState :
(
  eDALIV2AddrStateIdle             := 0,
  eDALIV2AddrStateRemoveLamp       := 1,
  eDALIV2AddrStateReinsertLamp     := 2,
  eDALIV2AddrStateAddressingLamp   := 3
);
END_TYPE
```

5.3.5 E_DALIV2DataFrameType

```
TYPE E_DALIV2DataFrameType :
(
  eDALIV2DataFrameType16Bit    := 1,
  eDALIV2DataFrameType24Bit    := 3,
  eDALIV2DataFrameTypeOsram    := 6
);
END_TYPE
```

5.3.6 E_DALIV2DimmingCurve

```
TYPE E_DALIV2DimmingCurve :
(
  eDALIV2DimmingCurveLogarithmic   := 0,
  eDALIV2DimmingCurveLinear        := 1
);
END_TYPE
```

5.3.7 E_DALIV2EventScheme

```
TYPE E_DALIV2EventScheme :
(
  eDALIV2EventSchemeUnkown           := -1,
  eDALIV2EventSchemeInstance         := 0,
(* (default) Instance addressing, using instance type and number. *)
  eDALIV2EventSchemeDevice           := 1,
(* Device addressing, using short address and instance type. *)
  eDALIV2EventSchemeDeviceInstance   := 2, (* Device/
instance addressing, using short address and instance number. *)
  eDALIV2EventSchemeDeviceGroup      := 3,
(* Device group addressing, using device group and instance type. *)
  eDALIV2EventSchemeInstanceGroup    := 4
(* Instance group addressing, using instance group and type. *)
);
END_TYPE
```

5.3.8 E_DALIV2InstAddrType

```
TYPE E_DALIV2InstAddrType :
(
  eDALIV2InstAddrTypeNumber          := 0,    (* Instance number (0-31) *)
  eDALIV2InstAddrTypeGroup           := 1,    (* Instance group (0-31) *)
  eDALIV2InstAddrTypeType            := 2,    (* Instance type (0-31) *)
  eDALIV2InstAddrTypeFeatureNumber   := 3,    (* Feature on instance number level (0-31) *)
  eDALIV2InstAddrTypeFeatureGroup    := 4,    (* Feature on instance group level (0-31) *)
  eDALIV2InstAddrTypeFeatureType     := 5,    (* Feature on instance type level (0-31) *)
  eDALIV2InstAddrTypeFeatureBroadcast:= 6,    (* Feature on instance broadcast level *)
  eDALIV2InstAddrTypeBroadcast       := 7,    (* Instance broadcast *)
  eDALIV2InstAddrTypeFeatureDevice   := 8,    (* Feature on device level *)
  eDALIV2InstAddrTypeDevice          := 9     (* Device *)
);
END_TYPE
```

5.3.9 E_DALIV2OperationMode

```
TYPE E_DALIV2OperationMode :
(
  eDALIV2OperationModeDALI    := 0,
  eDALIV2OperationModeDSI     := 1
);
END_TYPE
```

5.3.10 E_DALIV2PowerSupplyMode

```
TYPE E_DALIV2PowerSupplyMode :
(
  eDALIV2PowerSupplyModeOn    := 0,
  eDALIV2PowerSupplyModeOff   := 1,
  eDALIV2PowerSupplyModeAuto  := 2
);
END_TYPE
```

5.3.11 ST_DALIV2ChangeAddressList

```
TYPE E_DALIV2CurrentAddressingState :
STRUCT
    nOldAddress             : BYTE;
    nNewAddress             : BYTE;
    nRandomAddressHigh      : BYTE;
    nRandomAddressMiddle    : BYTE;
    nRandomAddressLow       : BYTE;
    nErrors                 : DWORD;
END_STRUCT
END_TYPE
```

5.3.12 ST_DALIV2ControlTable

```
TYPE ST_DALIV2ControlTable :
STRUCT
  nActualValue     : UINT;
  nControlValue    : BYTE;
  nSwitchRange     : UINT;
END_STRUCT
END_TYPE
```

nActualValue: Measured light value.

nControlValue: Control value of a node. Valid range: 0 or *nMinLevelMasterDev* ... *nMaxLevelMasterDev*.

nSwitchRange: Switching range around the node. *nSwitchRange* can only be "0" for the first node of the unused table-range.

5.3.13 ST_DALIV2DeviceSettings

```
TYPE ST_DALIV2DeviceSettings :
STRUCT
  nErrors                : DWORD;
  bPresent               : BOOL;
  nActualLevel           : BYTE;
  nPowerOnLevel          : BYTE;
  nSystemFailureLevel    : BYTE;
  nMinLevel              : BYTE;
  nMaxLevel              : BYTE;
  nFadeRate              : BYTE;
  nFadeTime              : BYTE;
  nRandomAddress         : DWORD;
  nGroups                : WORD;
  nSceneLevels           : ARRAY [0..15] OF BYTE;
  nStatus                : BYTE;
  nMajorVersion          : BYTE;
  nMinorVersion          : BYTE;
  nDeviceType            : BYTE;
  nPhysicalMinLevel      : BYTE;
END_STRUCT
END_TYPE
```

5.3.14 ST_DALIV2DeviceSettingsType01

```
TYPE ST_DALIV2DeviceSettingsType01 :
STRUCT
  nErrors                   : DWORD;
  bPresent                  : BOOL;
  nBatteryCharge            : UINT;     (* 0..254, 255->Error *)
  tDurationTestResult       : TIME;     (* 0..510 min *)
  tLampEmergencyTime        : TIME;     (* 0..255 h *)
  tLampTotalOperationTime   : TIME;     (* 0..1024 h *)
  nEmergencyLevel           : BYTE;     (* 0..254 *)
  nEmergencyMinLevel        : BYTE;     (* 0..254 *)
  nEmergencyMaxLevel        : BYTE;     (* 0..254 *)
  tRatedDuration            : TIME;     (* 0..510 min *)
  nNextFunctionTest         : UINT;     (* 0..255 *)
  nNextDurationTest         : UINT;     (* 0..255 *)
  nFunctionTestInterval     : UINT;     (* 0..255 *)
  nDurationTestInterval     : UINT;     (* 0..255 *)
  nTestExecutionTimeout     : UINT;     (* 0..255 *)
  nProlongTime              : UINT;     (* 0..255 *)
  nEmergencyMode            : BYTE;
  nFeatures                 : BYTE;
  nFailureStatus            : BYTE;
  nEmergencyStatus          : BYTE;
END_STRUCT
END_TYPE
```

The following variables undergo an adjustment to their target display range when read, hence deviating from the read display of the DALI device:

- tDurationTestResult
- tLampEmergencyTime

- tLampTotalOperationTime
- tRatedDuration

In the case of the remaining variables, the conversion either does not make sense (e.g. *nEmergencyLevel*) or display is not possible, as for example in the case of *nNextDurationTest*, since the range of the variable of type TIME is insufficient.

5.3.15 ST_DALIV2FileLogging

```
TYPE ST_DALIV2FileLogging :
STRUCT
  sTimestamp          : STRING(30);
  sController         : STRING(20);
  sLineName           : STRING(10);
  sAddress            : STRING(2);
  sDescription        : STRING(20);
  sLocation           : STRING(20);
  sTestDuration       : STRING(8);
  sResult             : STRING(240);
END_STRUCT
END_TYPE
```

5.3.16 ST_DALIV2InData

```
TYPE ST_DALIV2InData :
STRUCT
  nStatus     : BYTE;
  nDummy      : BYTE;
  nData       : WORD;
END_STRUCT
END_TYPE
```

5.3.17 ST_DALIV2OutData

```
TYPE ST_DALIV2OutData :
STRUCT
  nCtrl        : BYTE;
  nDummy       : BYTE;
  nData        : WORD;
END_STRUCT
END_TYPE
```

5.3.18 ST_DALIV2SequenceTable

```
TYPE ST_DALIV2SequenceTable :
STRUCT
  nTargetValue     : BYTE;
  tRampTime        : TIME;
  tProlongTime     : TIME;
END_STRUCT
END_TYPE
```

nTargetValue: Target-value.

tRampTime: Time to reach the target-value.

tProlongTime: Time to stay on the target-value.

5.3.19 ST_DALIV2SwapShortAddressList

```
TYPE ST_DALIV2SwapShortAddressList :
STRUCT
  bShortAddressValid     : BOOL;
  nNewShortAddress       : BYTE;
END_STRUCT
END_TYPE
```

5.3.20 ST_KL6821InData

```
TYPE ST_KL6821InData :
STRUCT
  nStatus       : WORD;
  arrData       : ARRAY [0..3] OF BYTE;
END_STRUCT
```

5.3.21 ST_KL6821OutData

```
TYPE ST_KL6821OutData :
STRUCT
  nCtrl    : WORD;
  arrData       : ARRAY [0..3] OF BYTE;
END_STRUCT
END_TYPE
```

5.4 Error codes

Value (hex)	Value (dec)	Description
0x0000	0	No error.
0x0001	1	No answer from the DALI terminal.
0x0002	2	No answer from the DALI ballast.
0x0003	3	Communication buffer overflow.
0x0004	4	No answer from the communication block.
0x0005	5	DALI collision detected on the backward channel: During the transfer of a DALI telegram, a collision with the send data of another DALI slave was detected.
0x0006	6	DALI collision detected on the forward channel: During the transfer of a DALI telegram, a collision with the send data of another DALI master was detected. The error also occurs when the 24 V is missing on the power contacts of the KL6811.
0x0007	7	When using the internal DALI power supply unit: overload of the internal DALI power supply unit of the KL6811 (bus under-voltage).
0x0008	8	Parameter *eCommandPriority* is out of range.
0x0009	9	Parameter *eAddrType* is out of range.
0x000A	10	Parameter *nAddr* is a short address and is out of range.
0x000B	11	Parameter *nAddr* is a group address and is out of range.
0x000C	12	Parameter *nGroup* is out of range.
0x000D	13	Parameter *nScene* is out of range.
0x000E	14	Parameter *nStartWithShortAddress* is out of range.
0x000F	15	no further free short addresses.
0x0010	16	Parameter *nNewShortAddress* is out of range.
0x0011	17	Parameter *nShortAddress01* is out of range.
0x0012	18	Parameter *nShortAddress02* is out of range.
0x0013	19	Parameter *nFreeShortAddress* is out of range.
0x0014	20	The short address in parameter *nFreeShortAddress* is occupied within the DALI line.
0x0015	21	Parameter *arrSwapShortAddressList* contains invalid values.
0x0016	22	Parameter *nHysteresis* is out of range.
0x0017	23	Parameter *nProlongValue* is not in the valid range.
0x0018	24	Actual value of Master-Device is 255 for too long -> Device is possibly broken.
0x0019	25	Parameter *nEndLevel* is not in the valid range.
0x001A	26	Target-value *nEndLevel* is not reached after twice the ramp-time.

Value (hex)	Value (dec)	Description
0x001B	27	FB_DALIV2LightControl() [▸ 68]: One or more setpoint-values *(nSetpoint)* in the table *arrControlTable* are not in the valid DALI-Range (0..254).
0x001C	28	FB_DALIV2LightControl() [▸ 68]: The switch-range *nSwitchRange* of the 1st or 2nd element of the table *arrControlTable* is “0”. Thus it´s assumed, that the table has no or only one element.
0x001D	29	FB_DALIV2LightControl() [▸ 68]: At least one input-value *nActualValue* in the table *arrControlTable* lies in the switch range of its neighbour.
0x001E	30	FB_DALIV2LightControl() [▸ 68]: At least one setpoint *(nSetpoint)* in the table *arrControlTable* is not in the valid range. Will only be checked after starting with *bStart*.
0x001F	31	An internal state-query of the Master-Device tells that the state "fading active" is set for too long after the expected end of the dimming. See also STATUS INFORMATION [▸ 366]
0x0020	32	Parameter *nPresenceValue* is not in the valid range.
0x0021	33	Timeout during internal addressing (see also FB_DALIV2AddressingIntRandomAddressing() [▸ 42]). The terminal has not sent a reply following the start of internal addressing.
0x0022	34	The terminal has returned an error during internal addressing (see also FB_DALIV2AddressingIntRandomAddressing() [▸ 42]).
0x0023	35	At least one test is still pending or active.
0x0024	36	Device in automatic-test-mode.
0x0025	37	Battery not fully charged for duration-test.
0x0026	38	Test canceled, no valid emergency-Status/Mode during test.
0x0027	39	Test canceled Test-mode not reached.
0x0028	40	Test canceled because of timeout.
0x0029	41	Error executing DALI-command.
0x002A	42	Error writing into log-file.
0x002B	43	Device not in 'normal-mode'.
0x002C	44	FB_DALIV2Sequencer() [▸ 72]: The start-index *nStartIndex* is not within the valid range (1..50).
0x002D	45	FB_DALIV2Sequencer() [▸ 72]: The start-index *nStartIndex* points to an entry, which is marked as a sequence-end (zero-entry).
0x002E	46	FB_DALIV2ChangeAddressList() [▸ 46]: The change-list *arrChangeAddressList* is empty.
0x002F	47	FB_DALIV2ChangeAddressList() [▸ 46]: The change-list *arrChangeAddressList* contains an invalid short-addresses (>63).
0x0030	48	FB_DALIV2ChangeAddressList() [▸ 46]: The change-list *arrChangeAddressList* contains an double-entry regarding either the new or the old short-addresses.
0x0031	49	FB_DALIV2ChangeAddressList() [▸ 46]: The change-list *arrChangeAddressList* containes an new-address which is already used by a device not affected by the changes in the DALI line. Adresses are re-programmed.
0x0032	50	FB_KL6811Config() [▸ 99]: An error occurred during configuration of the terminal.
0x0033	51	FB_KL6811Config() [▸ 99]: Parameter *eOperationMode* is not in the valid range.
0x0034	52	The constant DALI_MESSAGE_QUEUE_ENTRIES is outside the valid range (2-250).
0x0035	53	The constant DALI_RESPONSE_TABLE_ENTRIES is outside the valid range (2-250).

Value (hex)	Value (dec)	Description
0x0036	54	The constant DALI_EVENT_TABLE_ENTRIES is outside the valid range (2-250).
0x0037	55	When using the internal DALI power supply unit: Power supply unit fault detected.
0x0038	56	The process image was disabled by the DI1 or DI2 inputs of the terminal.
0x0039	57	Parameter *eInstAddrType* is outside the valid range.
0x003A	58	Parameter *eDataFrameType* is outside the valid range.
0x003B	59	DSI is not supported by the Bus Terminal.
0x003C	60	Parameter *nEventPriority* is outside the valid range.
0x003D	61	Parameter *nGroup* lies outside of the valid range.
0x003E	62	Parameter *nInstanceGroup* is outside the valid range.
0x003F	63	Parameter *eEventScheme* is outside the valid range.
0x0040	64	Parameter *eEventFilter* is outside the valid range.
0x0041	65	Parameter *nInstAddr* is outside the valid range.
0x0042	66	Parameter *ePowerSupplyMode* is outside the valid range.
0x0043	67	Parameter *eCommandKBusWatchdog* is outside the valid range.
0x0044	68	Parameter *eCommandDI1RisingEdge* is outside the valid range.
0x0045	69	Parameter *eCommandDI1FallingEdge* is outside the valid range.
0x0046	70	Parameter *eCommandDI2RisingEdge* is outside the valid range.
0x0047	71	Parameter *eCommandDI2FallingEdge* is outside the valid range.
0x0048	72	During internal addressing (see also FB_DALIV2AddressingIntRandomAddressing [▸ 42]()) the terminal has detected that there is no further short address available.
0x0049	73	During internal addressing (see also FB_DALIV2AddressingIntRandomAddressing [▸ 42]()) the terminal has detected that several devices have the same long address.
0x004A	74	Internal addressing (see also FB_DALIV2AddressingIntRandomAddressing [▸ 42]()) has failed 3 times.
0x004B	75	The communication buffer for sending the DALI commands has been blocked for longer than permitted.
0x004C	76	The constant DALIV2_TIMEOUT_LOCK_MESSAGE_QUEUE is outside the valid range (0-15 s).
0x004D	77	During internal addressing (see also FB_DALIV2AddressingIntRandomAddressing [▸ 42]()) the terminal has detected a short circuit on the bus.
0x004E	78	Short circuit detected on the DALI bus.

6 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: https://www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Beckhoff Support

Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
Fax:	+49 5246 963 9157
e-mail:	support@beckhoff.com

Beckhoff Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
Fax:	+49 5246 963 479
e-mail:	service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
Fax:	+49 5246 963 198
e-mail:	info@beckhoff.com
web:	https://www.beckhoff.com

More Information:

www.beckhoff.com/tx1200

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com