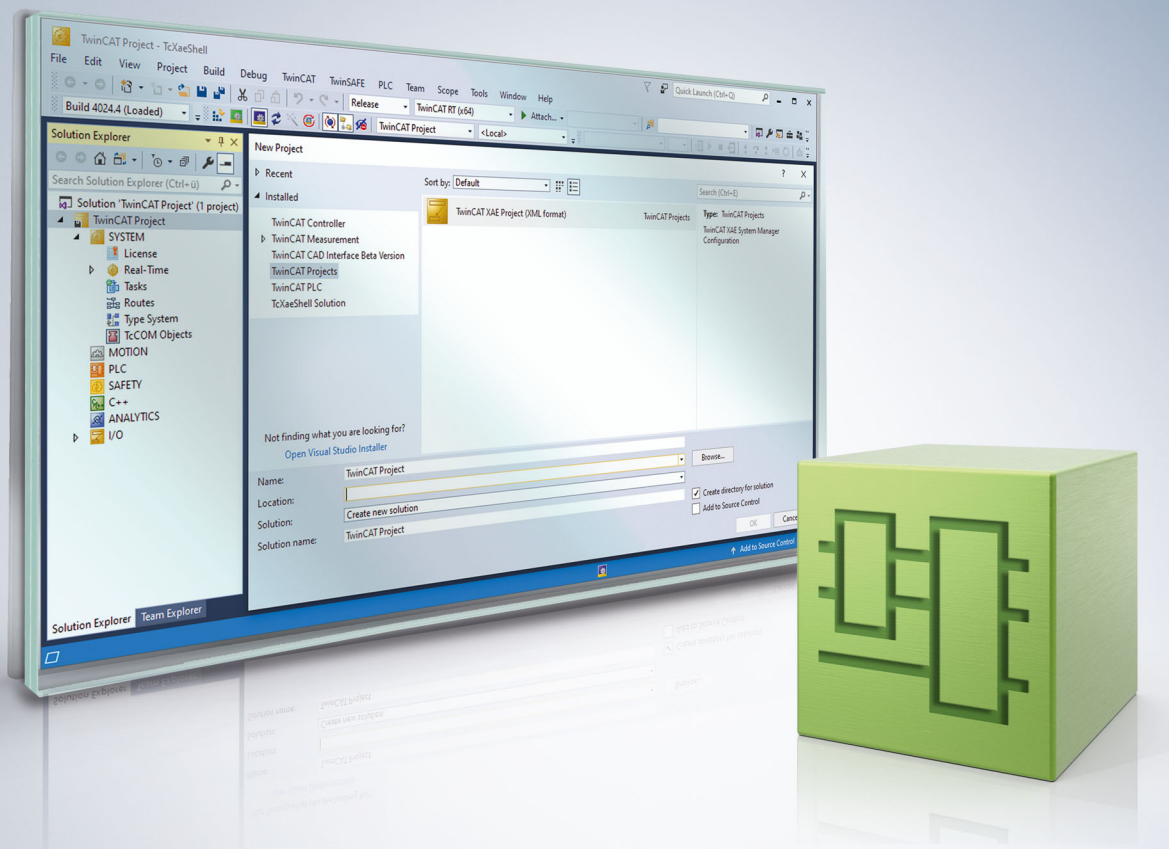
BECKHOFF New Automation Technology

Manual | EN

TE1000

TwinCAT 3 | PLC Library: Tc3_IOLink



2023-08-09 | Version: 1.1.0

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 For your safety

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

⚠ DANGER
Hazard with high risk of death or serious injury.

⚠ WARNING
Hazard with medium risk of death or serious injury.

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

Warning of damage to property or environment

NOTICE
The environment, equipment, or data may be damaged.

Information on handling the product

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

1.3 Notes on information security

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

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

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

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

2 Overview

The PLC library Tc3_IoLink is used for communication with IoLink devices.

Function blocks are available for this purpose that support the "Common Profile" and "Smart Sensor Profile" and enable parameters to be read and written.

System requirements

Target System	WinXP, WES, Win7, WES7, WEC7 IPC or CX, (x86, x64, ARM)
Min. TwinCAT Version	3.1.4024.25
Min. TwinCAT Level	TC1200 TC3 PLC

3 Data types and enums

3.1 Identification and diagnosis

The enums and structures are used in combination with the IOL_IdentificationAndDiagnosis [▶ 29].

3.1.1 E_Iollad_Function

Specifies the function at the function block IOL_IdentificationAndDiagnosis [▶ 29].

Name	Value	Description
NoFunction	0x00	A request is neglected, no function is executed.
ReadAll	0x01	A request starts reading back the current identification and diagnostic parameter values from the device.
Read	0x02	A request starts reading back the current diagnostic parameter values by reading DeviceStatus and DetailedDeviceStatus from the device.
WriteIdent	0x03	A request causes a previously created value for ApplicationSpecificTagIn, LocationTagIn and FunctionTagIn to be written to the device.

3.1.2 E_Iollad_State

State of the function block IOL_IdentificationAndDiagnosis [▶ 29].

Name	Description
Idle	Function block is in idle state
Startup	Function block is started
Done	Function block has completed
ReadProfileCharacteristic	Write identification parameters
ReadVendorIDAndDeviceID	Reads vendor and device ID
ReadVendorName	Reads vendor name
ReadVendorText	Reads vendor text
ReadProductName	Reads product name
ReadProductID	Reads the product ID
ReadProductText	Reads the product text
ReadSerialNumber	Reads the serial number
ReadHardwareRev	Reads the hardware revision
ReadFirmwareRev	Reads the firmware revision
ReadApplicationTag	Reads the ApplicationSpecificTag
ReadFunctionTag	Reads the FunctionTag
ReadLocationTag	Reads the LocationTag
ReadDiag	Reads out the diagnosis
ReadDiagDetails	Reads out the DetailedDiagnosis
WriteApplicationTag	Writes the ApplicationTag
WriteLocationTag	Writes the LocationTag
WriteFunctionTag	Writes the FunctionTag
WriteBackup	Triggers the data storage

3.1.3 E_IolProfileIdentifier

Defines the profiles that an IO-Link device can have. One device can support multiple profiles.

Name	Value	Description
NoProfileSupported	0x0000	No profile is supported
FunctionClass	0x0001	Function classes are supported
Switching sensor profile type 1		
SSP_1_1	0x0002	Defines the fixed switching sensor with the profile 0x8005 without 0x800C
SSP_1_2	0x0003	Defines the fixed switching sensor with the profile 0x8005 and 0x800C
Switching sensor profile type 2		
SSP_2_1	0x0004	Defines the adjustable switching sensor with the profile 0x8007 without 0x800C
SSP_2_2	0x0005	Defines the adjustable switching sensor with the profile 0x8008 without 0x800C
SSP_2_3	0x0006	Defines the adjustable switching sensor with the profile 0x8009 without 0x800C
SSP_2_4	0x0007	Defines the adjustable switching sensor with the profile 0x8007 with 0x800C
SSP_2_5	0x0008	Defines the adjustable switching sensor with the profile 0x8008 with 0x800C
SSP_2_6	0x0009	Defines the adjustable switching sensor with the profile 0x8009 with 0x800C
Measuring Device profile types 3		
SSP_3_1	0x000A	Defines the profiles 0x800A without 0x800C
SSP_3_2	0x000B	Defines the profiles 0x800B without 0x800C
SSP_3_3	0x000C	Defines the profiles 0x800A with 0x800C
SSP_3_4	0x000D	Defines the profiles 0x800B with 0x800C
Overview of Functionclasses		
Dev_Ident	0x8000	With this profile, the Firmware Revision, ApplicationSpecificTag and ProductID parameters must be supported by the IO-Link device
Multi_Ch	0x8001	Defines the process data type of the IO-Link device
PDV	0x8002	Defines the process data transmission of the device.
Dev_Diag	0x8003	With this profile, the Device Status and Detailed Device Status parameters must be supported by the IO-Link device
Teach_Ch	0x8004	
Fixed_SSC	0x8005	Fixed switching sensors in the Smart Sensor Profile are devices that support exactly one binary output signal (switching signal). The setpoint of this switching output is predefined during the manufacturing process and is thus fixed for the application.
Adjust_SSC	0x8006	
Teach_Single_Value	0x8007	A setpoint is defined by a "Teachpoint" (TP). The teach-in process is "static", i.e. the measured value is constant during teach-in.
Teach_Two_Value	0x8008	A setpoint is defined by two "Teachpoints" (TP).
Teach_Dyn	0x8009	"Dynamic teach-in" (within a time period) A single setpoint or both setpoints of an SSC are taught-in via acquired measured values during a specific time period. The teach-in is "dynamic", i.e. the measured value is not constant during the teach-in. As a rule, the minimum and maximum values within this timespan are used to determine the setpoints.

Name	Value	Description
Meas_Data_Ch_Std	0x800A	The function class provides a standardized process data structure and some additional information on how to interpret the transmitted data, such as physical unit or measurement limits. (default resolution)
Meas_Data_Ch_High	0x800B	The function class provides a standardized process data structure and some additional information on how to interpret the transmitted data, such as physical unit or measurement limits. (high resolution)
Sensor_Ctrl	0x800C	The deactivation signal can be used to switch off the sensor transducer.
Example of the profile identification of an extended profile		
Ident_And_Diag	0x4000	Combines the profiles 0x8000, 0x8002, 0x8003 and 0x8100
Visual_Localizations	0x8100	With this profile, the FunctionTag and Location parameters must be supported by the IO-Link device

3.1.4 E_IolProfileIdentifier_Bounds

Value range of the profiles and function classes.

Name	Value	Description
NoProfileSupported	0x0000	No profile supported
DeviceProfileID_Start	0x0001	Profile range start
DeviceProifleID_End	0x3FFF	Profile range end
CommonApplicationProfileID_start	0x4000	CommonApplicationProfileID start
CommonApplicationProfileID_end	0x7FFF	CommonApplicationProfileID end
FunctionClassID_start	0x8000	Function class range start
FunctionClassID_end	0xBFFF	Function class range end
Reserved_start	0xC000	Reserved range start
Reserved_end	0xFFFF	Reserved range end

3.1.5 ST_IolIdentificationObjects

Identification objects as a structure output by the function block IOL_IdentificationAndDiagnosis [▶ 29].

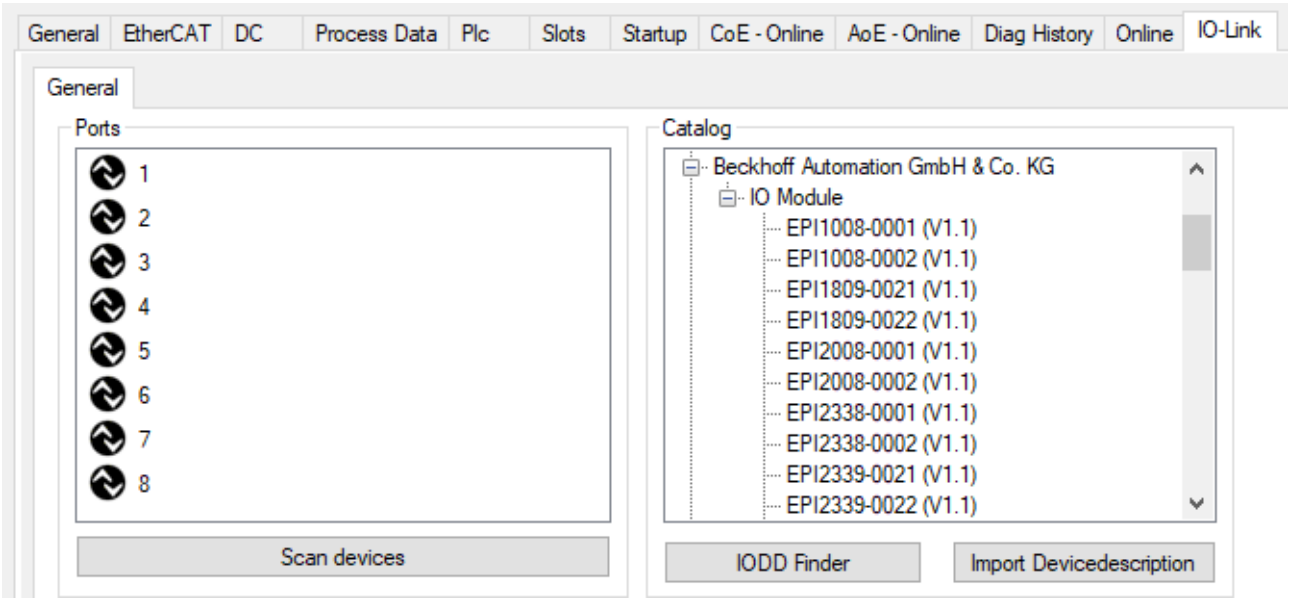
Name	Data type
nVendorID	WORD
nDeviceID	DWORD
sVendorname	STRING(64)
sVendorText	STRING(64)
sProductName	STRING(64)
sProductID	STRING(64)
sProductText	STRING(64)
sSerialNumber	STRING(16)
sHardwareRevision	STRING(64)
sFirmwareRevision	STRING(64)
sApplicationSpecificTag	STRING(64)
sLocationTag	STRING(64)
sFunctionTag	STRING(64)

3.2 Devices

Can be used in conjunction with the state channel to determine the state of communication.

3.2.1 E_IolPort

Specifies the number of the port via an enumeration.



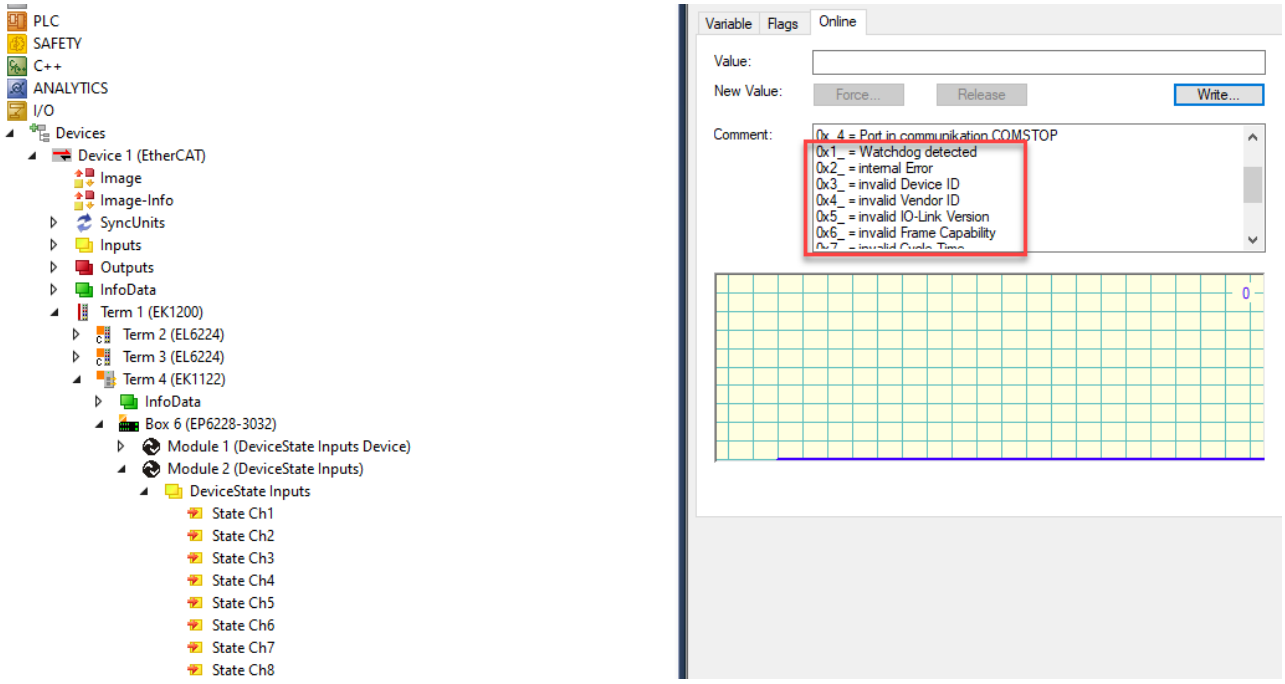
Name	Value
Port1	1
Port2	2
Port3	3
Port4	4
Port5	5
Port6	6
Port7	7
Port8	8

Sample

```
fbIolSlave.Port:=E_IolPort.Port1;
```

3.2.2 E_IolPortError

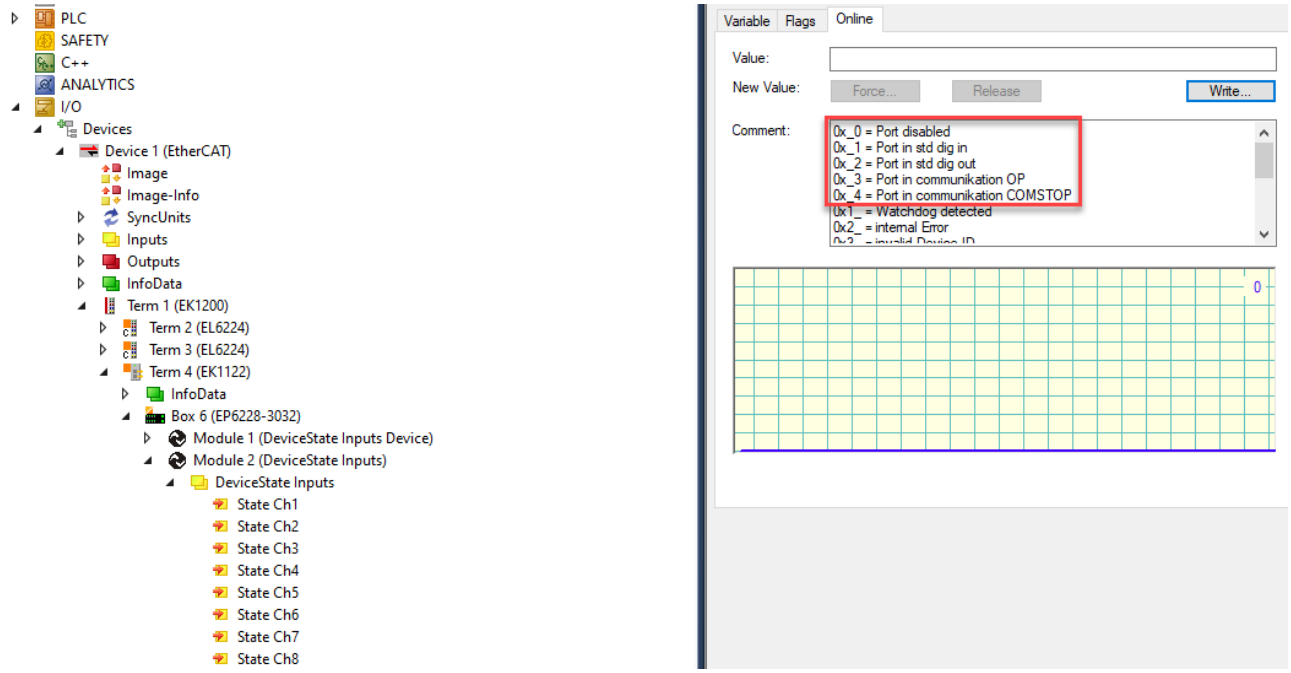
Outputs the errors of the IO-Link communication of the state channel.



Name	Value
NoError	0
WatchdogDetected	1
InternalError	2
InvalidDeviceID	3
InvalidVendorID	4
InvalidIOLinkVersion	5
InvalidFrameCapability	6
InvalidCycletime	7
InvalidPdInLength	8
InvalidPdOutLenght	9
NoDeviceDetected	10

3.2.3 E_IolPortState

Information about the communication type of the state channel



Name	Value
Disabled	0
stDigIn	1
stDigOut	2
CommunicationOP	3
CommunicationComstop	4

3.3 Specification

3.3.1 E_IolDeviceStatus

Indicates the state of the IO-Link device and comes from the index 0x0024 [▶ 70].

Name	Value	Description
Operating	0x00	IO-Link device works properly
MaintenanceRequired	0x01	Although the process data is valid, the internal diagnosis indicate that the device is about to lose its ability to function correctly.
OutOfSpecification	0x02	Although the process data is valid, the internal diagnosis indicate that the device is operating outside the specified measuring range or environmental conditions.
FunctionalCheck	0x03	Process data is temporarily invalid due to intentional manipulation of the device
Failure	0x04	Process data invalid due to a malfunction of the device or its peripheral devices. The device is not able to perform its intended function.

3.3.2 E_IolError

Contains fixed IO-Link error codes [▶ 74].

Name	Value	Description
NoError	0x0000	No error message
DeviceApplicationError	0x8000	Device-specific application error

Name	Value	Description
IndexNotAvailable	0x8011	The index is not available
SubindexNotAvailable	0x8012	The subindex is not available
ServiceTemporarilyNotAvailable	0x8020	A parameter for a read or write service is not accessible because the current state of the device application is not accessible.
ServiceTemporarilyNotAvailable_LocalControl	0x8021	A parameter is not accessible to a read or write service due to an ongoing local operation on the device (e.g. operation or parameterization via an on-board device control panel).
ServiceTemporarilyNotAvailable_DeviceControl	0x8022	A read or write service is not accessible due to a remotely triggered state of the device application (e.g. parameterization during a remotely triggered learning process or calibration).
AccessDenied	0x8023	A write service is trying to access a read-only parameter. Or a read service tries to access a read-only parameter.
ParameterValueOutOfRange	0x8030	A write service to a parameter that is outside its allowed value range
ParameterValueAboveLimit	0x8031	A write service accesses a parameter above the specified value range.
ParameterValueBelowLimit	0x8032	A write service accesses a parameter below the specified value range.
ParameterLengthOverrun	0x8033	The content of a write service for a parameter is greater than the length specified for the parameter. Or a data object is too large to be processed by the device application (e.g. ISDU buffer limitation).
ParameterLengthUnderrun	0x8034	The content of a write service for a parameter is smaller than the length specified for the parameter. Or a data object is too small to be processed by the device application (e.g. ISDU buffer limitation).
FunctionNotAvailable	0x8035	A write service that is not supported by the device application (for example, a system command with an unimplemented value).
FunctionTemporarilyUnavailable	0x8036	A write service that calls a device function that is not available due to the current state of the device application (e.g. System Command).
InvalidParameterSet	0x8040	the single parameter transmission does not match other actual parameter settings (e.g. overlapping setpoints for a binary data setting).
InconsistentParameterSet	0x8041	At the end of a block parameter transfer with ParamDownloadEnd or ParamDownloadStore if the plausibility check reveals inconsistencies
ApplicationNoReady	0x8082	When a read or write service is denied due to a temporarily unavailable application (for example, peripheral controls during startup).

3.3.3 E_IolSystemcommand

Specifies the values for the System Command.

Name	Value	Description
ParamUploadStart	0x01	Starts parameter upload
ParamUploadEnd	0x02	Stops parameter upload
ParamDownloadStart	0x03	Starts parameter download
ParamDownloadEnd	0x04	Stops parameter download
ParamDownloadStore	0x05	Finalizes the parameterization and starts the data storage
DevcieReset	0x80	Reset to initialization value

Name	Value	Description
ApplicationReset	0x81	Application-specific parameters are reset
RestoreFactorySettings	0x82	Factory settings

3.3.4 ST_IolAccessLocks

Structure specifies the AcessLocks. The AcessLocks determine which access is physically permitted by the IO-Link device.

Name	Data type
bParameterAcess	BIT
bDataStorage	BIT
bLocalParametrization	BIT
bLocalUserInterface	BIT
bReserved1	BIT
bReserved2	BIT
bReserved3	BIT
bReserved4	BIT
nReserved1	BYTE

3.3.5 ST_IolDetailedDeviceStatusEntry

Structure of the Detailed Device Status Entry

Name	Data type
nEventqualifier	BYTE
nEventCode	WORD

3.3.6 ST_IolDirectParameter1

Defines the DirectParameter1 as a structure.

Name	Data type
Reserved1	BYTE
MasterCycleTime	BYTE
MinCycleTime	BYTE
FrameCapability	BYTE
VersionID	BYTE
ProcessDataInLen	BYTE
ProcessDataOutLen	BYTE
VendorID1	BYTE
VendorID2	BYTE
DeviceId1	BYTE
DeviceID2	BYTE
DeviceID3	BYTE
Reserved2	BYTE
Reserved3	BYTE
Reserved4	BYTE
Reserved5	BYTE

3.3.7 ST_IolDirectParameter2

Structure defines the DirectParameter2

Name	Data type
DeviceSpecificParameter01	BYTE
DeviceSpecificParameter02	BYTE
DeviceSpecificParameter03	BYTE
DeviceSpecificParameter04	BYTE
DeviceSpecificParameter05	BYTE
DeviceSpecificParameter06	BYTE
DeviceSpecificParameter07	BYTE
DeviceSpecificParameter08	BYTE
DeviceSpecificParameter09	BYTE
DeviceSpecificParameter10	BYTE
DeviceSpecificParameter11	BYTE
DeviceSpecificParameter12	BYTE
DeviceSpecificParameter13	BYTE
DeviceSpecificParameter14	BYTE
DeviceSpecificParameter15	BYTE
DeviceSpecificParameter16	BYTE

3.3.8 T_IolDetailedDeviceStatus

Event history of the IO-Link device

```
TYPE T_IolDetailedDeviceStatus :
ARRAY[0..63] OF ST_IolDetailedDeviceStatusEntry;
END_TYPE
```

3.3.9 T_IolProfileCharacteristics

Profile characteristics of the IO-Link device

```
TYPE T_IolProfileCharacteristics :
ARRAY[0..31] OF WORD;
END_TYPE
```

3.3.10 T_IolMaxOctedString

Definition of the maximum string size

```
TYPE T_IolMaxOctedString :
ARRAY[0..231] OF BYTE;
END_TYPE
```

3.3.11 T_IolMaxString

Defines the maximum size of an IO-Link string.

```
TYPE T_IolMaxString : STRING(232);
END_TYPE
```

3.3.12 ST_IolIndex

Specifies the index and subindex at the IO-Link parameter.

Name	Data type
nIndex	WORD

Name	Data type
nSubindex	BYTE

3.4 Smart Sensor Profile

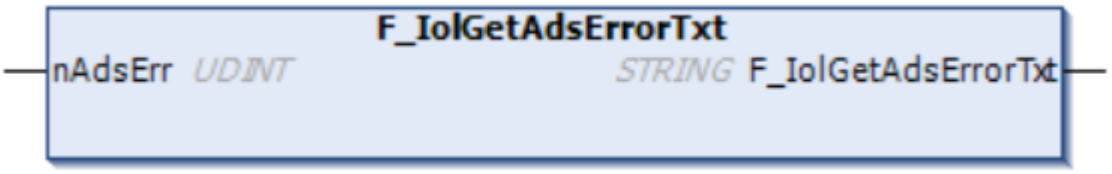
3.4.1 E_IoIMdc_ValueStatus

Defines the status of the process data

Name	Value	Description
OK	0	Process data are valid
PDInvalid	1	Process data are invalid
NoData	2	No data
OutOfRangePos	3	Process data are outside the value range
OutOfRangeNeg	4	Process data are outside the value range
StateNotDefined	5	Undefined state

4 Functions

4.1 F_IolGetAdsErrorTxt



Function converts an ADS error into text form.

Inputs

Name	Type	Description
nAdsErr	UDINT	ADS Return Codes

Outputs

Name	Type	Description
F_IolGetAdsErrorTxt	STRING	Displays the ADS return code as text string.

Sample

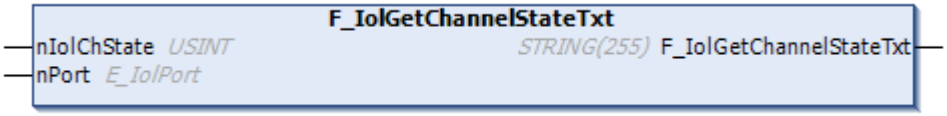
```
VAR
  hr             :HRESULT;
  nADSError      :E_AdsErr := E_AdsErr.NOERR;
  sADSErrorText  :STRING;
END_VAR
```

```
nAdsError := F_IolGetAdsError(hr);
sADSErrorText := F_IolGetAdsErrorTxt(nAdsError);
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

4.2 F_IolGetChannelStateTxt



Function converts state channel and port into text form.

Inputs

Name	Type	Description
nIolChState	USINT	IO-Link Channel State
nPort	E_IolPort [▶ 12]	IO-Link port

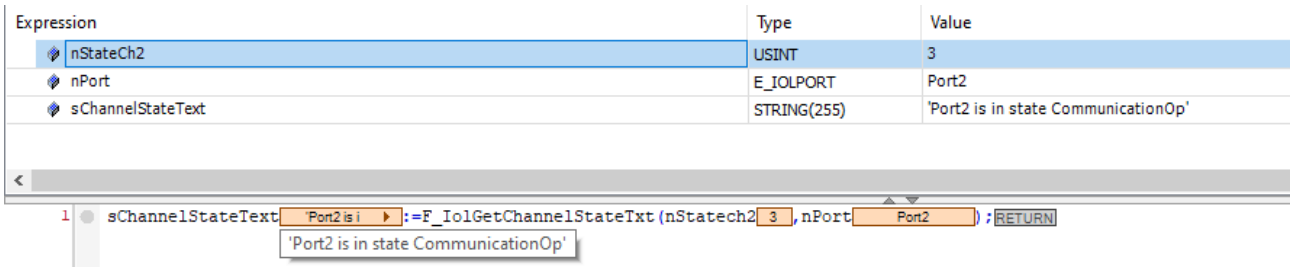
Outputs

Name	Type	Description
F_IolGetChannelStateTxt	String(255)	Converts the channel state into text information.

Example

```
VAR
  nStateCh2  AT%I*    :USINT;
  nPort               :E_IolPort:=E_iolPort.Port2;
  sChannelStateText   :STRING(255);
END_VAR
```

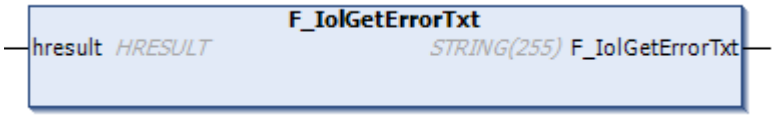
```
sChannelStateText := F_IolGetChannelStateTxt(nStateCh2,nPort);
```



Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

4.3 F_IolGetErrorTxt



Returns the IO-Link error as string format in text form.

Inputs

Name	Type	Description
hresult	HRESULT	Return value

Outputs

Name	Type	Description
F_IolGetErrorTxt	String(255)	Shows the IO-Link error as a string

Example

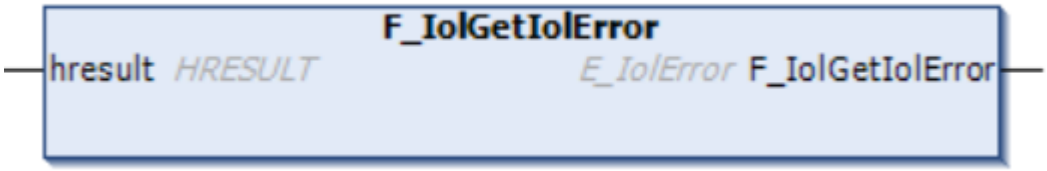
```
VAR
  hr          :HRESULT;
  sErrorText  :STRING;
END_VAR
```

```
sErrorText := F_IolGetErrorTxt(hr);
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

4.4 F_IolGetIolError



Function returns an IO-Link error code as E_IolError [▶ 14].

Inputs

Name	Type	Description
hresult	HRESULT	Return value

Outputs

Name	Type	Description
F_IolGetIolError	E_IolError [▶ 14]	IO-Link error code as enum

Sample

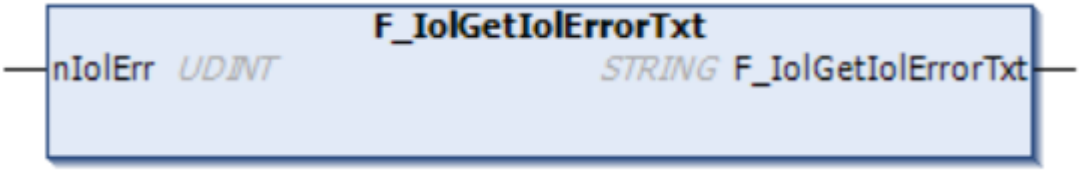
```
VAR
  hr          :HRESULT;
  nIOLError   :E_IolErr := E_IolErr.NOERR;
END_VAR
```

```
nIOLError := F_IolGetIolError(hr);
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

4.5 F_IolGetIolErrorTxt



The function returns the IO-Link error code as a string

Inputs

Name	Type	Description
nIolErr	UDINT	IO-Link error

Outputs

Name	Type	Description
F_IolGetIolErrorTxt	String	IO-Link error code as text

Sample

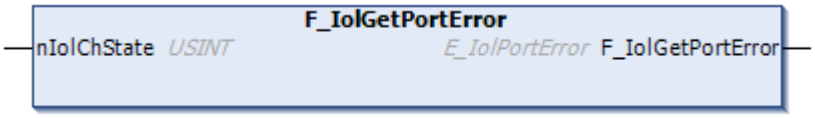
```
VAR
  hr            :HRESULT;
  nIolError     :E_IolErr := E_IolErr.NOERR;
  sIolErrorText :STRING;
END_VAR
```

```
nIolError := F_IolGetIolError(hr);
sIolErrorText := F_IolGetIolErrorTxt(nIolError);
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

4.6 F_IolGetPortError



Function returns the state channel as E_IolPortError.

Inputs

Name	Type	Description
nIolChState	USINT	IO-Link Channel State

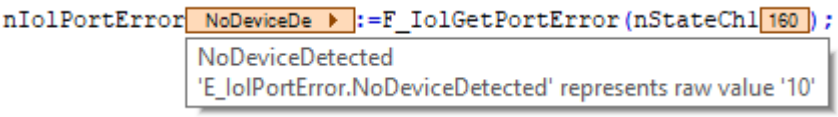
Outputs

Name	Type	Description
E_IolGetPortError	E_IolPortError [▶ 12]	Display IO-Link port error

Example

```
VAR
  nStateCh1  AT%I*    :USINT;
  nIolPortError       :E_IolPortError;
END_VAR
```

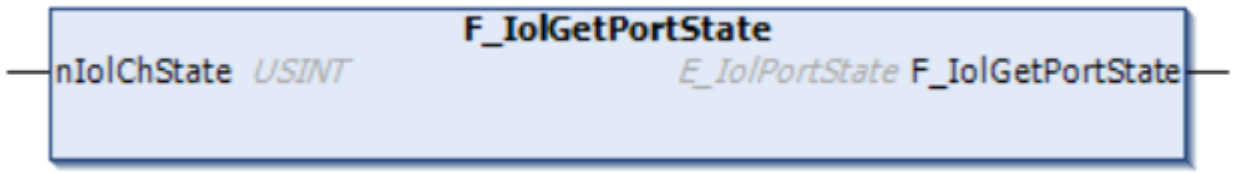
```
nIolPortError := F_IolGetPortError(nStateCh1);
```



Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

4.7 F_IolGetPortState



Function converts the variable State Ch into the E_IolPortState.

The data type E_IolPortstate can be further used afterwards.

Inputs

Name	Type	Description
nIolChState	USINT	IO-Link channel state

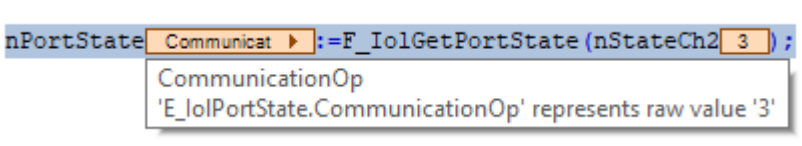
Outputs

Name	Type	Description
E_IoLGetPortState	E_IolPortState [▶ 13]	Displays the port state as an enum

Sample

```
VAR
  nStateCh2  AT%I*     :USINT;
  nPortState           :E_IolPortState;
END_VAR
```

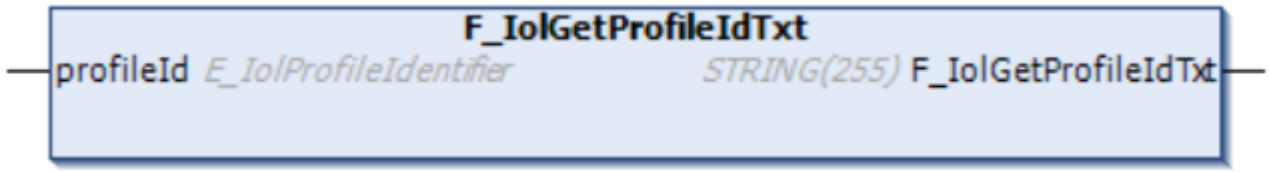
```
nPortState := F_IolGetPortState(nStateCh2);
```



Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

4.8 F_IolGetProfileIdTxt



Converts the E_IolProfileIdentifier type to a text format.

Inputs

Name	Type	Description
profileId	E_IolProfileIdentifier [▶ 10]	IO-Link profiles

Outputs

Name	Type	Description
F_IoLGetProfileIdTxt	String(255)	IO-Link profiles as text form

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

4.9 F_IolIsBusy

Function indicates whether a function block is still being processed.

Inputs

Name	Type	Description
hresult	HRESULT	Return value

Outputs

Name	Type	Description
F_IolIsBusy	BOOL	True if function block is still being processed

Sample

```
VAR
  hr                :HRESULT;
  fbIIolStdVarSlave :FB_IolStdVarSlave;
  nRead             :DWORD;
END_VAR
```

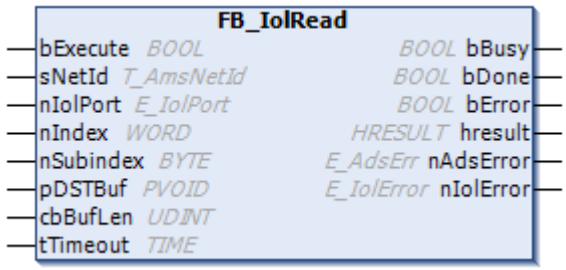
```
hr := fbIIolStdVarSlave.read(16#0019, 0, ADR(nRead), SIZEOF(nRead));
IF NOT F_IolIsBusy(hr) THEN
  IF FAILED(hr) THEN
    //Error Handling
  ELSE
    //Success
  END_IF
END_IF
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

5 Function blocks

5.1 FB_IolRead



Function block for reading IO-Link parameters in a non-object-oriented solution.

Inputs

Name	Type	Description
bExecute	BOOL	Rising edge to start the function block
sNetid	T-AmsNetid	AMS NETID of the IO-Link master
nIolPort	E_IolPort [▶ 12]	IO-Link port of the IO-Link device
nIndex	WORD	Index of the IO-Link device parameter
nSubindex	BYTE	Subindex of the IO-Link device parameter
pDSTBuf	PVOID	Pointer to target buffer
cbBufLen	UDINT	Size of the data bytes to be read
tTimeout	TIME	Maximum allowed execution time

Outputs

Name	Type	Description
bBusy	BOOL	Busy flag of the function block
bDone	BOOL	Done flag of the function block
bError	BOOL	Error flag of the function block
hResult	HRESULT	Return value of this function block
nADSError	E_AdsErr	ADS error code
nIolError	E_IolError [▶ 14]	IO-Link error code

Sample

The following function block is part of the IoLink sample project. For more information, see the chapter Samples [▶ 69].

```
FUNCTION_BLOCK FB_Sample_01_IolRead
VAR_INPUT
  bExecute        :BOOL := FALSE;
END_VAR
```

```
VAR_OUTPUT
  bBusy             :BOOL := FALSE;
  bDone             :BOOL := FALSE;
  bError            :BOOL := FALSE;
  nAdsError         :E_AdsErr := E_AdsErr.NOERR;
  nIolError         :E_IolError := E_IolError.NoError;
  sErrorTxt         :STRING(255) := '';
  sApplicationTag   :STRING(32) := '';
END_VAR
```

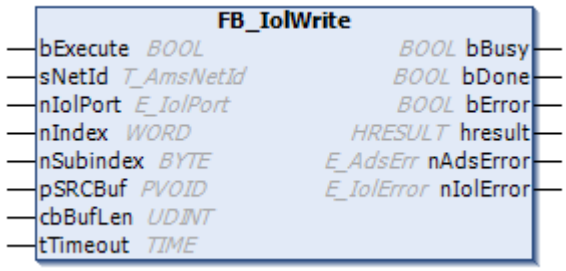
```
VAR
  bExecuteOld       :BOOL;
  nState            :INT := 0;
  fbIolRead         :FB_IolRead;
END_VAR
```

```
////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
  bBusy := TRUE;
  nState := 10;     // READ
ELSIF NOT bExecute THEN
  bBusy := FALSE;
  bError := FALSE;
  bDone := FALSE;
  nAdsError := E_AdsErr.NOERR;
  nIolError := E_IolError.NoError;
  sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF
// IDLE
0:;
// READ
10:
  sApplicationTag := '';
  fbIolRead(bExecute := FALSE);
  fbIolRead.sNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);
  fbIolRead.nIolPort := E_IolPort.Port1;
  fbIolRead.pDSTBuf := ADR(sApplicationTag);
  fbIolRead.cbBufLen := SIZEOF(sApplicationTag);
  fbIolRead.nIndex := 16#18;
  fbIolRead.nSubindex := 0;
  fbIolRead.bExecute := TRUE;
  nState := 20;    // WAIT_FOR_DONE

// WAIT_FOR_DONE
20:
  IF fbIolRead.bDone THEN
    bDone := TRUE;
    bBusy := FALSE;
    nState := 0; // IDLE
  END_IF
END_CASE

////
// call FBs
//
fbIolRead();

////
// Error Handling
//
IF NOT bError AND bBusy THEN
  IF fbIolRead.bError THEN
    nState := 0;
    bBusy := FALSE;
    bError := TRUE;
    nAdsError := fbIolRead.nAdsError;
    nIolError := fbIolRead.nIolError;
    sErrorTxt := F_IolGetErrorTxt(fbIolRead.hresult);
  END_IF
END_IF
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

5.2 FB_IolWrite



Function block for writing IO-Link parameters in a non-object-oriented solution.

Inputs

Name	Type	Description
bExecute	BOOL	Rising edge to start the function block
sNetid	T-AmsNetid	AMS NETID of the IO-Link master
nIolPort	E_IolPort [▶ 12]	IO-Link port of the IO-Link device
nIndex	WORD	Index of the IO-Link device parameter
nSubindex	BYTE	Subindex of the IO-Link device parameter
pSCRBuf	PVOID	Pointer to source buffer
cbBufLen	UDINT	Size of the data bytes to be read
tTimeout	TIME	Maximum allowed execution time

Outputs

Name	Type	Description
bBusy	BOOL	Busy flag of the function block
bDone	BOOL	Done flag of the function block
bError	BOOL	Error flag of the function block
hResult	HRESULT	Return value of this function block
nADSError	E_AdsErr	ADS error code
nIolError	E_IolError [▶ 14]	IO-Link error code

Sample

The following function block is part of the IoLink sample project. For more information, see the chapter Samples [▶ 69].

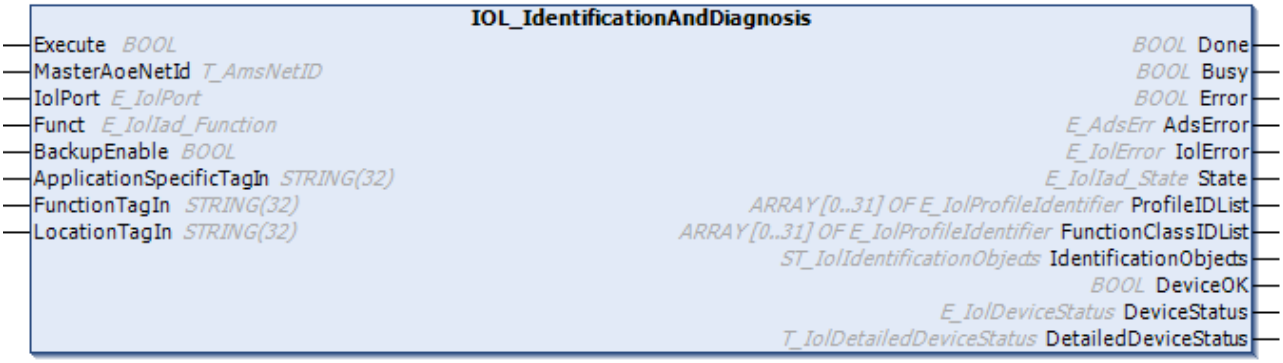
```
FUNCTION_BLOCK FB_Sample_02_IolWrite
VAR_INPUT
    bExecute                : BOOL := FALSE;
    sApplicationTag         : STRING(32) := '';
END_VAR
VAR_OUTPUT
    bBusy                   : BOOL := FALSE;
    bDone                   : BOOL := FALSE;
    bError                  : BOOL := FALSE;
    nAdsError               : E_AdsErr := E_AdsErr.NOERR;
    nIolError               : E_IolError := E_IolError.NoError;
    sErrorTxt               : STRING(255) := '';
END_VAR
VAR
    bExecuteOld             : BOOL;
    nState                  : INT := 0;
    fbIolWrite              : FB_IolWrite;
END_VAR
```

```
////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
    bBusy := TRUE;
    nState := 10;    // WRITE
ELSIF NOT bExecute THEN
    bBusy := FALSE;
    bError := FALSE;
    bDone := FALSE;
    nAdsError := E_AdsErr.NOERR;
    nIolError := E_IolError.NoError;
    sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF

// IDLE
0:;

// WRITE
10:
    sApplicationTag := '';

    fbIolWrite(bExecute := FALSE);
    fbIolWrite.sNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);
    fbIolWrite.nIolPort := GVL_IO.nTerm2EPI1008;
    fbIolWrite.pSRCBuf := ADR(sApplicationTag);
    fbIolWrite.cbBufLen := INT_TO_UDINT(LEN(sApplicationTag));
    fbIolWrite.nIndex := 16#18;
    fbIolWrite.nSubindex := 0;
    fbIolWrite.bExecute := TRUE;

    nState := 20; // WAIT_FOR_DONE

// WAIT_FOR_DONE
20:
    IF fbIolWrite.bDone THEN
        bDone := TRUE;
        bBusy := FALSE;
        nState := 0; // IDLE
    END_IF
END_CASE

////
// call FBs
//
fbIolWrite();

////
// Error Handling
//
IF NOT bError AND bBusy THEN
    IF fbIolWrite.bError THEN
        nState := 0;
        bBusy := FALSE;
        bError := TRUE;
        nAdsError := fbIolWrite.nAdsError;
        nIolError := fbIolWrite.nIolError;
        sErrorTxt := F_IolGetErrorTxt(fbIolWrite.hresult);
    END_IF
END_IF
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

5.3 IOL_IdentificationAndDiagnosis



Function block for identification and diagnosis of an IO-Link device. This function block is implemented by the Common IO-Link Profile. The ApplicationSpecificTag, the FunctionTag and the LocationTag of an IO-Link device can be written. The function classes and ProfileIDs of a device can be read. Additionally the function block gives a diagnosis on the state of a device and its identification, which can include the Vendor Name, Vendor ID, Device Name, and Device ID.

Inputs

Name	Type	Description
Execute	BOOL	A trigger causes the function selected with the Function variable to be executed:
MasterAoENetid	T_AmsNetid	AMS NETID of the IO-Link master
IolPort	E_IolPort [▶ 12]	IO-Link port of the IO-Link device
Func	E_IolIad_Function [▶ 9]	This variable selects the functionality to be triggered by a request: 0 = no_func A request is neglected, no function is executed 1 = rd_all A request starts the read-back of the current identification and diagnostic parameter values from the device. 2 = rd_diag A request starts reading back the current diagnostic parameter values by reading DeviceStatus and DetailedDeviceStatus from the device. 3 = wr_ ident A request causes a previously created value for ApplicationSpecificTagIn, LocationTagIn and FunctionTagIn to be written to the device.
BackupEnable	BOOL	This variable configures the behavior of the FB in case of the requested function wr_ident. "true" = enabled The backup mechanism is triggered by the FB. "false" = disabled The backup mechanism is not triggered by the FB.
ApplicationSpecificIN	String(32)	Parameters of the IO-Link device
FunctionTagIN	String(32)	Parameters of the IO-Link device
LoactionTagIN	String(32)	Parameters of the IO-Link device

Outputs

Name	Type	Description
Done	BOOL	The signal is set when the FB has completed a requested operation.
Busy	BOOL	The signal is set when the FB executes a requested operation.
Error	BOOL	The signal is set when an error occurred during the execution of a requested operation.
AdsError	E_ADSErr	ADS error code
IoLError	E_IolError [▶ 14]	IO-Link error
State	E_Iollad_State [▶ 9]	The value represents the current status of the FB operation and the executed functions. The content is system specific and contains the status information.
ProfileIDList	ARRAY [0..31] of E_IolProfileIdentifier [▶ 10]	List of ProfileIDs supported by the device
FunctionClassIDList	ARRAY [0..31] of E_IolProfileIdentifier [▶ 10]	List of FunctionClassIDs supported by the device
IdentificationObjects	ST_IolIdentificationObjects [▶ 11]	Structured list of identification objects, for more details.
DeviceOK	BOOL	The signal is set if no further diagnostic information is available, it is false if further information is available at DeviceStatus and DetailedDeviceStatus.
DeviceStatus	E_IolDeviceStatus [▶ 14]	Status of the IO-Link device
DetailedDeviceStatus	T_IolDetailedDeviceStatus [▶ 17]	ARRAY [0..63] of ST_DetailedDeviceStatusEntry

Example

The following function block is part of the IoLink sample project. For more information, see the chapter Samples [▶ 69].

```
FUNCTION_BLOCK FB_Sample_06_IdentificationAndDiagnosis
VAR_INPUT
    sApplicationTagIn       : STRING(32) := '';
    sFunctionTagIn          : STRING(32) := '';
    sLocationTagIn          : STRING(32) := '';
    bExecute                : BOOL;
END_VAR
VAR_OUTPUT
    bBusy                   : BOOL := FALSE;
    bDone                   : BOOL := FALSE;
    bError                  : BOOL := FALSE;

    sIolError               : STRING;
    sAdsError               : STRING;
    aProfileIDList          : ARRAY [0..31] OF E_IolProfileIdentifier;
    aFunctionClassIDList    : ARRAY[0..31] OF E_IolProfileIdentifier;
    stIDentObjects          : ST_IolIdentificationObjects;
    bDeviceOK               : BOOL;
    nDeviceStatus           : E_IolDeviceStatus;
    DetailedDeviceStatus    : T_IolDetailedDeviceStatus;
END_VAR
VAR
    bExecuteOld             : BOOL;
    nState                  : INT := 0;
    IolDiagAndIdent         : IOL_IdentificationAndDiagnosis;
END_VAR
////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
    bBusy := TRUE;
    nState := 10;    // START
ELSIF NOT bExecute THEN
```

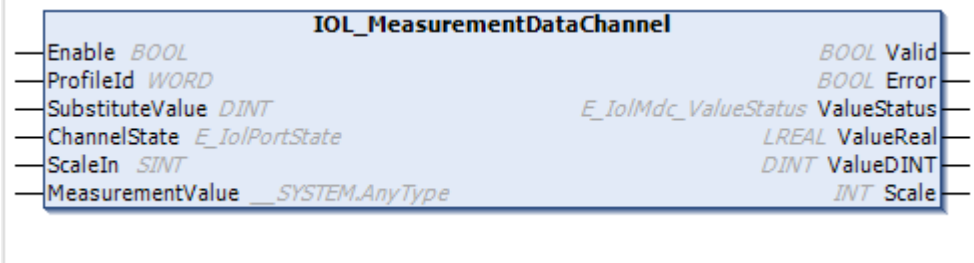
```
    bBusy := FALSE;
    bError := FALSE;
    bDone := FALSE;
    sIolError:='';
    sADSError:='';
END_IF
bExecuteOld := bExecute;


CASE nState OF

// IDLE
0:;

// START
10:
    IolDiagAndIdent(Execute := FALSE);

    IolDiagAndIdent.MasterAoeNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);
    IolDiagAndIdent.IolPort := E_iolport.Port4;
    IolDiagAndIdent.Funct := E_IolIad_Function.ReadAll;
    IolDiagAndIdent.ApplicationSpecificTagIn := sApplicationTagIn;
    IolDiagAndIdent.FunctionTagIn := sFunctionTagIn;
    IolDiagAndIdent.LocationTagIn := sLocationTagIn;

    IolDiagAndIdent.Execute := TRUE;
    nState := 20; // WAIT FOR DONE

// WAIT FOR DONE
20:
    IF NOT IOlDiagAndIdent.Busy AND NOT IOlDiagAndIdent.Error THEN
        nState := 30; // DONE
    END_IF

// DONE
30:
    aProfileIDList := IolDiagAndIdent.ProfileIDList;
    aFunctionClassIDList := IolDiagAndIdent.FunctionClassIDList;
    stIDentObjects := IolDiagAndIdent.IdentificationObjects;
    bDeviceOK := IolDiagAndIdent.DeviceOK;
    nDeviceStatus := IolDiagAndIdent.DeviceStatus;
    DetailedDeviceStatus := IolDiagAndIdent.DetailedDeviceStatus;

    IolDiagAndIdent.Execute := FALSE;
    nState := 0; // INIT

END_CASE


////
// call FBs
//
```

```
IolDiagAndIdent();

////
// Error Handling
//
IF NOT bError AND bBusy THEN
    IF IolDiagAndIdent.Error THEN
        nState := 0;
        bBusy := FALSE;
        bError := TRUE;
        sIolError := F_IolGetIolErrorTxt(ioldiagandIdent.IolError);
        sAdsError := F_IolGetadsErrorTxt(ioldiagandident.AdsError);
    END_IF
END_IF
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

5.4 IOL_MeasurementDataChannel



The measuring data channel defines the process data structure, functions, and presentation of measuring sensors. The function block is defined to provide derived status signals and enable a standardized interface for user programs.

Inputs

Name	Type	Description
Enable	BOOL	Activates the use of the function block
ProfileID	WORD	Selected ProfileID, or process data layout 1 = SSP 3.1 = profile 0x000A 2 = SSP 3.2 = profile 0x000B 3 = SSP 3.3 = profile 0x000C 4 = SSP 3.4 = profile 0x000D
SubstituteValue	DINT	The specified value is applied to ValueReal and ValueDINT if ValueStatus is not 0.
ChannelState	E_IolPortState [▶ 13]	
ScaleIN	SINT	Scale of the IO-Link device
ProcessData	ANY_INT	Measured value

Outputs

Name	Type	Description
Valid	BOOL	If true, the supplied values are valid and can be used for further calculations.
Error	BOOL	If true, an internal error is detected and further information is provided by the function block.
ValueStatus	E_IoLMDC_ValueStatus [▶ 18]	Provides error codes
ValueReal	LREAL	Process data in real format for evaluation in the PLC
ValueINT	DINT	Process data in Dint format for evaluation in the PLC
Scale	INT	Process data scaling factor

Example

The following function block is part of the IoLink sample project. For more information, see the chapter Samples [▶ 69].

```
FUNCTION_BLOCK FB_Sample_07_MeasurementDataChannel
VAR_INPUT
  nStateChannel    AT%I*:USINT;
  nScale           AT%I*:SINT;
  MVDistance       AT%I*:INT;
END_VAR
VAR_OUTPUT
  bValid          :BOOL;
  bError          :BOOL;
  nValueStatus    :E_IolMdc_ValueStatus;
  nValueDINT      :DINT;
  nValueREAL      :REAL;
```

```
  nScaleOut       :INT;
END_VAR
VAR
  fbMeasurement:IOL_MeasurementDataChannel_INT;
END_VAR
```

```
fbMeasurement(
  Enable := TRUE,
  SubstituteValue := 500,
  ChannelState := nStateChannel,
  ScaleIn := nScale ,
  MeasurementValue := MVDistance,
  Valid=>bValid,
  Error=>bError,
  ValueStatus=>nValueStatus,
  ValueReal=>nValueREAL,
  ValueDINT=>nValueDINT
  Scale=>nScaleOut );
```

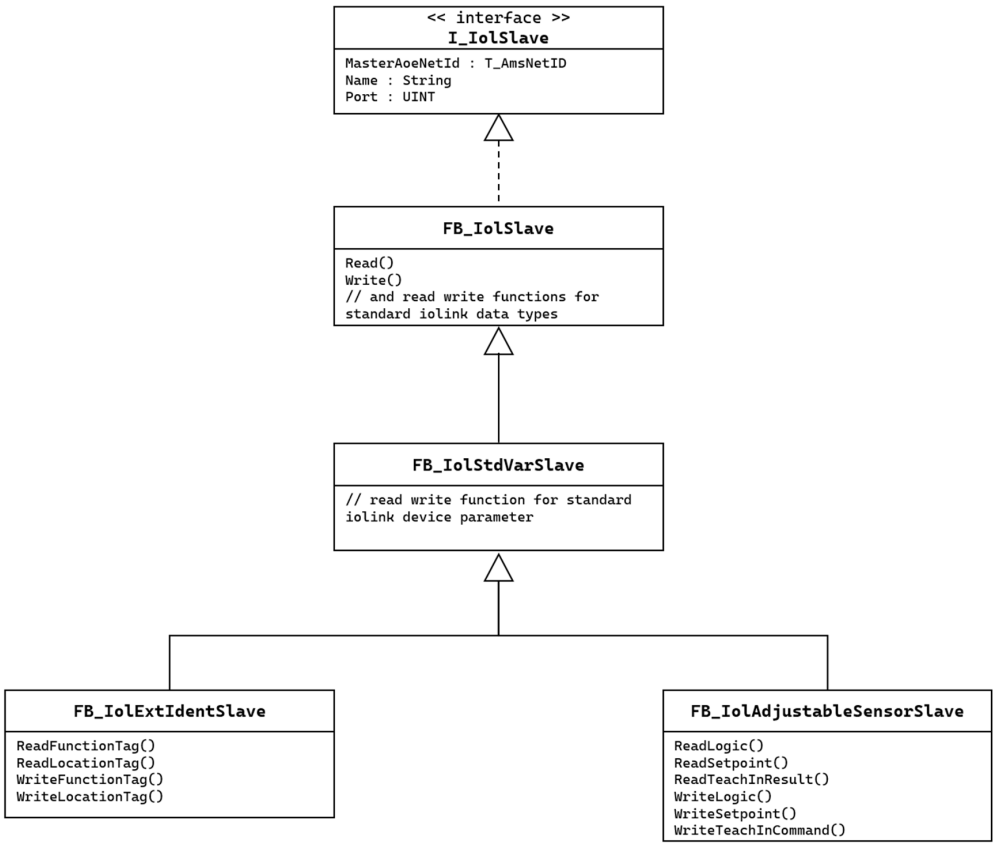
Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6 Object-oriented function blocks

The object-oriented function blocks are listed here. A direct call of the function block is no longer necessary. Instead, you configure the properties and call the methods directly.

The following UML diagram shows the structure and inheritance.



6.1 I_IolSlave

The interface defines all important parameters to communicate acyclically with an IO-Link device.

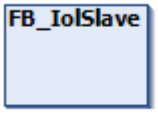
Interfaces

Type	Description
I_IolSlave	Defines IO-Link device

Properties

Name	Type	Access	Definition location	Initial value	Description
MasterAoENetid	T_AmsNetid	Get	Local		AMS Netid from IO-Link master
Name	String(255)	Get, Set	Local		Name
Port	UINT	Get, Set	Local		Port of the IO-Link device

6.2 FB_IolSlave



Function block for the readout of IO-Link parameters.

Sample

```
VAR
  bInitDone                : BOOL;
  sTerm3AmsNetId  AT%I*    : AMSNETID;
  nTerm2EPI2328            : E_IoLPort := E_IolPort.Port2;
  fbIOLSlave               : FB_IolSlave;
END_VAR

IF NOT bInitDone THEN
  bInitDone                 := TRUE;                    //Assigns it only once
  fbIolSlave.MasterAoeNetId := F_CreateAmsNetId(sTerm3AmsNetId)   //Creates NETID String
  fbIolSlave.Port           := nTerm2EPI2328;           //Assigns Port
  fbIolSlave.Name           := 'EPI2328';               //Gives device a name
END_IF
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.3.0

6.2.1 Methods

6.2.1.1 READ

Reads out a parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
pDSTBuf	PVOID	Address of the buffer which is to receive the data that has been read. The programmer is himself responsible for dimensioning the buffer to a size that can accept LEN bytes. The buffer can be a single variable, an array or a structure, whose address can be determined with the ADR operator.
cbBufLen	UDINT	Number, in bytes, of the data to be read.

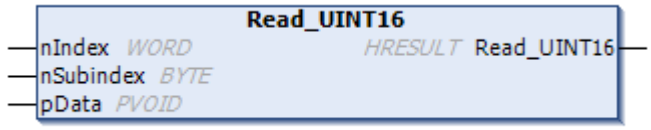
Outputs

Name	Type	Description
READ	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.2 READ_UINT16



Reads a UINT16 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link parameter 0 for the entire structure
nSubindex	BYTE	Subindex of the IO-Link parameter 0 for the entire structure
pData	PVOID	Address of the buffer that is to receive the read data. The programmer himself is responsible for dimensioning the buffer to a size that can hold LEN bytes. The buffer can be a single variable, an array or a structure, whose address can be found with the ADR operator.

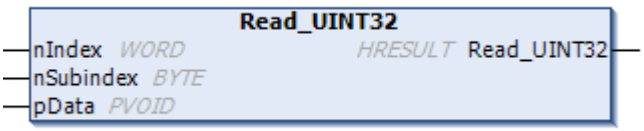
Outputs

Name	Type	Description
Read_Uint16	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.3 READ_UINT32



Reads a UINT32 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link parameter 0 for the entire structure
nSubindex	BYTE	Subindex of the IO-Link parameter 0 for the entire structure
pData	PVOID	Address of the buffer that is to receive the read data. The programmer himself is responsible for dimensioning the buffer to a size that can hold LEN bytes. The buffer can be a single variable, an array or a structure, whose address can be found with the ADR operator.

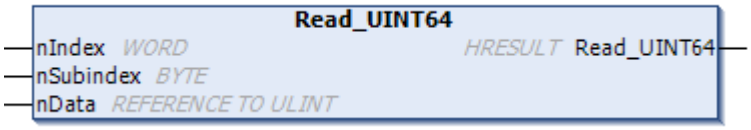
Outputs

Name	Type	Description
Read_Uint32	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.4 READ_UINT64



Reads a UINT64 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link parameter
nSubindex	BYTE	Subindex of the IO-Link parameter 0 for the entire structure
nData	Reference to ULINT	Reference to the data type ULINT

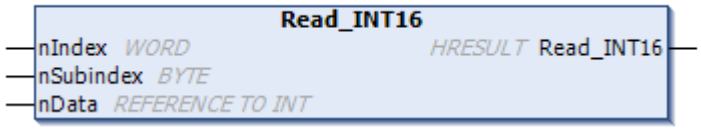
Outputs

Name	Type	Description
Read_UINT64	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.5 READ_INT16



Reads an INT16 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link parameter
nSubindex	BYTE	Subindex of the IO-Link parameter 0 for the entire structure
nData	Reference to INT	Reference to the data type INT

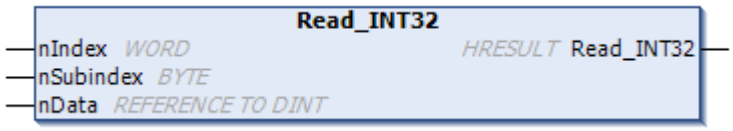
Outputs

Name	Type	Description
Read_INT16	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.6 READ_INT32



Reads a UINT32 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link parameter
nSubindex	BYTE	Subindex of the IO-Link parameter 0 for the entire structure
nData	Reference to DINT	Reference to the data type DINT

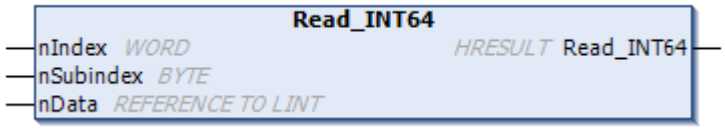
Outputs

Name	Type	Description
Read_INT32	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.7 READ_INT64



Reads an INT64 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link parameter
nSubindex	BYTE	Subindex of the IO-Link parameter 0 for the entire structure
nData	Reference to LINT	Reference to the data type LINT

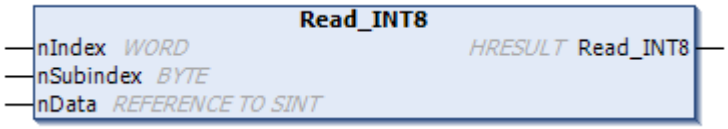
Outputs

Name	Type	Description
Read_INT64	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.8 READ_INT8



Reads a UINT8 parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link parameter
nSubindex	BYTE	Subindex of the IO-Link parameter 0 for the entire structure
nData	Reference to SINT	Reference to the data type SINT

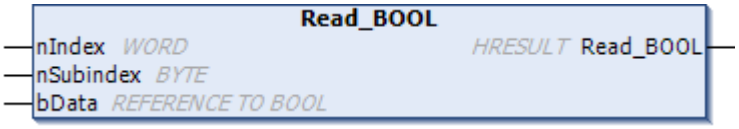
Outputs

Name	Type	Description
Read_INT8	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.9 READ_BOOL



Reads a BOOL parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
bData	Reference to BOOL	Reference to the data type BOOL

Outputs

Name	Type	Description
Read_BOOL	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.10 READ_OctedString

Reads an OctetString parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
arrData	Reference to T_IolMaxOctedString [▶ 17]	Reference to the T_IolMaxOctedString type
nLen	UDINT	Variable size

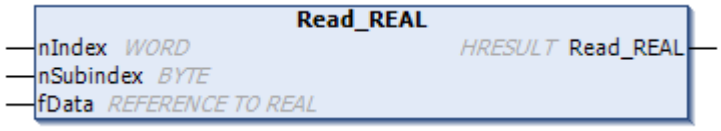
Outputs

Name	Type	Description
Read_OctedString	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.11 READ_Real



Reads a REAL parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
fData	Reference to REAL	Reference to the data type REAL

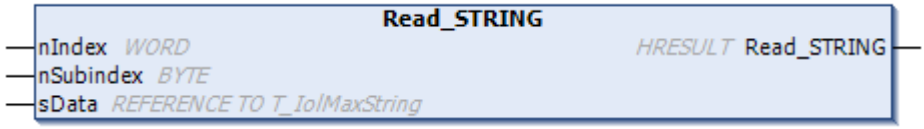
Outputs

Name	Type	Description
Read_REAL	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.12 READ_String



Reads a STRING parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
sData	Reference to T_IolMaxString [▶ 17]	Reference to the T_IolMaxString type

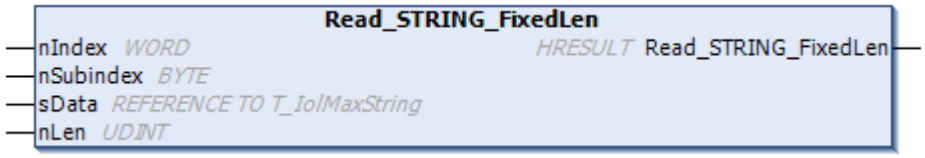
Outputs

Name	Type	Description
Read_STRING	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.13 READ_String_FixedLen



Reads a STRING parameter with a limited length of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
sData	Reference to T_IolMaxString [▶ 17]	Reference to the T_IolMaxString type
nLen	UDINT	Length of the string

Outputs

Name	Type	Description
Read_String_FixedLen	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.14 READ_Uint8

Reads a UINT8 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	Reference to USINT	Reference to the data type USINT;

Outputs

Name	Type	Description
Read_UINT8	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.15 Write

Writes a parameter of an IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
pDSTBuf	PVOID	Address of the buffer which is to receive the data that has been written. The programmer is himself responsible for dimensioning the buffer to a size that can accept LEN bytes. The buffer can be a single variable, an array or a structure, whose address can be determined with the ADR operator.
cbBufLen	UDINT	Number, in bytes, of the data to be written.

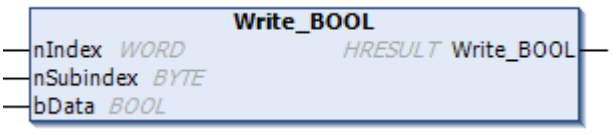
Outputs

Name	Type	Description
Write	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.16 Write_BOOL



Writes a BOOL parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
bData	BOOL	Value that is written

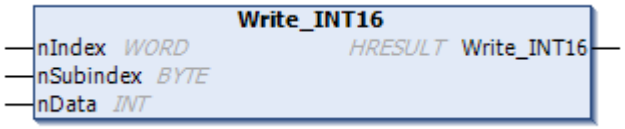
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.17 Write_INT16



Writes an INT16 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	INT	INT value that is written

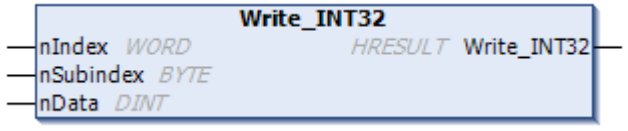
Outputs

Name	Type	Description
Write_INT16	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.18 Write_INT32



Writes an INT32 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	DINT	DINT value that is written

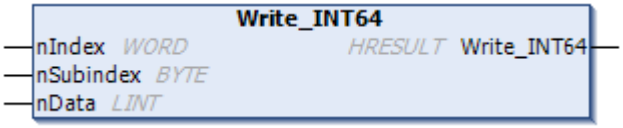
Outputs

Name	Type	Description
Write_INT32	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.19 Write_INT64



Writes an INT64 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	LINT	LINT value that is written

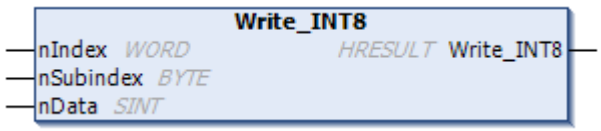
Outputs

Name	Type	Description
Write_INT64	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.20 Write_INT8



Writes an INT8 parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	SINT	SINT value that is written

Outputs

Name	Type	Description
Write_INT8	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.21 Write_OCTEDSTRING

Writes an OCTEDSTRING parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
arrData	T_IolMaxString [▶ 17]	T_IolMaxString value that is written
nLen	UDINT	Length of the string

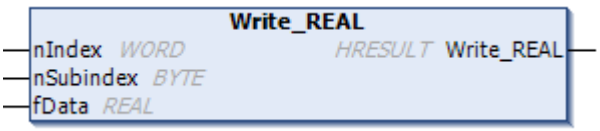
Outputs

Name	Type	Description
Write_OCTEDSTRING	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.22 Write_REAL



Writes a REAL parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
fData	REAL	REAL value that is written

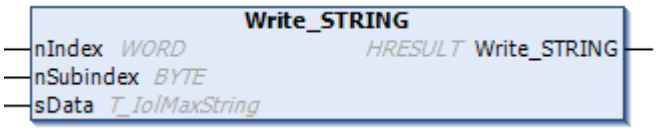
Outputs

Name	Type	Description
Write_Real	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.23 Write_STRING



Writes a string parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
sData	T_IolMaxString [▶ 17]	T_IolMaxString value that is written.

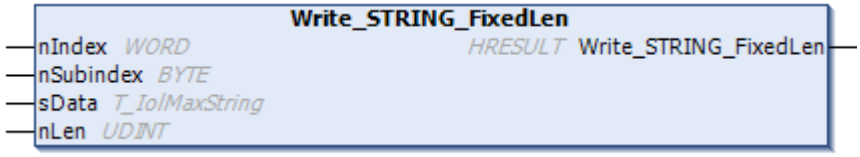
Outputs

Name	Type	Description
Write_STRING	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.24 Write_STRING_FixedLen



Writes a string parameter with a limited length of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
sData	T_IolMaxString [▶ 17]	T_IolMaxString value that will be written
nLen	UDINT	Number of characters

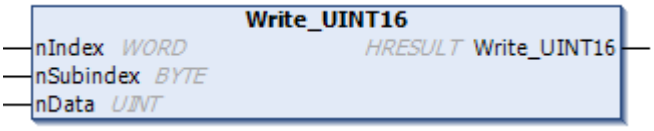
Outputs

Name	Type	Description
Write_STRING_FixedLen	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.25 Write_UINT16



Writes an UINT16 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	INT	INT value that is written

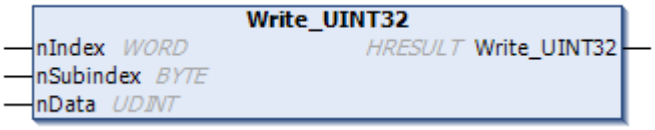
Outputs

Name	Type	Description
Write_UINT16	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.26 Write_UINT32



Writes an UINT32 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	UDINT	UDINT value that is written

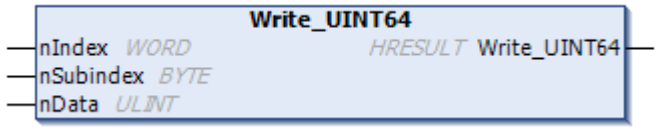
Outputs

Name	Type	Description
Write_UINT32	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.27 Write_UINT64



Writes an UINT64 parameter of the IO-Link device and switches the bytes to high endian.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	ULINT	ULINT value that is written

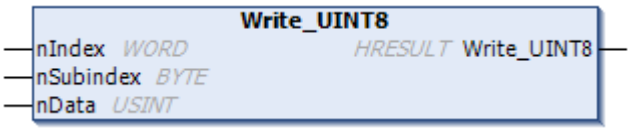
Outputs

Name	Type	Description
Write_UINT64	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.1.28 Write_UINT8



Writes an UINT8 parameter of the IO-Link device.

Inputs

Name	Type	Description
nIndex	WORD	Index of the IO-Link Parameter
nSubindex	BYTE	Subindex of the IO-Link Parameter 0 for the whole Structure
nData	USINT	USINT value that is written

Outputs

Name	Type	Description
Write_UINT8	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.2.2 Sample

The use of individual functions can look like this. In this sample, the ApplicationSpecificTag is written via Write and then read and checked via Read method.

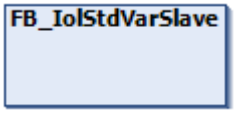
The following function block is part of the IoLink sample project. For more information, see the chapter Samples [▶ 69].

```
FUNCTION_BLOCK FB_Sample_03_IolSlave
VAR_INPUT
    bExecute                : BOOL := FALSE;
    sApplicationTag         : STRING(32) := '';
END_VAR
VAR_OUTPUT
    bBusy                   : BOOL := FALSE;
    bDone                   : BOOL := FALSE;
    bError                  : BOOL := FALSE;
    nAdsError               : E_AdsErr := E_AdsErr.NOERR;
    nIolError               : E_IolError := E_IolError.NoError;
    sErrorTxt               : STRING(255) := '';
END_VAR
VAR
    bExecuteOld             : BOOL;
    nState                  : INT := 0;
    fbIolSlave              : FB_IolSlave;
    bInitDone               : BOOL := FALSE;
    hr                      : HRESULT;
    sApplicationTagRead     : STRING(32) := ''

END_VAR
```

```
////
// Init
//
IF NOT bInitDone THEN
    bInitDone := TRUE;
    fbIolSlave.MasterAoeNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);
    fbIolSlave.Port := GVL_IO.nTerm2EPI1008;
    fbIolSlave.Name := 'EPI1008';
END_IF

////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
    bBusy := TRUE;
    nState := 10;    // WRITE
```

```
ELSIF NOT bExecute THEN
    bBusy := FALSE;
    bError := FALSE;
    bDone := FALSE;
    nAdsError := E_AdsErr.NOERR;
    nIolError := E_IolError.NoError;
    sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF

// IDLE
0:;

// WRITE
10:
    hr := fbIolSlave.Write(16#0018, 0, ADR(sApplicationTag), 32);

    IF NOT F_IolIsBusy(hr) THEN
        IF FAILED(hr) THEN
            bBusy := FALSE;
            bError := TRUE;
            sErrorTxt := fbIolSlave.ErrorTxt;
            nAdsError := F_IolGetAdsError(hr);
            nIolError := F_IolGetIolError(hr);
            nState := 0; // IDLE;
        ELSE
            nState := 20; // READ;
        END_IF
    END_IF

// READ
20:
    hr := fbIolSlave.Read(16#0018, 16#00, ADR(sApplicationTagRead), 32);

    IF NOT F_IolIsBusy(hr) THEN
        IF FAILED(hr) THEN
            bBusy := FALSE;
            bError := TRUE;
            sErrorTxt := fbIolSlave.ErrorTxt;
            nAdsError := F_IolGetAdsError(hr);
            nIolError := F_IolGetIolError(hr);
            nState := 0; // IDLE;
        ELSE
            nState := 30; // CHECK;
        END_IF
    END_IF

// CHECK
30:
    IF sApplicationTag <> sApplicationTagRead THEN
        bBusy := FALSE;
        bError := TRUE;
        sErrorTxt := 'sApplicationTag and sApplicationTagRead are not equal!';
    ELSE
        bBusy := FALSE;
        bDone := TRUE;
        nState := 0;
    END_IF
END_CASE

////
// call FBs
//
fbIolSlave();
```

6.3 FB_IolStdVarSlave



This function block inherits from the FB_IOlSlave [▶ 35]. It is intended for writing and reading standardized IO-Link parameters.

Sample

```
VAR
  bInitDone               : BOOL;
  sTerm3AmsNetId AT%I*    : AMSNETID;
  nTerm2EPI2328           : E_IoLPort := E_IolPort.Port2;
  fbIOLStdVarSlave              : FB_IolStdVarSlave;
END_VAR
```

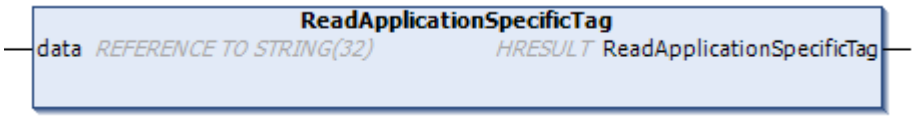
```
IF NOT bInitDone THEN
  bInitDone                 := TRUE;          //Assigns it only once
  fbIOLStdVarSlave.MasterAoeNetId := F_CreateAmsNetId(sTerm3AmsNetId);    //Creates NETID String
  fbIOLStdVarSlave.Port           := nTerm2EPI2328;  //Assigns Port
  fbIOLStdVarSlave.Name           := 'EPI2328';      //Gives device a name
END_IF
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1 Methods

6.3.1.1 ReadApplicationSpecificTag



Reads out the ApplicationSpecificTag.

Inputs

Name	Type	Description
data	Reference to STRING(32)	Variable of the buffer that is to receive the read data.

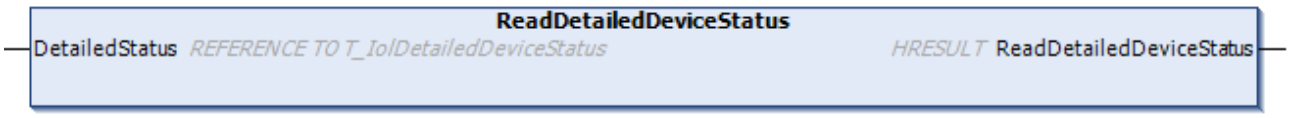
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.2 ReadDetailedDeviceStatus



Reads out the Detailed Device Status of an IO-Link device.

Inputs

Name	Type	Description
DetailedStatus	Reference to T_IolDetailedDeviceStatus [▶ 17]	Variable of the buffer that is to receive the read data.

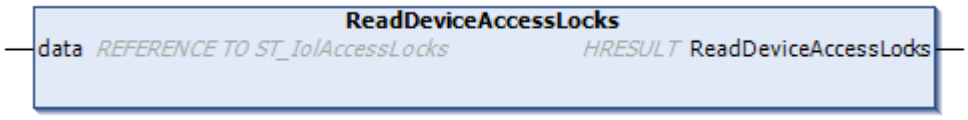
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.3 ReadDeviceAccessLocks



Reads out the device access lock of an IO-Link device.

Inputs

Name	Type	Description
data	Reference to ST_IolAcessLocks [▶ 16]	Specifies the device access lock

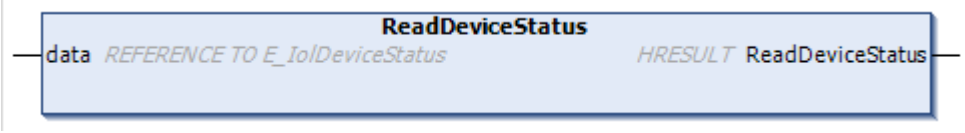
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.4 ReadDeviceStatus



Reads out the device status of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to E_IolDeviceStatus [▶ 14]	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.5 ReadDirectParameter1

Reads out the DirectParameter1 page of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to ST_IolDirectParameter1 [▶ 16]	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.6 ReadDirectParameter2

Reads out the DirectParamerer2 from an IO-Link device.

Inputs

Name	Type	Description
data	Reference to ST_IolDirectParameter2 [▶ 17]	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.7 ReadFirmwareVersion

Reads out the firmware version of an IO-Link device.

Inputs

Name	Type	Description
data	Reference to String(64)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.8 ReadHardwareVersion

Reads out the hardware version of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to STRING(64)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.9 ReadProductID

Reads out the product ID of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to STRING(64)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.10 ReadProductName

Reads out the product name of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to STRING(64)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.11 ReadProductText

Reads out the product text of an IO-Link device.

Inputs

Name	Type	Description
data	Reference to STRING(64)	Variable of the buffer that is to receive the read data.

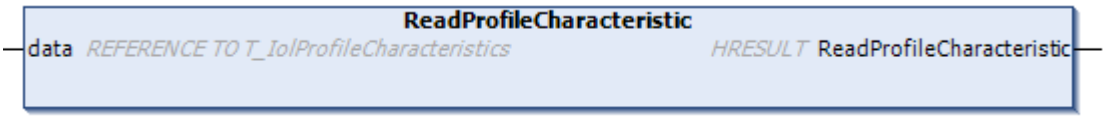
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.12 ReadProfileCharacteristic



Reads out the profile characteristics of an IO-Link device.

Inputs

Name	Type	Description
data	Reference to T_IolProfileCharacteristics [▶ 17]	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.13 ReadSerialNumber

Method reads out the serial number of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to STRING(16)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.14 ReadVendorName

Reads out the vendor name of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to STRING(64)	Variable of the buffer that is to receive the read data.

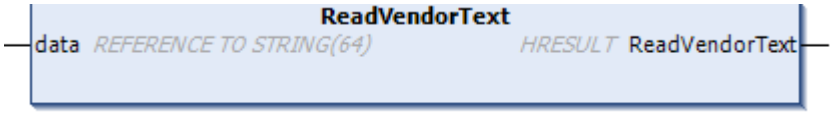
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.15 ReadVendorText



Reads out the vendor text of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to STRING(64)	Variable of the buffer that is to receive the read data.

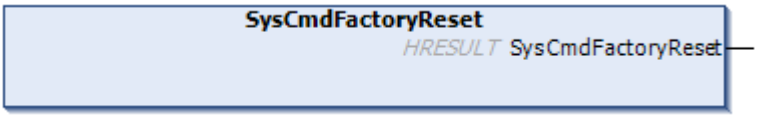
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.16 SysCmdFactoryReset



Resets the IO-Link device to factory settings.

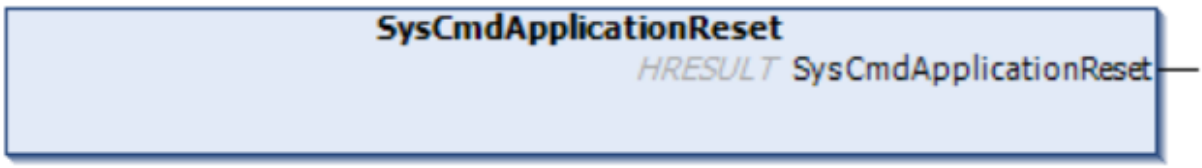
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.17 SysCmdApplicationReset



Resets the IO-Link device to the application settings.

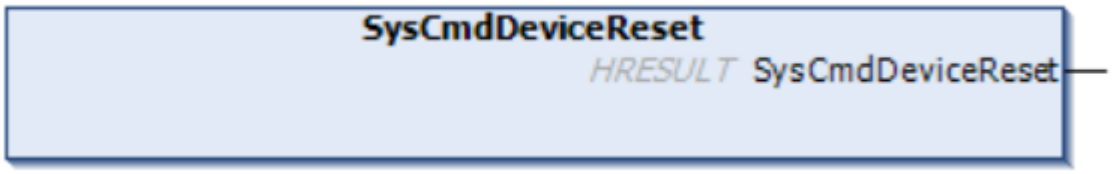
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.18 SysCmdDeviceReset



Resets the IO-Link device to device settings.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.19 SysCmdBlockParamStart

Starts the block parameterization.

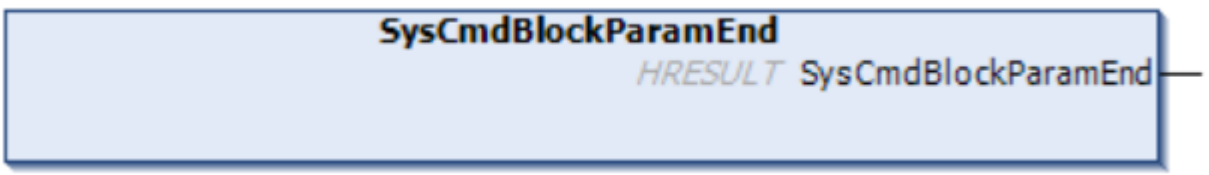
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.20 SysCmdBlockParamEnd



Ends the block parameterization.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.21 SysCmdStoreParameterServer

Sets the data storage via the PLC.

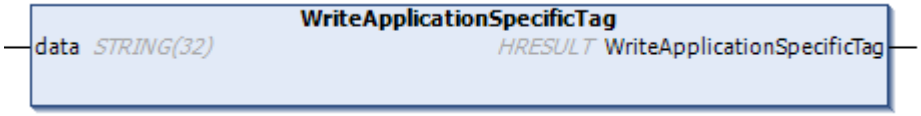
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.22 WriteApplicationSpecificTag



Writes the IO-Link parameter ApplicationSpecifigTag.

Inputs

Name	Type	Description
data	STRING(32)	Variable of the buffer that is to receive the read data.

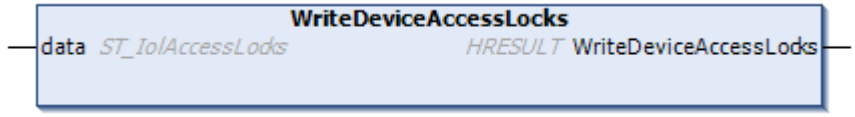
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.23 WriteDeviceAccessLocks



Writes the DeviceAcessLocks of an IO-Link device.

Inputs

Name	Type	Description
data	Reference to ST_IolAccessLocks [▶ 16]	Variable of the buffer that is to receive the written data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.24 WriteDirectParameter2

Writes the DirectParameter2 parameter of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to ST_IolDirectParameter2 [▶ 17]	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.25 WriteProfileCharacteristic

Writes the ProfileCharacteristic parameter of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to T_IolProfileCharacteristics [▶ 17]	Variable of the buffer that is to receive the written data.

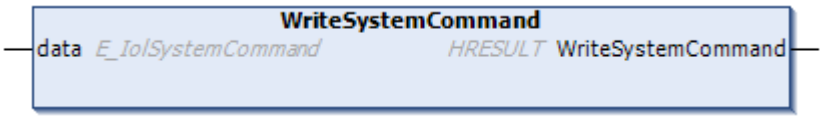
Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.1.26 WriteSystemCommand



Writes any system command for the IO-Link device.

Inputs

Name	Type	Description
data	Reference to E_IolSystemcommand [▶ 15]	Variable of the buffer that is to receive the written data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.3.2 Sample

In this sample, the ApplicationSpecificTag is written using the WriteApplicationspecificTag methods and then read and checked using the ReadApplicationSpecificTag method.

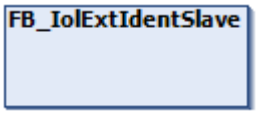
```
FUNCTION_BLOCK FB_Sample_04_IolStdVarSlave
VAR_INPUT
  bExecute          :BOOL := FALSE;
  sApplicationTag   :STRING(32) := '';
END_VAR
VAR_OUTPUT
  bBusy             :BOOL := FALSE;
  bDone             :BOOL := FALSE;
  bError            :BOOL := FALSE;
  nAdsError         :E_AdsErr := E_AdsErr.NOERR;
  nIolError         :E_IolError := E_IolError.NoError;
```

```
  sErrorTxt        :STRING(255)  := '';
END_VAR
VAR
  bExecuteOld          :BOOL;
  nState               :INT := 0;
  fbIolSlave           :FB_IolStdVarSlave;
  bInitDone            :BOOL := FALSE;
  hr                   :HRESULT;
  sApplicationTagRead : STRING(32)  := '';
END_VAR
```

```
////
// Init
//
IF NOT bInitDone THEN
  bInitDone := TRUE;
  fbIolSlave.MasterAoeNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);    //Creates NETID String
  fbIolSlave.Port := GVL_IO.nTerm2EPI1008;
  fbIolSlave.Name := 'EPI1008';
END_IF

////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
  bBusy := TRUE;
  nState := 10;  // WRITE
ELSIF NOT bExecute THEN
  bBusy := FALSE;
  bError := FALSE;
  bDone := FALSE;
  nAdsError := E_AdsErr.NOERR;
  nIolError := E_IolError.NoError;
  sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF

// IDLE
0:;

// WRITE
10:
  hr := fbIolSlave.WriteApplicationSpecificTag(sApplicationTag);

  IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
      bBusy := FALSE;
      bError := TRUE;
      sErrorTxt := fbIolSlave.ErrorTxt; // TODO
      nAdsError := F_IolGetAdsError(hr);
      nIolError := F_IolGetIolError(hr);
      nState := 0; // IDLE;
    ELSE
      nState := 20; // WRITE;
    END_IF
  END_IF

// READ
20:
  hr := fbIolSlave.ReadApplicationSpecificTag(sApplicationTagRead);

  IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
      bBusy := FALSE;
      bError := TRUE;
      sErrorTxt := fbIolSlave.ErrorTxt; // TODO
      nAdsError := F_IolGetAdsError(hr);
      nIolError := F_IolGetIolError(hr);
      nState := 0; //  IDLE;
    ELSE
      nState := 30; // CHECK;
    END_IF
  END_IF

// CHECK
```

```
30:
  IF sApplicationTag <> sApplicationTagRead THEN
    bBusy := FALSE;
    bError := TRUE;
    sErrorTxt := 'sApplicationTag and sApplicationTagRead are not equal!';
  ELSE
    bBusy := FALSE;
    bDone := TRUE;
    nState := 0;
  END_IF
END_CASE

////
// call FBs
//
fbIolSlave();
```

6.4 FB_IolExtIdentSlave



FB_IoLExtSlave inherits from the function block FB_IolSlave.

Only IO-Link devices with the profile: 0x8100 [▶ 10] have these parameters.

Sample

```
VAR
  bInitDone                : BOOL;
  sTerm3AmsNetId  AT%I*    : AMSNETID;
  nTerm2EPI2328            : E_IoLPort := E_IolPort.Port2;
  fbIOLExtIdentSlave       : FB_IolExtIdentSlave;
END_VAR
```

```
IF NOT bInitDone THEN
  bInitDone                 := TRUE;                  //Assigns it only once
  fbIOLExtIdentSlave.MasterAoeNetId := F_CreateAmsNetId(sTerm3AmsNetId)   //Creates NETID String
  fbIOLExtIdentSlave.Port           := nTerm2EPI2328;          //Assigns Port
  fbIOLExtIdentSlave.Name           := 'EPI2328';              //Gives device a name
END_IF
```

Requirements

Development environment	Required PLC library
TwinCAT from v3.1.4024.25	Tc3_IoLink from v3.3.4.0

6.4.1 Methods

6.4.1.1 ReadFunctionTag

Function to read the FunctionTag.

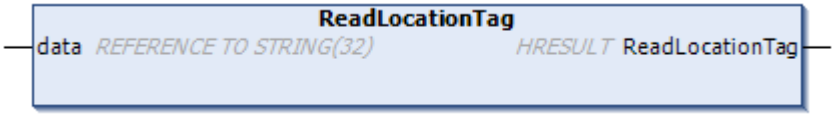
Inputs

Name	Type	Description
data	Reference to STRING(32)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

6.4.1.2 ReadLocationTag



Function to read out the LocationTag. The LocationTag is a parameter that contains the location information of the IO-Link device.

Inputs

Name	Type	Description
data	Reference to STRING(32)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

6.4.1.3 WriteFunctionTag

Writes the FunctionTag parameter.

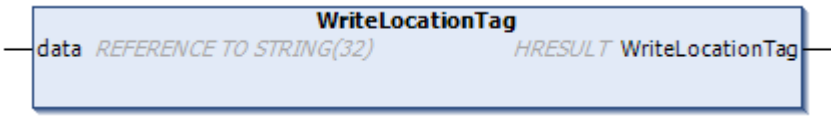
Inputs

Name	Type	Description
data	STRING(32)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

6.4.1.4 WriteLocationTag



Writes the LocationTag of an IO-Link device.

Inputs

Name	Type	Description
data	STRING(32)	Variable of the buffer that is to receive the read data.

Outputs

Name	Type	Description
hresult	HRESULT	Return value

6.4.2 Sample

In this sample, the ApplicationSpecificTag is written using the WriteApplicationspecificTag methods and then read and checked using the ReadApplicationSpecificTag method.

```
FUNCTION_BLOCK FB_Sample_04_IolStdVarSlave
VAR_INPUT
  bExecute          :BOOL := FALSE;
  sApplicationTag   :STRING(32) := '';
END_VAR
VAR_OUTPUT
  bBusy             :BOOL := FALSE;
  bDone             :BOOL := FALSE;
  bError            :BOOL := FALSE;
  nAdsError         :E_AdsErr := E_AdsErr.NOERR;
  nIolError         :E_IolError := E_IolError.NoError;
  sErrorTxt         :STRING(255)  := '';
END_VAR
VAR
  bExecuteOld          :BOOL;
  nState              :INT := 0;
  fbIolSlave          :FB_IolStdVarSlave;
  bInitDone           :BOOL := FALSE;
  hr                  :HRESULT;
  sApplicationTagRead : STRING(32)  := '';
END_VAR
```

```
////
// Init
//
IF NOT bInitDone THEN
  bInitDone := TRUE;
  fbIolSlave.MasterAoeNetId := F_CreateAmsNetId(GVL_IO.sTerm2AmsNetId);    //Creates NETID String
  fbIolSlave.Port := GVL_IO.nTerm2EPI1008;
  fbIolSlave.Name := 'EPI1008';
END_IF

////
// Execute/Busy State Machine
//
IF bExecute AND NOT bExecuteOld AND NOT bBusy THEN
  bBusy := TRUE;
  nState := 10;  // WRITE
ELSIF NOT bExecute THEN
  bBusy := FALSE;
  bError := FALSE;
  bDone := FALSE;
  nAdsError := E_AdsErr.NOERR;
  nIolError := E_IolError.NoError;
  sErrorTxt := '';
END_IF
bExecuteOld := bExecute;

////
// FB State Machine
//
CASE nState OF

// IDLE
0:;

// WRITE
10:
```

```
  hr := fbIolSlave.WriteApplicationSpecificTag(sApplicationTag);

  IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
      bBusy := FALSE;
      bError := TRUE;
      sErrorTxt := fbIolSlave.ErrorTxt; // TODO
      nAdsError := F_IolGetAdsError(hr);
      nIolError := F_IolGetIolError(hr);
      nState := 0; // IDLE;
    ELSE
      nState := 20; // WRITE;
    END_IF
  END_IF

// READ
20:
  hr := fbIolSlave.ReadApplicationSpecificTag(sApplicationTagRead);

  IF NOT F_IolIsBusy(hr) THEN
    IF FAILED(hr) THEN
      bBusy := FALSE;
      bError := TRUE;
      sErrorTxt := fbIolSlave.ErrorTxt; // TODO
      nAdsError := F_IolGetAdsError(hr);
      nIolError := F_IolGetIolError(hr);
      nState := 0; //  IDLE;
    ELSE
      nState := 30; // CHECK;
    END_IF
  END_IF

// CHECK
30:
  IF sApplicationTag <> sApplicationTagRead THEN
    bBusy := FALSE;
    bError := TRUE;
    sErrorTxt := 'sApplicationTag and sApplicationTagRead are not equal!';
  ELSE
    bBusy := FALSE;
    bDone := TRUE;
    nState := 0;
  END_IF
END_CASE

////
// call FBs
//
fbIolSlave();
```

7 Samples

Sample code and configurations for this product can be obtained from the corresponding repository on GitHub: https://github.com/Beckhoff/PlcLibrary_Tc3_IoLink_Sample.
There you have the option to clone the repository or download a ZIP file containing the sample.

8 Appendix

8.1 Parameter data exchange

An intelligent IO-Link sensor/actuator can support parameterization by ISDU (**I**ndexed **S**ervice **D**ata **U**nit). The PLC must explicitly query or, when marked as such, send these acyclic service data.

Access ISDU

TwinCAT supports access via ADS and via the CoE directory.

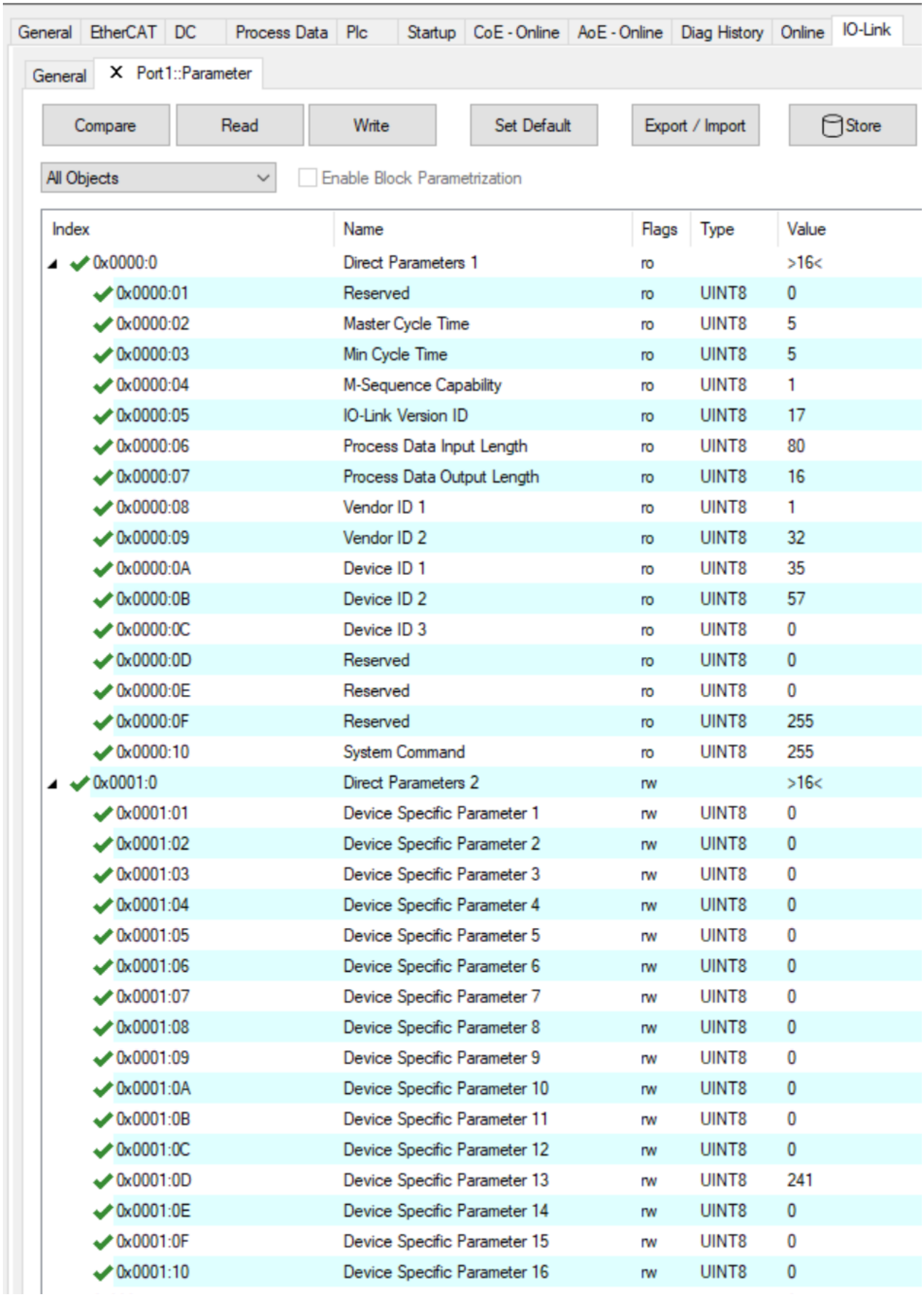
The respective parameter is addressed via the so-called ISDU index. The following ranges are available:

Name	Index range
System	0x00..0x0F
Identification	0x10..0x1F
Diagnostic	0x20..0x2F
Communication	0x30..0x3F
Preferred Index	0x40..0xFE
Extended Index	0x0100..0x3FFF
	the range 0x4000..0xFFFF is reserved

The use of the implementation of these ranges is the responsibility of the sensor/actuator vendor. For clarification you see only some possible indices with designation listed here. Refer to the corresponding chapter in the info system Object description and parameterization.

Direct parameters

The indices 0x0000 and 0x0001 are special parameters that each IO-Link device possesses. An IO-Link master requires the information from Direct Parameter 1 to establish IO-Link communication with the device.



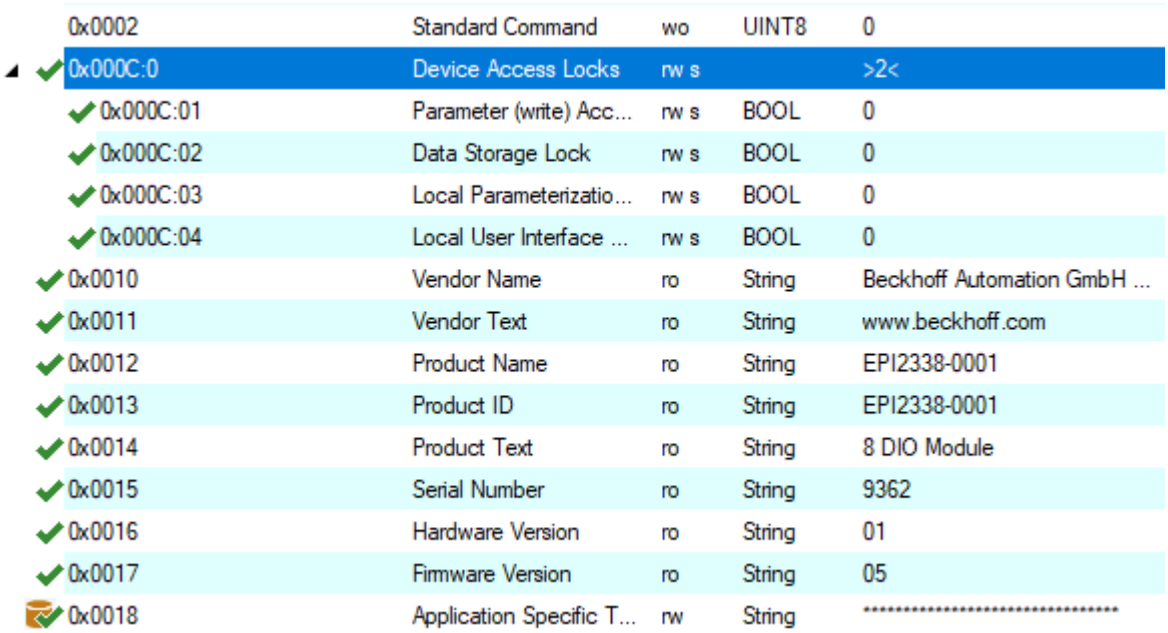
Index	Name	Flags	Type	Value
0x0000:0	Direct Parameters 1	ro		>16<
0x0000:01	Reserved	ro	UINT8	0
0x0000:02	Master Cycle Time	ro	UINT8	5
0x0000:03	Min Cycle Time	ro	UINT8	5
0x0000:04	M-Sequence Capability	ro	UINT8	1
0x0000:05	IO-Link Version ID	ro	UINT8	17
0x0000:06	Process Data Input Length	ro	UINT8	80
0x0000:07	Process Data Output Length	ro	UINT8	16
0x0000:08	Vendor ID 1	ro	UINT8	1
0x0000:09	Vendor ID 2	ro	UINT8	32
0x0000:0A	Device ID 1	ro	UINT8	35
0x0000:0B	Device ID 2	ro	UINT8	57
0x0000:0C	Device ID 3	ro	UINT8	0
0x0000:0D	Reserved	ro	UINT8	0
0x0000:0E	Reserved	ro	UINT8	0
0x0000:0F	Reserved	ro	UINT8	255
0x0000:10	System Command	ro	UINT8	255
0x0001:0	Direct Parameters 2	rw		>16<
0x0001:01	Device Specific Parameter 1	rw	UINT8	0
0x0001:02	Device Specific Parameter 2	rw	UINT8	0
0x0001:03	Device Specific Parameter 3	rw	UINT8	0
0x0001:04	Device Specific Parameter 4	rw	UINT8	0
0x0001:05	Device Specific Parameter 5	rw	UINT8	0
0x0001:06	Device Specific Parameter 6	rw	UINT8	0
0x0001:07	Device Specific Parameter 7	rw	UINT8	0
0x0001:08	Device Specific Parameter 8	rw	UINT8	0
0x0001:09	Device Specific Parameter 9	rw	UINT8	0
0x0001:0A	Device Specific Parameter 10	rw	UINT8	0
0x0001:0B	Device Specific Parameter 11	rw	UINT8	0
0x0001:0C	Device Specific Parameter 12	rw	UINT8	0
0x0001:0D	Device Specific Parameter 13	rw	UINT8	241
0x0001:0E	Device Specific Parameter 14	rw	UINT8	0
0x0001:0F	Device Specific Parameter 15	rw	UINT8	0
0x0001:10	Device Specific Parameter 16	rw	UINT8	0

Index	Subindex	Name	Flag	Type	Definition
Direct Parameter1					

Index	Subindex	Name	Flag	Type	Definition
0x0000	0x01	Reserved	ro	UINT8	Reserved
0x0000	0x02	Master Cycle Time	ro	UINT8	Master cycle time
0x0000	0x03	Min Cycle Time	ro	UINT8	Minimum cycle time
0x0000	0x04	M-Sequence Capability	ro	UINT8	M sequence
0x0000	0x05	IO-Link Version ID	ro	UINT8	IO-Link Revisions-ID
0x0000	0x06	Process Data Input Length	ro	UINT8	Process data input length
0x0000	0x07	Process Data Output length	ro	UINT8	Process data output length
0x0000	0x08	Vendor ID 1	ro	UINT8	Vendor ID 1
0x0000	0x09	Vendor ID 2	ro	UINT8	Vendor ID 2
0x0000	0xA	Device ID1	ro	UINT8	Device ID 1
0x0000	0xB	Device ID 2	ro	UINT8	Device ID 2
0x0000	0xC	Device ID3	ro	UINT8	Device ID 3
0x0000	0xD	Reserved	ro	UINT8	Reserved
0x0000	0xE	Reserved	ro	UINT8	Reserved
0x0000	0xF	Reserved	ro	UINT8	Reserved
Direct Parameter 2					
0x0001	0x01	Device Specific Parameter 1	rw	UINT8	Device-specific
0x0001	0x02	Device Specific Parameter 2	rw	UINT8	Device-specific
0x0001	0x03	Device Specific Parameter 3	rw	UINT8	Device-specific
0x0001	0x04	Device Specific Parameter 4	rw	UINT8	Device-specific
0x0001	0x05	Device Specific Parameter 5	rw	UINT8	Device-specific
0x0001	0x06	Device Specific Parameter 6	rw	UINT8	Device-specific
0x0001	0x07	Device Specific Parameter 7	rw	UINT8	Device-specific
0x0001	0x08	Device Specific Parameter 8	rw	UINT8	Device-specific
0x0001	0x09	Device Specific Parameter 9	rw	UINT8	Device-specific
0x0001	0xA	Device Specific Parameter 10	rw	UINT8	Device-specific
0x0001	0xB	Device Specific Parameter 11	rw	UINT8	Device-specific
0x0001	0xC	Device Specific Parameter 12	rw	UINT8	Device-specific
0x0001	0xD	Device Specific Parameter 13	rw	UINT8	Device-specific
0x0001	0xE	Device Specific Parameter 14	rw	UINT8	Device-specific
0x0001	0xF	Device Specific Parameter 15	rw	UINT8	Device-specific
0x0001	0x10	Device Specific Parameter 16	rw	UINT8	Device-specific

Specified parameters

The standard IO-Link parameters are also listed here.



	0x0002	Standard Command	wo	UINT8	0
◢	0x000C:0	Device Access Locks	rw s		>2<
	0x000C:01	Parameter (write) Acc...	rw s	BOOL	0
	0x000C:02	Data Storage Lock	rw s	BOOL	0
	0x000C:03	Local Parameterizatio...	rw s	BOOL	0
	0x000C:04	Local User Interface ...	rw s	BOOL	0
	0x0010	Vendor Name	ro	String	Beckhoff Automation GmbH ...
	0x0011	Vendor Text	ro	String	www.beckhoff.com
	0x0012	Product Name	ro	String	EPI2338-0001
	0x0013	Product ID	ro	String	EPI2338-0001
	0x0014	Product Text	ro	String	8 DIO Module
	0x0015	Serial Number	ro	String	9362
	0x0016	Hardware Version	ro	String	01
	0x0017	Firmware Version	ro	String	05
	0x0018	Application Specific T...	rw	String	**********

Index	Subindex	Name	Flag	Type	Definition
Device Access Locks					
0x000C	0x01	Parameter (Write) Access Lock	rw s	BOOL	Locks the write access
0x000C	0x02	Data Storage Lock	rw s	BOOL	Locks the Data Storage access
0x000C	0x03	Local Parametrization Lock	rw s	BOOL	Locks the local parameter access
0x000C	0x04	Local Userinterface Lock	rw s	BOOL	Locks the local user interface
Detection parameters					
0x0010	0x00	Vendor Name	ro	STRING	Vendor name
0x0011	0x00	Vendor Text	ro	STRING	Vendor text
0x0012	0x00	Product Name	ro	STRING	Product name
0x0013	0x00	Product ID	ro	STRING	Product ID
0x0014	0x00	Product Text	ro	STRING	Product text
0x0015	0x00	Serial Number	ro	STRING	Serial number
0x0016	0x00	Hardware version	ro	STRING	Hardware version
0x0017	0x00	FirmwareVersion	ro	STRING	Firmware version
0x0018	0x00	ApplicationSpecificTag	rw	STRING	Application specific text
0x0019	0x00	Function Tag	rw	STRING	Function text

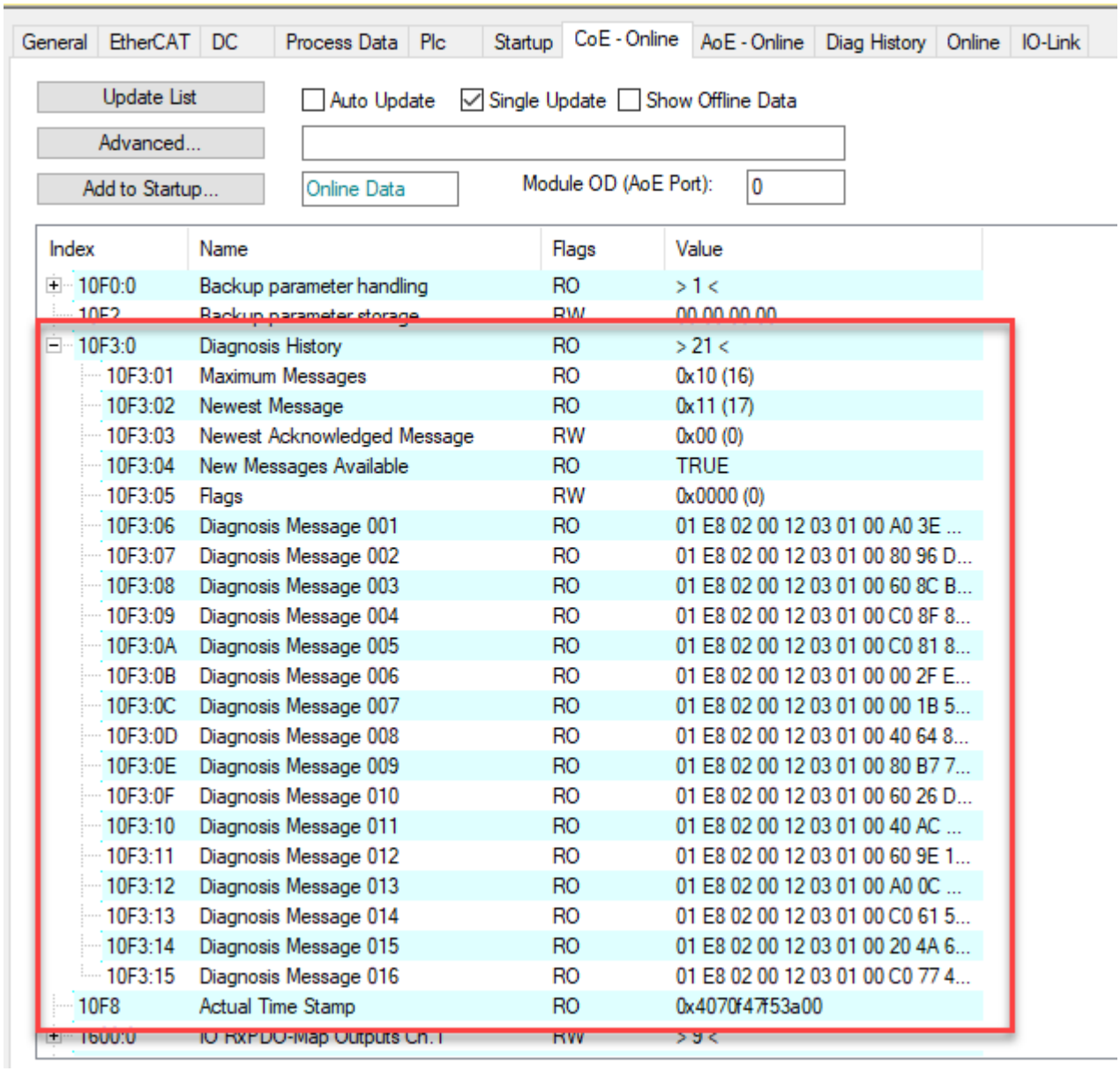
Device Status and Detailed Device Status

The parameters 0x0024 Device Status and 0x0025 Detailed Device Description are optional parameters for standard IO-Link devices. However, if an IO-Link device supports profiles, these parameters are mandatory.

Index	Subindex	Name	Flag	Type	Definition
0x0024	0x00	Device Status	ro	Uint8	Device status
Detailed Device Status					
0x0025	0x01	Detailed Device Status	ro s	OctetString	Event of an IO-Link device
0x0025	0x02	Detailed Device Status	ro s	OctetString	Event of an IO-Link device

Index	Subindex	Name	Flag	Type	Definition
0x0025	0x03	Detailed Device Status	ro s	OctetString	Event of an IO-Link device
0x0025	0x04	Detailed Device Status	ro s	OctetString	Event of an IO-Link device

If an IO-Link device does not support the parameter 0x0025, the event history can also be read from the CoE parameter 0x10F3 using the FB_EcCoeSdoRead function block.



8.2 Error Codes

8.2.1 IO-Link error codes

Error codes are generated in the event of an error during ADS access to an IO-Link device.

The possible error codes are listed in the following tables.

Example of an AdsReturnCode

AdsReturnCode 0x**80110700**

- **80**: Device Application Error (IO-Link Spec),
- **11**: Index not Available (IO-Link Spec),
- **0700**: General ADS Error

ErrorTypes (IO-Link Spec)

Incident	Error Code	Additional Code	Name	Definition
Device application error – no details	0x80	0x00	APP_DEV	This ErrorType shall be used if the requested service has been refused by the Device application and no detailed
Index not available	0x80	0x11	IDX_NOTAVAIL	This ErrorType shall be used whenever a read or write access occurs to a not existing Index.
Subindex not available	0x80	0x12	SUBIDX_NOTAVAIL	This ErrorType shall be used whenever a read or write access occurs to a not existing Subindex.
Service temporarily not available	0x80	0x20	SERV_NOTAVAIL	This ErrorType shall be used if a parameter is not accessible for a read or write service due to the current state of the Device application.
Service temporarily not available – local control	0x80	0x21	SERV_NOTAVAIL_LOCCTRL	This ErrorType shall be used if a parameter is not accessible for a read or write service due to an ongoing local operation at the Device (for example operation or parameterization via an on-board Device control panel).
Service temporarily not available – Device control	0x80	0x22	SERV_NOTAVAIL_DEVCTRL	This ErrorType shall be used if a read or write service is not accessible due to a remote triggered state of the device application (for example parameterization during a remote triggered teach-in operation or calibration).
Access denied	0x80	0x23	IDX_NOT_WRITEABLE	This ErrorType shall be used if a write service tries to access a read-only parameter.
Parameter value out of range	0x80	0x30	PAR_VALOUTOFRNG	This ErrorType shall be used for a write service to a parameter outside its permitted range of values.
Parameter value above limit	0x80	0x31	PAR_VALGTLIM	This ErrorType shall be used for a write service to a parameter above its specified value range.
Parameter value below limit	0x80	0x32	PAR_VALLTLIM	This ErrorType shall be used for a write service to a parameter below its specified value range.
Parameter length overrun	0x80	0x33	VAL_LENOVRRUN	This ErrorType shall be used when the content of a write service to a parameter is greater than the parameter specified length. This ErrorType shall also be used, if a data object is too large to be processed by the Device application (for example ISDU buffer restriction).
Parameter length underrun	0x80	0x34	VAL_LENUNDRUN	This ErrorType shall be used when the content of a write service to a parameter is less than the parameter specified length (for example write access of an Unsigned16 value to an Unsigned32 parameter).
Function not available	0x80	0x35	FUNC_NOTAVAIL	This ErrorType shall be used for a write service with a command value not supported by the Device application (for example a SystemCommand with a value not implemented).
Function temporarily unavailable	0x80	0x36	FUNC_UNAVAILTEMP	This ErrorType shall be used for a write service with a command value calling a Device function not available due to the current state of the Device application (for example a SystemCommand).
Invalid parameter set	0x80	0x40	PAR_SETINVALID	This ErrorType shall be used if values sent via single parameter transfer are not consistent with other actual parameter settings (for example overlapping set points for a binary data setting
Inconsistent parameter set	0x80	0x41	PAR_SETINCONSIST	This ErrorType shall be used at the termination of a block parameter transfer with ParamDownloadEnd or ParamDownloadStore if the plausibility check shows inconsistencies
Application not ready	0x80	0x82	APP_DEVNOTRDY	This ErrorType shall be used if a read or write service is refused due to a temporarily unavailable application (for example peripheral controllers during startup).
Vendor specific	0x81	0x00	UNSPECIFIC	This ErrorType will be propagated directly to higher level processing elements as an error (no warning) by the Master.
Vendor specific	0x81	0x01 to 0xFF	VENDOR_SPECIFIC	

Derived ErrorTypes (IO-Link Spec)

Incident	Error Code	Additional Code	Name	Definition
Master – Communication error	0x10	0x00	COM_ERR	The Master generates a negative service response with this ErrorType if a communication error occurred during a read or write service, for example the SDCI connection is interrupted.
Master – ISDU timeout	0x11	0x00	I-SERVICE_TIMEOUT	The Master generates a negative service response with this ErrorType, if a Read or Write service is pending longer than the specified I-Service timeout in the Master.
Device Event – ISDU error (DL, Error, single shot, 0x5600)	0x11	0x00	I-SERVICE_TIMEOUT	If the Master received an Event with the EventQualifier and the EventCode 0x5600, a negative service response indicating a service timeout is generated and returned to the requester (Master – ISDU timeout).
Device Event – ISDU illegal service primitive (AL, Error, single shot, 0x5800)	0x11	0x00	I-SERVICE_TIMEOUT	If the Master received an Event with the EventQualifier and the EventCode 0x5800, a negative service response indicating a service timeout is generated and returned to the requester (Master – ISDU timeout).
Master – ISDU checksum error	0x56	0x00	M_ ISDU_CHECKSUM	The Master generates a negative service response with this ErrorType, if its data link layer detects an ISDU checksum error.
Master – ISDU illegal service primitive	0x57	0x00	M_ ISDU_ILLEGAL	The Master generates a negative service response with this ErrorType, if its data link layer detects an ISDU illegal service primitive.
Device Event – ISDU buffer overflow (DL, Error, single shot, 0x5200)	0x80	0x33	VAL_LENOVRRUN	If the Master received an Event with the EventQualifier and the EventCode 0x5200, a negative service response indicating a parameter length overrun is generated and returned to the requester (see parameter length overrun) Events from legacy Devices shall be redirected in compatibility mode to this derived ErrorType

Further error diagnosis options

Device State Inputs Device (0x1A05)

It is indicated in the PDO "Device Diag" (0xF101:0D) that at least one event has occurred in the "Diag History".
"Device State" is the standard status bit for EtherCAT slaves and shows, for example, that communication with one of the slaves has been interrupted.

Device State Inputs (0x1A04)

The status of the IO-Link devices is displayed at the respective port (see Comment field in the System Manager).

Nominal/actual comparison of the parameter objects

The 0x90n0 (Info Data) indices can be used to validate the 0x80n0 configuration indices of the connected IO-Link device.
In the event of an error, these objects can be used to compare the configuration with the actual state.

Lost Frame Counter

The Lost Frame counter in object 0xA0n0:02 is for the diagnosis of the transmission quality. TwinCAT provides the possibility here to diagnose problems, e. g. with the wiring, EMC or power supply.

8.2.2 ADS Return Codes

Grouping of error codes:
Global error codes: ADS Return Codes [▶ 77]... (0x9811_0000 ...)
Router error codes: ADS Return Codes [▶ 77]... (0x9811_0500 ...)
General ADS errors: ADS Return Codes [▶ 78]... (0x9811_0700 ...)
RTime error codes: ADS Return Codes [▶ 79]... (0x9811_1000 ...)

Global error codes

Hex	Dec	HRESULT	Name	Description
0x0	0	0x98110000	ERR_NOERROR	No error.
0x1	1	0x98110001	ERR_INTERNAL	Internal error.
0x2	2	0x98110002	ERR_NORTIME	No real time.
0x3	3	0x98110003	ERR_ALLOCLOCKEDMEM	Allocation locked – memory error.
0x4	4	0x98110004	ERR_INSERTMAILBOX	Mailbox full – the ADS message could not be sent. Reducing the number of ADS messages per cycle will help.
0x5	5	0x98110005	ERR_WRONGRECEIVEHMSG	Wrong HMSG.
0x6	6	0x98110006	ERR_TARGETPORTNOTFOUND	Target port not found – ADS server is not started or is not reachable.
0x7	7	0x98110007	ERR_TARGETMACHINENOTFOUND	Target computer not found – AMS route was not found.
0x8	8	0x98110008	ERR_UNKNOWNCMDID	Unknown command ID.
0x9	9	0x98110009	ERR_BADTASKID	Invalid task ID.
0xA	10	0x9811000A	ERR_NOIO	No IO.
0xB	11	0x9811000B	ERR_UNKNOWNAMSCMD	Unknown AMS command.
0xC	12	0x9811000C	ERR_WIN32ERROR	Win32 error.
0xD	13	0x9811000D	ERR_PORTNOTCONNECTED	Port not connected.
0xE	14	0x9811000E	ERR_INVALIDAMSLENGTH	Invalid AMS length.
0xF	15	0x9811000F	ERR_INVALIDAMSNETID	Invalid AMS Net ID.
0x10	16	0x98110010	ERR_LOWINSTLEVEL	Installation level is too low –TwinCAT 2 license error.
0x11	17	0x98110011	ERR_NODEBUGINTAVAILABLE	No debugging available.
0x12	18	0x98110012	ERR_PORTDISABLED	Port disabled – TwinCAT system service not started.
0x13	19	0x98110013	ERR_PORTALREADYCONNECTED	Port already connected.
0x14	20	0x98110014	ERR_AMSSYNC_W32ERROR	AMS Sync Win32 error.
0x15	21	0x98110015	ERR_AMSSYNC_TIMEOUT	AMS Sync Timeout.
0x16	22	0x98110016	ERR_AMSSYNC_AMSERROR	AMS Sync error.
0x17	23	0x98110017	ERR_AMSSYNC_NOINDEXINMAP	No index map for AMS Sync available.
0x18	24	0x98110018	ERR_INVALIDAMSPORT	Invalid AMS port.
0x19	25	0x98110019	ERR_NOMEMORY	No memory.
0x1A	26	0x9811001A	ERR_TCPSEND	TCP send error.
0x1B	27	0x9811001B	ERR_HOSTUNREACHABLE	Host unreachable.
0x1C	28	0x9811001C	ERR_INVALIDAMSFRAGMENT	Invalid AMS fragment.
0x1D	29	0x9811001D	ERR_TLSSEND	TLS send error – secure ADS connection failed.
0x1E	30	0x9811001E	ERR_ACCESSDENIED	Access denied – secure ADS access denied.

Router error codes

Hex	Dec	HRESULT	Name	Description
0x500	1280	0x98110500	ROUTERERR_NOLOCKEDMEMORY	Locked memory cannot be allocated.
0x501	1281	0x98110501	ROUTERERR_RESIZEMEMORY	The router memory size could not be changed.
0x502	1282	0x98110502	ROUTERERR_MAILBOXFULL	The mailbox has reached the maximum number of possible messages.
0x503	1283	0x98110503	ROUTERERR_DEBUGBOXFULL	The Debug mailbox has reached the maximum number of possible messages.
0x504	1284	0x98110504	ROUTERERR_UNKNOWNPORTTYPE	The port type is unknown.
0x505	1285	0x98110505	ROUTERERR_NOTINITIALIZED	The router is not initialized.
0x506	1286	0x98110506	ROUTERERR_PORTALREADYINUSE	The port number is already assigned.

Hex	Dec	HRESULT	Name	Description
0x507	1287	0x98110507	ROUTERERR_NOTREGISTERED	The port is not registered.
0x508	1288	0x98110508	ROUTERERR_NOMOREQUEUES	The maximum number of ports has been reached.
0x509	1289	0x98110509	ROUTERERR_INVALIDPORT	The port is invalid.
0x50A	1290	0x9811050A	ROUTERERR_NOTACTIVATED	The router is not active.
0x50B	1291	0x9811050B	ROUTERERR_FRAGMENTBOXFULL	The mailbox has reached the maximum number for fragmented messages.
0x50C	1292	0x9811050C	ROUTERERR_FRAGMENTTIMEOUT	A fragment timeout has occurred.
0x50D	1293	0x9811050D	ROUTERERR_TOBEREMOVED	The port is removed.

General ADS error codes

Hex	Dec	HRESULT	Name	Description
0x700	1792	0x98110700	ADSERR_DEVICE_ERROR	General device error.
0x701	1793	0x98110701	ADSERR_DEVICE_SRVNOTSUPP	Service is not supported by the server.
0x702	1794	0x98110702	ADSERR_DEVICE_INVALIDGRP	Invalid index group.
0x703	1795	0x98110703	ADSERR_DEVICE_INVALIDOFFSET	Invalid index offset.
0x704	1796	0x98110704	ADSERR_DEVICE_INVALIDACCESS	Reading or writing not permitted.
0x705	1797	0x98110705	ADSERR_DEVICE_INVALIDSIZE	Parameter size not correct.
0x706	1798	0x98110706	ADSERR_DEVICE_INVALIDDATA	Invalid data values.
0x707	1799	0x98110707	ADSERR_DEVICE_NOTREADY	Device is not ready to operate.
0x708	1800	0x98110708	ADSERR_DEVICE_BUSY	Device is busy.
0x709	1801	0x98110709	ADSERR_DEVICE_INVALIDCONTEXT	Invalid operating system context. This can result from use of ADS blocks in different tasks. It may be possible to resolve this through multitasking synchronization in the PLC.
0x70A	1802	0x9811070A	ADSERR_DEVICE_NOMEMORY	Insufficient memory.
0x70B	1803	0x9811070B	ADSERR_DEVICE_INVALIDPARM	Invalid parameter values.
0x70C	1804	0x9811070C	ADSERR_DEVICE_NOTFOUND	Not found (files, ...).
0x70D	1805	0x9811070D	ADSERR_DEVICE_SYNTAX	Syntax error in file or command.
0x70E	1806	0x9811070E	ADSERR_DEVICE_INCOMPATIBLE	Objects do not match.
0x70F	1807	0x9811070F	ADSERR_DEVICE_EXISTS	Object already exists.
0x710	1808	0x98110710	ADSERR_DEVICE_SYMBOLNOTFOUND	Symbol not found.
0x711	1809	0x98110711	ADSERR_DEVICE_SYMBOLVERSIONINVALID	Invalid symbol version. This can occur due to an online change. Create a new handle.
0x712	1810	0x98110712	ADSERR_DEVICE_INVALIDSTATE	Device (server) is in invalid state.
0x713	1811	0x98110713	ADSERR_DEVICE_TRANSMODENOTSUPP	AdsTransMode not supported.
0x714	1812	0x98110714	ADSERR_DEVICE_NOTIFYHNDINVALID	Notification handle is invalid.
0x715	1813	0x98110715	ADSERR_DEVICE_CLIENTUNKNOWN	Notification client not registered.
0x716	1814	0x98110716	ADSERR_DEVICE_NOMOREHDLS	No further handle available.
0x717	1815	0x98110717	ADSERR_DEVICE_INVALIDWATCHSIZE	Notification size too large.
0x718	1816	0x98110718	ADSERR_DEVICE_NOTINIT	Device not initialized.
0x719	1817	0x98110719	ADSERR_DEVICE_TIMEOUT	Device has a timeout.
0x71A	1818	0x9811071A	ADSERR_DEVICE_NOINTERFACE	Interface query failed.
0x71B	1819	0x9811071B	ADSERR_DEVICE_INVALIDINTERFACE	Wrong interface requested.
0x71C	1820	0x9811071C	ADSERR_DEVICE_INVALIDCLSID	Class ID is invalid.
0x71D	1821	0x9811071D	ADSERR_DEVICE_INVALIDOBJID	Object ID is invalid.
0x71E	1822	0x9811071E	ADSERR_DEVICE_PENDING	Request pending.
0x71F	1823	0x9811071F	ADSERR_DEVICE_ABORTED	Request is aborted.
0x720	1824	0x98110720	ADSERR_DEVICE_WARNING	Signal warning.
0x721	1825	0x98110721	ADSERR_DEVICE_INVALIDARRAYIDX	Invalid array index.
0x722	1826	0x98110722	ADSERR_DEVICE_SYMBOLNOTACTIVE	Symbol not active.
0x723	1827	0x98110723	ADSERR_DEVICE_ACCESSDENIED	Access denied.
0x724	1828	0x98110724	ADSERR_DEVICE_LICENSENOTFOUND	Missing license.
0x725	1829	0x98110725	ADSERR_DEVICE_LICENSEEXPIRED	License expired.
0x726	1830	0x98110726	ADSERR_DEVICE_LICENSEEXCEEDED	License exceeded.
0x727	1831	0x98110727	ADSERR_DEVICE_LICENSEINVALID	Invalid license.
0x728	1832	0x98110728	ADSERR_DEVICE_LICENSESYSTEMID	License problem: System ID is invalid.
0x729	1833	0x98110729	ADSERR_DEVICE_LICENSENOTIMELIMIT	License not limited in time.
0x72A	1834	0x9811072A	ADSERR_DEVICE_LICENSEFUTUREISSUE	Licensing problem: time in the future.
0x72B	1835	0x9811072B	ADSERR_DEVICE_LICENSETIMETOLONG	License period too long.

Hex	Dec	HRESULT	Name	Description
0x72C	1836	0x9811072C	ADSERR_DEVICE_EXCEPTION	Exception at system startup.
0x72D	1837	0x9811072D	ADSERR_DEVICE_LICENSEDUPLICATED	License file read twice.
0x72E	1838	0x9811072E	ADSERR_DEVICE_SIGNATUREINVALID	Invalid signature.
0x72F	1839	0x9811072F	ADSERR_DEVICE_CERTIFICATEINVALID	Invalid certificate.
0x730	1840	0x98110730	ADSERR_DEVICE_LICENSEOEMNOTFOUND	Public key not known from OEM.
0x731	1841	0x98110731	ADSERR_DEVICE_LICENSERESTRICTED	License not valid for this system ID.
0x732	1842	0x98110732	ADSERR_DEVICE_LICENSEDEMODENIED	Demo license prohibited.
0x733	1843	0x98110733	ADSERR_DEVICE_INVALIDFNCID	Invalid function ID.
0x734	1844	0x98110734	ADSERR_DEVICE_OUTOFRANGE	Outside the valid range.
0x735	1845	0x98110735	ADSERR_DEVICE_INVALIDALIGNMENT	Invalid alignment.
0x736	1846	0x98110736	ADSERR_DEVICE_LICENSEPLATFORM	Invalid platform level.
0x737	1847	0x98110737	ADSERR_DEVICE_FORWARD_PL	Context – forward to passive level.
0x738	1848	0x98110738	ADSERR_DEVICE_FORWARD_DL	Context – forward to dispatch level.
0x739	1849	0x98110739	ADSERR_DEVICE_FORWARD_RT	Context – forward to real time.
0x740	1856	0x98110740	ADSERR_CLIENT_ERROR	Client error.
0x741	1857	0x98110741	ADSERR_CLIENT_INVALIDPARM	Service contains an invalid parameter.
0x742	1858	0x98110742	ADSERR_CLIENT_LISTEMPTY	Polling list is empty.
0x743	1859	0x98110743	ADSERR_CLIENT_VARUSED	Var connection already in use.
0x744	1860	0x98110744	ADSERR_CLIENT_DUPLINVOKEID	The called ID is already in use.
0x745	1861	0x98110745	ADSERR_CLIENT_SYNCTIMEOUT	Timeout has occurred – the remote terminal is not responding in the specified ADS timeout. The route setting of the remote terminal may be configured incorrectly.
0x746	1862	0x98110746	ADSERR_CLIENT_W32ERROR	Error in Win32 subsystem.
0x747	1863	0x98110747	ADSERR_CLIENT_TIMEOUTINVALID	Invalid client timeout value.
0x748	1864	0x98110748	ADSERR_CLIENT_PORTNOTOPEN	Port not open.
0x749	1865	0x98110749	ADSERR_CLIENT_NOAMSADDR	No AMS address.
0x750	1872	0x98110750	ADSERR_CLIENT_SYNCINTERNAL	Internal error in Ads sync.
0x751	1873	0x98110751	ADSERR_CLIENT_ADDHASH	Hash table overflow.
0x752	1874	0x98110752	ADSERR_CLIENT_REMOVEHASH	Key not found in the table.
0x753	1875	0x98110753	ADSERR_CLIENT_NOMORESYM	No symbols in the cache.
0x754	1876	0x98110754	ADSERR_CLIENT_SYNCRESINVALID	Invalid response received.
0x755	1877	0x98110755	ADSERR_CLIENT_SYNCPORTLOCKED	Sync Port is locked.
0x756	1878	0x98110756	ADSERR_CLIENT_REQUESTCANCELLED	The request was cancelled.

RTime error codes

Hex	Dec	HRESULT	Name	Description
0x1000	4096	0x98111000	RTERR_INTERNAL	Internal error in the real-time system.
0x1001	4097	0x98111001	RTERR_BADTIMERPERIODS	Timer value is not valid.
0x1002	4098	0x98111002	RTERR_INVALIDTASKPTR	Task pointer has the invalid value 0 (zero).
0x1003	4099	0x98111003	RTERR_INVALIDSTACKPTR	Stack pointer has the invalid value 0 (zero).
0x1004	4100	0x98111004	RTERR_PRIOEXISTS	The request task priority is already assigned.
0x1005	4101	0x98111005	RTERR_NOMORETCB	No free TCB (Task Control Block) available. The maximum number of TCBs is 64.
0x1006	4102	0x98111006	RTERR_NOMORESEMAS	No free semaphores available. The maximum number of semaphores is 64.
0x1007	4103	0x98111007	RTERR_NOMOREQUEUES	No free space available in the queue. The maximum number of positions in the queue is 64.
0x100D	4109	0x9811100D	RTERR_EXTIRQALREADYDEF	An external synchronization interrupt is already applied.
0x100E	4110	0x9811100E	RTERR_EXTIRQNOTDEF	No external sync interrupt applied.
0x100F	4111	0x9811100F	RTERR_EXTIRQINSTALLFAILED	Application of the external synchronization interrupt has failed.
0x1010	4112	0x98111010	RTERR_IRQLNOTLESSOREQUAL	Call of a service function in the wrong context
0x1017	4119	0x98111017	RTERR_VMXNOTSUPPORTED	Intel VT-x extension is not supported.
0x1018	4120	0x98111018	RTERR_VMXDISABLED	Intel VT-x extension is not enabled in the BIOS.
0x1019	4121	0x98111019	RTERR_VMXCONTROLSMISSING	Missing function in Intel VT-x extension.
0x101A	4122	0x9811101A	RTERR_VMXENABLEFAILS	Activation of Intel VT-x fails.

Specific positive HRESULT Return Codes:

HRESULT	Name	Description
0x0000_0000	S_OK	No error.
0x0000_0001	S_FALSE	No error. Example: successful processing, but with a negative or incomplete result.
0x0000_0203	S_PENDING	No error. Example: successful processing, but no result is available yet.
0x0000_0256	S_WATCHDOG_TIMEOUT	No error. Example: successful processing, but a timeout occurred.

TCP Winsock error codes

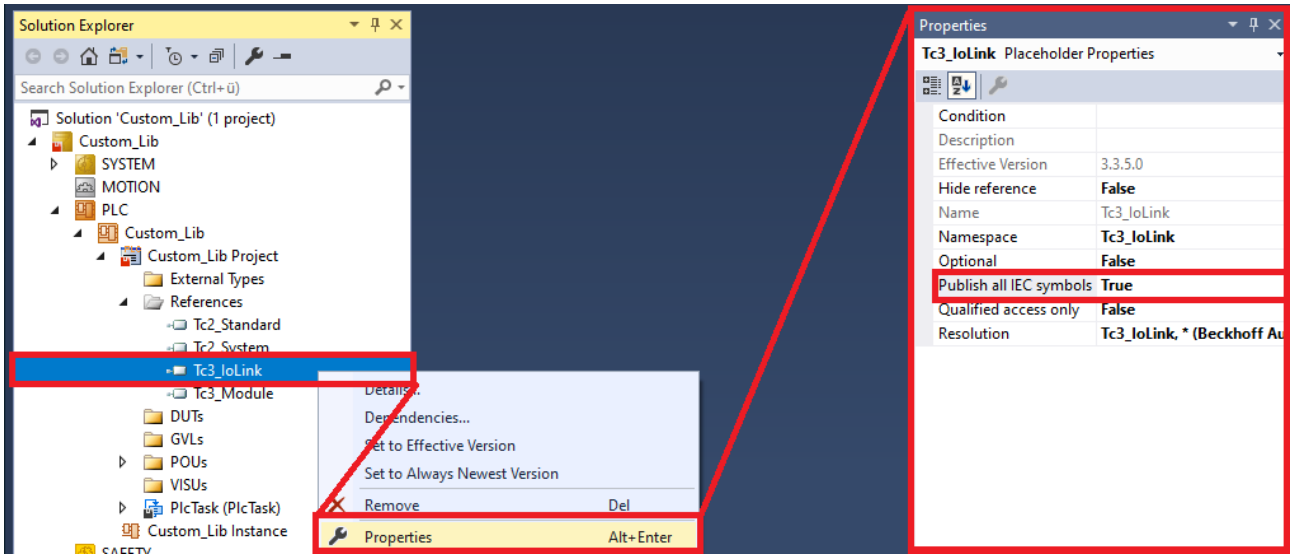
Hex	Dec	Name	Description
0x274C	10060	WSAETIMEDOUT	A connection timeout has occurred - error while establishing the connection, because the remote terminal did not respond properly after a certain period of time, or the established connection could not be maintained because the connected host did not respond.
0x274D	10061	WSAECONNREFUSED	Connection refused - no connection could be established because the target computer has explicitly rejected it. This error usually results from an attempt to connect to a service that is inactive on the external host, that is, a service for which no server application is running.
0x2751	10065	WSAEHOSTUNREACH	No route to host - a socket operation referred to an unavailable host.
More Winsock error codes: Win32 error codes			

8.3 Troubleshooting

Tc3_IoLink is being used within another library (Custom_Lib). During integration, the compiler does not find symbols from Custom_Lib.

By default, the symbols of an integrated lib are not published further.
This can be changed later.

To do this, open the Properties window and select **Publish all IEC symbols**:



For more information, refer to the PLC documentation in Command properties.

More Information:
www.beckhoff.com/te1000

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com