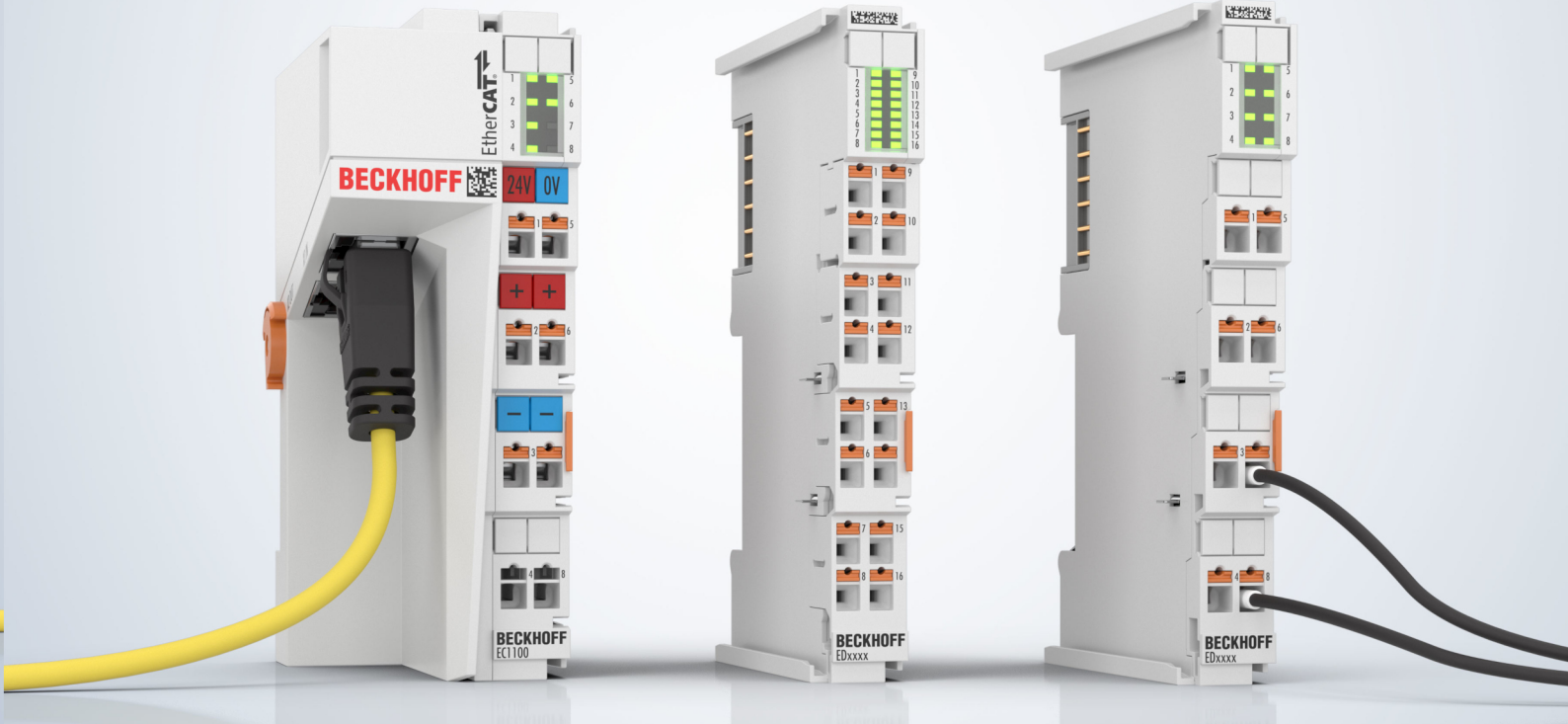
BECKHOFF New Automation Technology

Documentation | EN

ED6224

EtherCAT Terminal, 4-channel communication interface, IO-Link, master, push-in



2026-07-17 | Version: 1.1.0

Table of contents

1 Foreword

1.1 Product overview

This documentation covers the following products:

ED6224 [▶ 9]	EtherCAT Terminal, 4-channel communication interface, IO-Link, master, push-in

Use the tabular product overview or the product finder to find the right product for your application (https://www.beckhoff.com/IO).

1.2 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
The qualified personnel is obliged to always use the currently valid documentation.

The responsible staff must ensure that the application or use of the products described 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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® 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.

EtherCAT® and EtherCAT P® are registered trademarks and patented technologies, licensed by Beckhoff Automation GmbH, Germany.

Copyright

Third-party brands

Trademarks of third parties may be used in this documentation. You can find the trademark notices here:
https://www.beckhoff.com/trademarks

1.3 Guide through documentation

NOTICE

Further components of documentation

This documentation describes device-specific content. It is part of the modular documentation concept for Beckhoff I/O components. For the use and safe operation of the device / devices described in this documentation, additional cross-product descriptions are required, which can be found in the following table.

Title	Description
EtherCAT System Documentation (PDF)	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

The documentations can be viewed at and downloaded from the Beckhoff website (www.beckhoff.com) via:

- the “Documentation and Download” area of the respective product page,
- the Download finder,
- the Beckhoff Information System.

If you have any suggestions or proposals for our documentation, please send us an e-mail stating the documentation title and version number to: documentation@beckhoff.com

1.4 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

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.

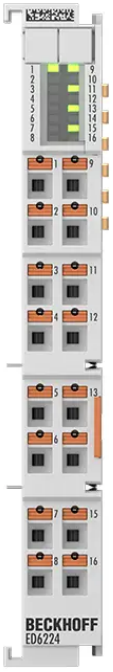
Warning of damage to property or environment

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.

2 Product description



EtherCAT Terminal, 4-channel communication interface, IO-Link, master, push-in

The ED6224 EtherCAT Terminal enables connection of up to four IO-Link devices. These can be sensors, actuators, or combined devices. A point-to-point connection is used between the terminal and the device. The terminal is parameterized via the EtherCAT master. IO-Link serves as an intelligent link between the fieldbus levels and the sensors or actuators, allowing process and parameterization data to be exchanged bidirectionally.

The parameterization of the IO-Link devices with service data can be carried out from TwinCAT via ADS or very conveniently via the integrated IO-Link commissioning tool.

In the standard setting, the ED6224 functions as a 4-channel input terminal (24 V DC) which communicates if necessary with connected IO-Link devices, parameterizes them, and, if applicable, changes their operation mode. Due to the integration in the HD housing with 16 connection points, each IO-Link device can be operated in 3-wire connection mode.

Additional connection points for 24 V DC and 0 V DC can be implemented using the potential distribution terminals of the ED918x series.

The ED-series EtherCAT Terminals have push-in connection technology which makes wiring easy and tool-free.

2.1 Technical data

General	ED6224
Number of channels | Total	4
Internal communication protocol	EtherCAT

Communication interface	ED6224
Number of ports	4
Interface	IO-Link
Protocol	IO-Link V1.1
Transmission type	serial
Device type	master
Transmission rate	4.8 kbaud, 38.4 kbaud, 230.4 kbaud
Number of devices	4
Device supply | Voltage	24 V DC
Device supply | Current	500 mA
Cable length | Point-to-point	max. 20 m

XFC	ED6224
Distributed clocks	yes
Distributed clocks | Accuracy	< 1 µs
Distributed clocks | Scope	64 bit
Timestamp	no
Oversampling	no

Supply and electrical isolation	ED6224
Electronics | Supply voltage	via E-bus, via power contacts
E-bus | Current consumption	typ. 120 mA
Power contacts | Input voltage	24 V DC (-15 %/+20 %)
Power contacts | Current carrying capacity	max. 10 A
Power contacts | Current consumption	load-dependent
Power contacts | Output voltage	corresponds to Power contacts | Input voltage
Electrical isolation | Channel/channel	no
Electrical isolation | Channel/bus	functional, 707 V DC (type test)

Environmental conditions	ED6224
Operating temperature	-25…+60°C
Storage temperature	-40…+85°C
Relative humidity	95%, no condensation
Installation position	variable

Standards and approvals	ED6224
Vibration resistance	conforms to EN 60068-2-6
Shock resistance	conforms to EN 60068-2-27
EMC immunity	conforms to EN 61000-6-2
EMC emission	conforms to EN 61000-6-4
Markings*)	CE
Approvals*)	-
*) Real applicable approvals/markings, see name plate on the side (product labeling)	

Housing data	ED-12-16pin
Weight	approx. 60 g
Protection rating	IP20
Material	polycarbonate
Dimensions | Width (single)	15 mm
Dimensions | Width (stacked)	12 mm
Dimensions | Height	100 mm
Dimensions | Depth	65 mm
Installation | Terminal/back wall	DIN rail, 35 mm (EN 60715)
Installation | Terminal/terminal	double slot and key connection
Pluggable wiring level	no
Connection mechanism	push-in
Connection cross-section | Solid wire	0.08…1.5 mm²
Connection cross-section | Stranded wire	0.25…1.5 mm²
Connection cross-section | With ferrule	0.14…0.75 mm²
Connection cross-section AWG | Solid wire	AWG28…16
Connection cross-section AWG | Stranded wire	AWG22…16
Connection cross-section AWG | With ferrule	AWG26…19
Stripping length	8…9 mm
Power contact | "+" (left)	yes
Power contact | "+" (right)	yes
Power contact | "-" (left)	yes
Power contact | "-" (right)	yes
Power contact | "⏚" (left)	no
Power contact | "⏚" (right)	no
Insertion lock for power contact "⏚"	yes

2.2 Connection and LEDs

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

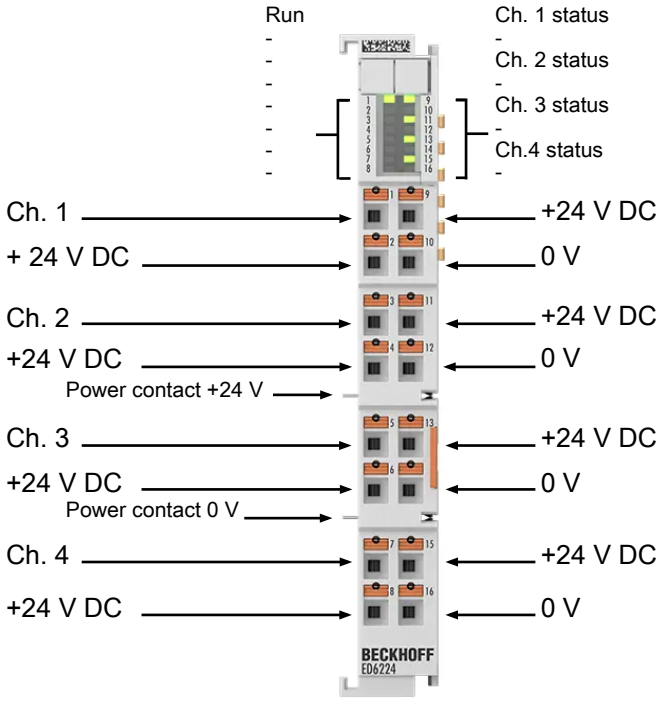


Fig. 1: ED6224

Connection

Terminal point		Description
Abbreviation	No.	
Ch 1	1	Input 1
+24 V DC	2	+24 V
Ch 2	3	Input 2
+24 V DC	4	+24 V
Ch 3	5	Input 3
+24 V DC	6	+24 V
Ch 4	7	Input 4
+24 V DC	8	+24 V
+24 V DC	9	+24 V
0 V	10	0 V
+24 V DC	11	+24 V
0 V	12	0 V
+24 V DC	13	+24 V
0 V	14	0 V
+24 V DC	15	+24 V
0 V	16	0 V

⚠ CAUTION

Risk of device damage!

The IO-Link devices must be powered from the 24 V power supply of the ED/EJ/EL6224 provided for this purpose, otherwise the IO-Link port may be damaged!

LEDs

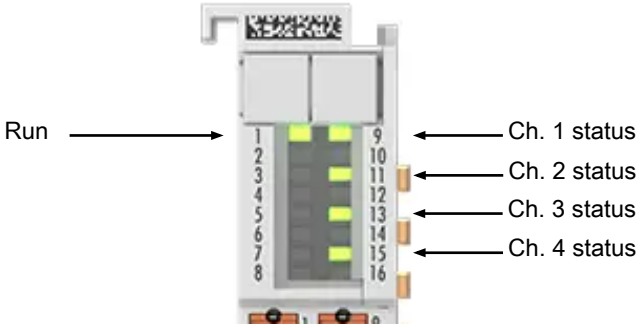


Fig. 2: ED6224 – LEDs

Name	No.	Color	Meaning	
RUN	1	green	This LED indicates the terminal's operating state:	
			off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal
			flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different default settings are applied
			single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager channels and the distributed clocks. Outputs remain in safe state
			on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
Ch. 1 - 4 status	9, 11, 13, 15	green	on / off	State of the signal line (if configured as STD in / out)
			2 short flashes	Establishing IO-Link communication
			flashing continuously	IO-Link communication established and in operation

3 Mounting and wiring

3.1 Notes on ESD protection

NOTICE

Devices can be destroyed by electrostatic charging!

The devices contain electrostatically sensitive components which can be damaged by improper handling.

- When handling the components, take care to avoid electrostatic discharge; also, prevent direct contact with the spring-loaded contacts (see illustration).
- Contact with highly insulating materials (synthetic fibers, plastic films, etc.) should be avoided when simultaneously handling components.
- When handling the components, ensure proper earthing of the environment (workplace, packaging, and persons).
- Each bus station must be terminated with the EL9011 or EL9012 end cap on its right side to guarantee the protection rating and ESD protection.

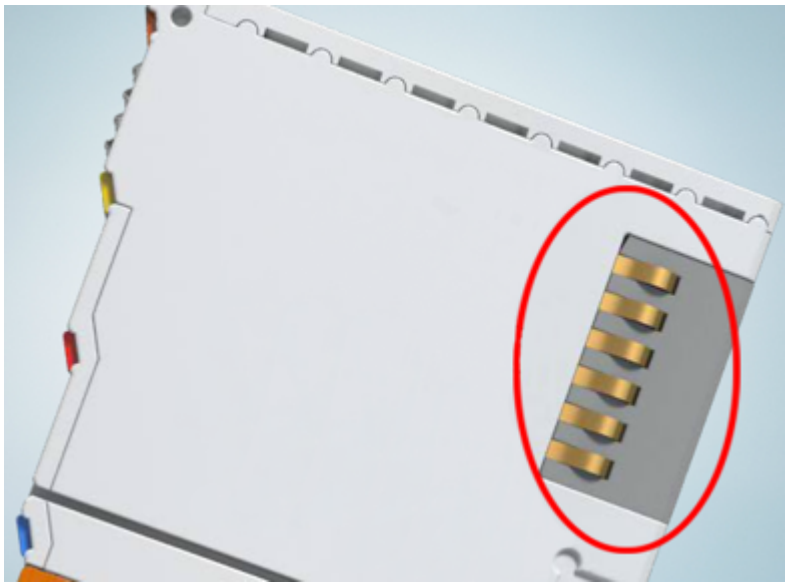


Fig. 3: Spring contacts of Beckhoff I/O components

3.2 Mounting rail installation

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

The bus terminal system is designed for installation in a control cabinet or junction box.

Installation

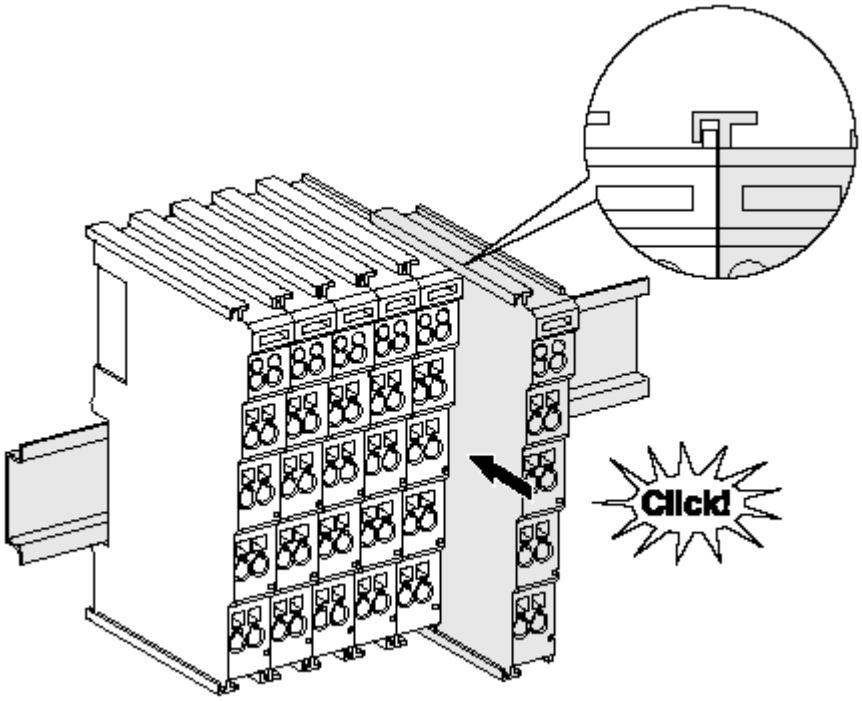


Fig. 4: Installation on the mounting rail

The bus couplers and bus terminals are attached to commercially available 35 mm mounting rails (DIN rail according to EN 60715) by applying slight pressure:

1. First attach the fieldbus coupler to the mounting rail.
2. The bus terminals are now attached on the right-hand side of the fieldbus coupler. To do this, connect the components with the slot and key and push the terminals against the mounting rail until the latch audibly engages on the mounting rail.
   If you snap the terminals onto the mounting rail first and then push them next to each other without the slot and key engaging, no functional connection will be established! When correctly assembled, no significant gap should be visible between the housing units.

Fastening the mounting rails

The locking mechanism of the terminals and couplers protrudes into the profile of the mounting rail. When installing the components, make sure that the locking mechanism doesn't come into conflict with the fixing bolts of the mounting rail. When fastening mounting rails with a height of 7.5 mm under the terminals and couplers, use flat fastening components such as countersunk head screws or blind rivets.

NOTICE

Eath the mounting rail!

Make sure the mounting rail is properly earthed.

Connections within a bus terminal block

The electric connections between the bus coupler and the bus terminals are made automatically by joining the components:

- The six spring-loaded contacts of the E-bus/K-bus deal with the data transfer and the supply of the bus terminal electronics.
- The power contacts deal with the supply for the field electronics and thus represent a supply rail within the bus terminal block. The power contacts are supplied via terminal points on the bus coupler (up to 24 V) or via power supply terminals for higher voltages.

Power contacts

When configuring a bus terminal block, note the contact assignments of the individual bus terminals, as some types

- do not loop through the power contacts or do not loop them through completely (e. g. analog bus terminals or digital 4-channel bus terminals),
- the power contacts are disconnected, thus marking the start of a new supply rail (power supply terminals).

Power contact ⏚

The power contact marked ⏚ (ground connection in accordance with IEC 60417-5017) can be used as a ground. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

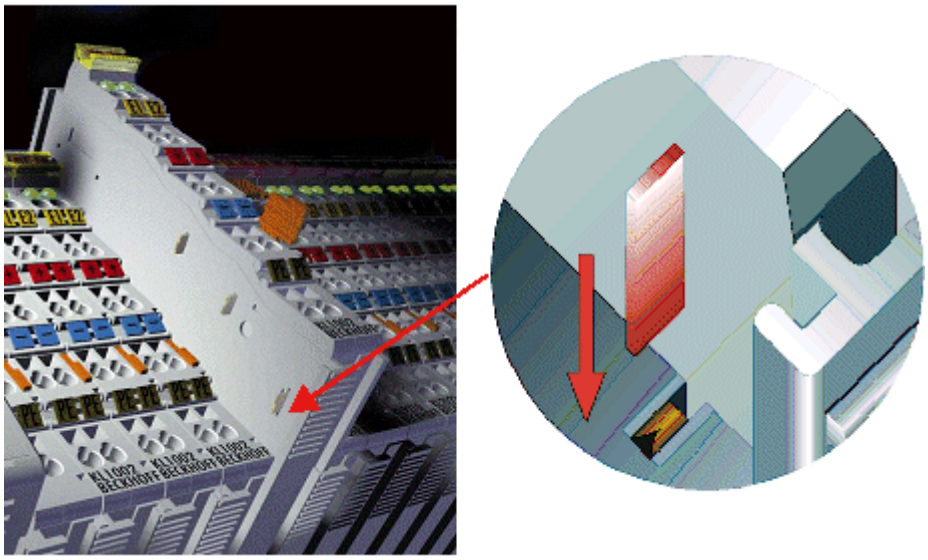


Fig. 5: Power contact on the left

⚠ WARNING

Risk of injury from electric shock!

The power contact marked ⏚ may not be used for other potentials!

NOTICE

Risk of damage to the device

Note that, for electromagnetic compatibility, the earth contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage to the terminal (e.g. disruptive discharge to the earth line during insulation testing of a consumer with a rated voltage of 230 V). For insulation testing, disconnect the earth supply line at the bus coupler or the power supply terminal! In order to decouple further supply points for testing, these power supply terminals can be released and pulled at least 10 mm from the group of terminals.

Disassembly

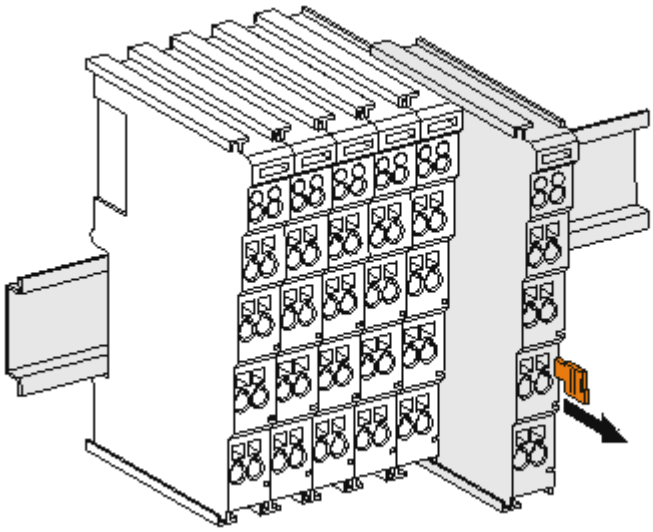


Fig. 6: Removal from mounting rail

Each terminal is secured by a lock on the mounting rail, which must be released for disassembly:

1. Pull down the terminal at its orange-colored straps from the mounting rail by approx. 1 cm. The rail locking of this terminal is automatically released, and you can now pull the terminal out of the bus terminal block with little effort.
2. To do this, grasp the unlocked terminal simultaneously at the top and bottom of the housing surfaces with your thumb and index finger and pull it out of the bus terminal block.

3.3 Positioning of passive terminals

Note on the placement of passive terminals in the bus terminal block

EtherCAT Terminals that do not take part in active data exchange within the bus terminal block are referred to as passive terminals. These terminals can be recognized because they do not consume power from the E-bus. To ensure optimal data exchange, no more than two passive terminals may be connected directly to each other!

Examples of the placement of passive terminals (highlighted in light color)

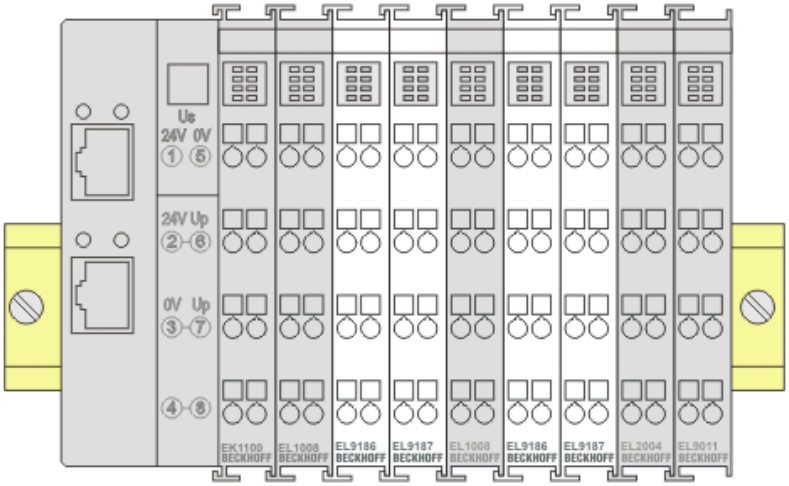


Fig. 7: Proper positioning

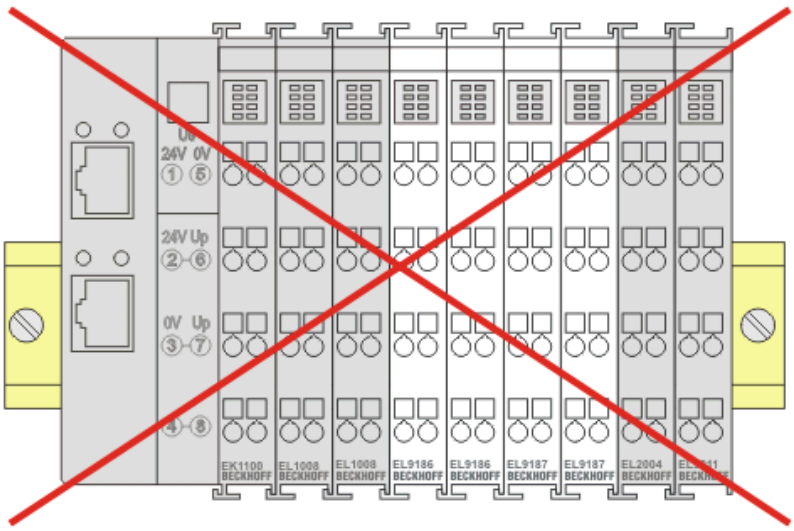


Fig. 8: Incorrect positioning

3.4 Installation positions

NOTICE

Constraints regarding installation position and operating temperature range

- Please refer to the technical data of the device to ascertain whether any restrictions regarding the installation position and/or the operating temperature range have been specified.
- When installing devices with increased heat dissipation, ensure that there is sufficient space above and below the devices during operation to guarantee adequate ventilation!

The installation positions and their names for mounting devices on mounting rails are specified below. The illustration of the devices in the following figures is an example.
The following applies to all installation positions: the reference direction “down” (see arrow) is the direction of the earth's gravitational force.

Horizontal installation (standard installation)

The mounting rail is mounted horizontally on a vertical mounting plate.
The connection level of the devices points to the front.

The devices are ventilated from below, which enables optimum cooling of the electronics through convection. This is therefore also the recommended installation position.

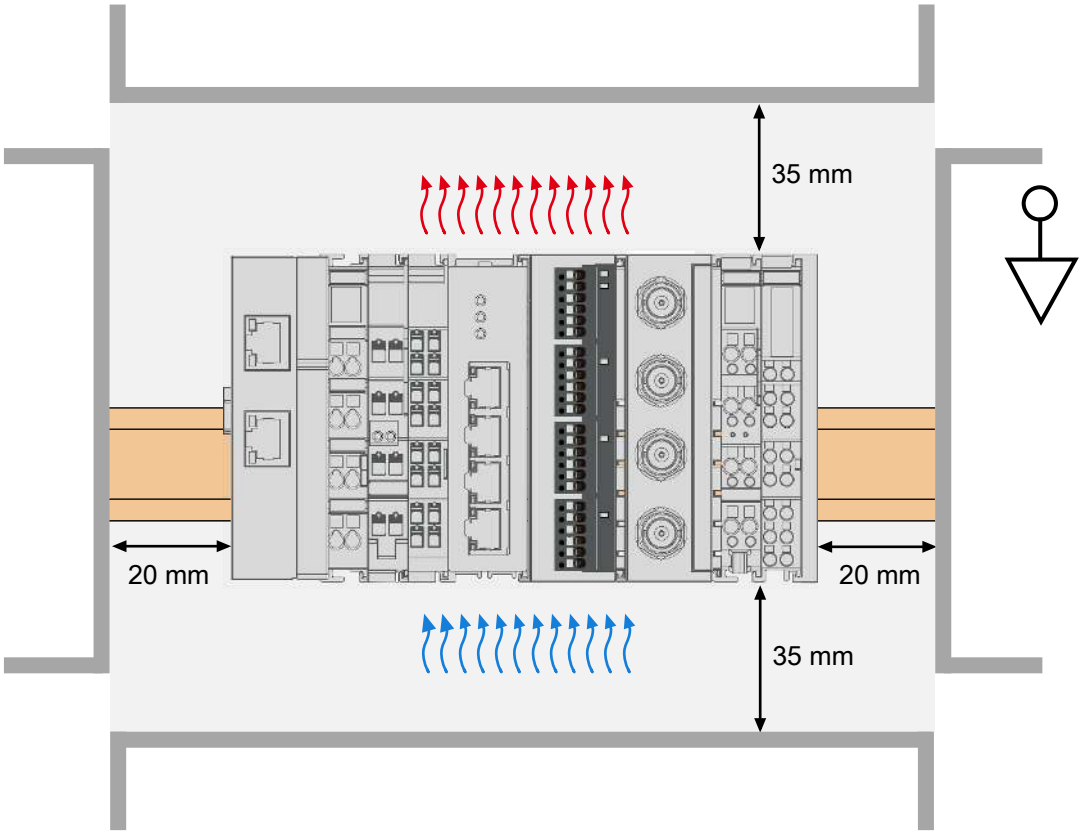


Fig. 9: Recommended minimum distances for standard installation position

NOTICE

Compliance with the minimum distances

Compliance with the minimum distances shown in the figure “Recommended minimum distances for standard installation position” is strongly recommended in all installation positions.

Vertical installation

The mounting rail is mounted vertically on a vertical mounting plate.
The connection level of the devices faces the front.
The devices can be arranged as follows:

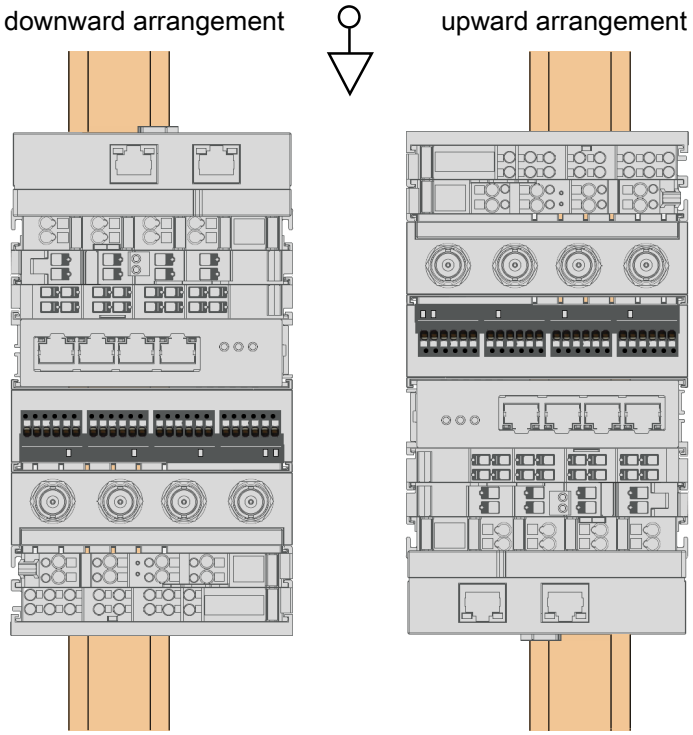


Fig. 10: Installation position: Vertical, downward arrangement (left)/upward arrangement (right)

Flat installation

The mounting rail is mounted on a horizontal mounting plate.
The connection level of the devices points upwards.

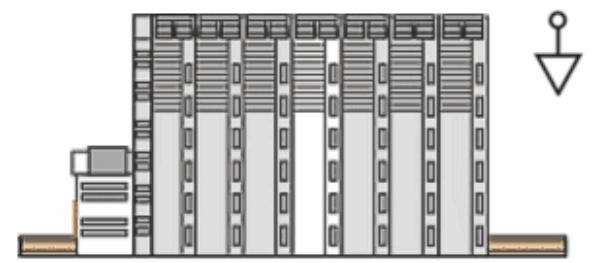


Fig. 11: Flat installation

NOTICE

Danger of sliding off the mounting rail

Particularly in the “vertical” installation position, but also in other installation positions under corresponding mechanical load, the terminal segment may move on the mounting rail. These can lead to undesirable malfunctions.

- If this risk exists, secure the terminal segment with appropriate locking devices, e.g. by clamping it to the mounting rail.

NOTICE

Compliance with the minimum distances

Compliance with the minimum distances shown in the figure “Recommended minimum distances for standard installation position” is strongly recommended in all installation positions.

Installation positions with ZB8610 fan cartridge

If the cooling should or must be increased for the intended application, the ZB8610 fan cartridge can be mounted on the underside of the device. In the horizontal installation position, the devices are ventilated from bottom to top by the fan cartridge. This optimum cooling is further enhanced by convection ventilation (see following figure).

The fan cartridge can be used in any installation position.

Further information on operation with and without a fan can be found in the technical data for the device (e.g. derating, information on installation positions, etc.).

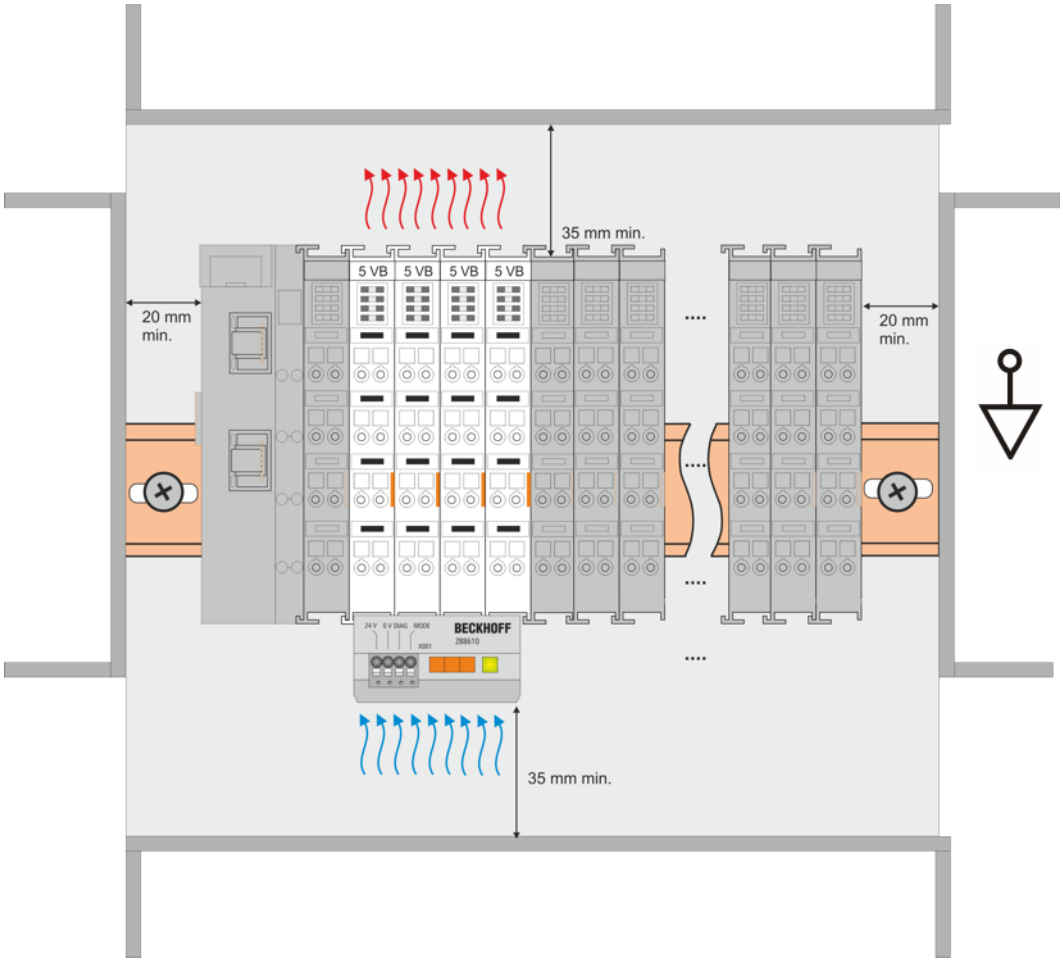


Fig. 12: Recommended minimum distances for operation with fan, using horizontal installation position as an example

NOTICE

Compliance with the minimum distances

Compliance with the minimum distances as shown in the figure “Recommended minimum distances for operation with fan” is strongly recommended.

3.5 Push-in connection technology (EC/ED/EFxxxx)

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

In the case of EC/ED/EFxxxx-series Terminals with push-in connection technology, solid conductors fitted with wire end ferrules can be plugged in directly without tools (see "Wiring [▶ 23]" section).

Ultrasonically compacted wires

Ultrasonically compacted (ultrasonically welded) wires can also be connected.

- Please note the information on the connection cross-section in the table housing data in the chapter Technical data of the relevant device!

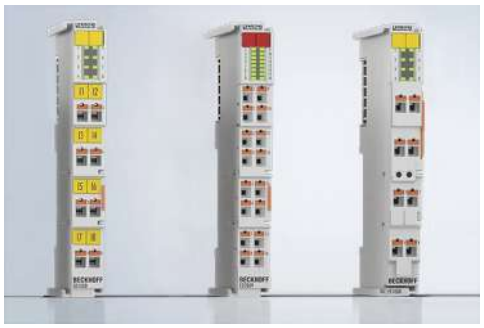


Fig. 13: Tool-free push-in connection; housing types ED-12-8pin, ED-12-16pin, and EF-12-8pin

The following connection options are available for optimum adaptation to the application:

- The EC/EDxxxx-series Terminals contain electronics and the connection level in one housing unit.
- The EFxxxx-series Terminals have a pluggable connection level. For servicing, the pluggable connection level allows the entire wiring to be removed from the top of the housing as a single plug. This reduces the installation time and eliminates the risk of wires being mixed up. The plug is only slightly protruding (for dimensions, see Technical data -> Housing data of the corresponding product). Installation and wiring are carried out as for the EDxxxx-series Terminals.
  Proceed as follows when replacing:
  - Press the unlocking hub and pull the lower part out of the terminal block.
  - Push in the replacement component and plug the connector with the upright wiring back in.

Strain relief of the cable

A tab for strain relief of the cable simplifies assembly in many applications and prevents entangling of the individual connection wires when the connector is pulled.

Overview of housing designs

EC/EDxxxx housing designs, contain the electronics and connection level in one housing	EFxxxx housing designs with pluggable wiring level	Description
EC-40-8pin	-	8 connection points on 40 mm
ED-12-8pin	EF-12-8pin	8 connection points on 12 mm
ED-24-2x8pin	EF-24-2x8pin	16 connection points on 24 mm
ED-12-16pin	-	16 connection points on 12 mm
ED-24-2x16pin	-	32 connection points on 24 mm

3.5.1 EC/ED/EFxxxx wiring with push-in connection technology

⚠ WARNING

Risk of injury through electric shock and damage to the device!

Bring the bus terminal system into a safe, de-energized state before beginning to mount, disassemble, or wire the bus terminals!

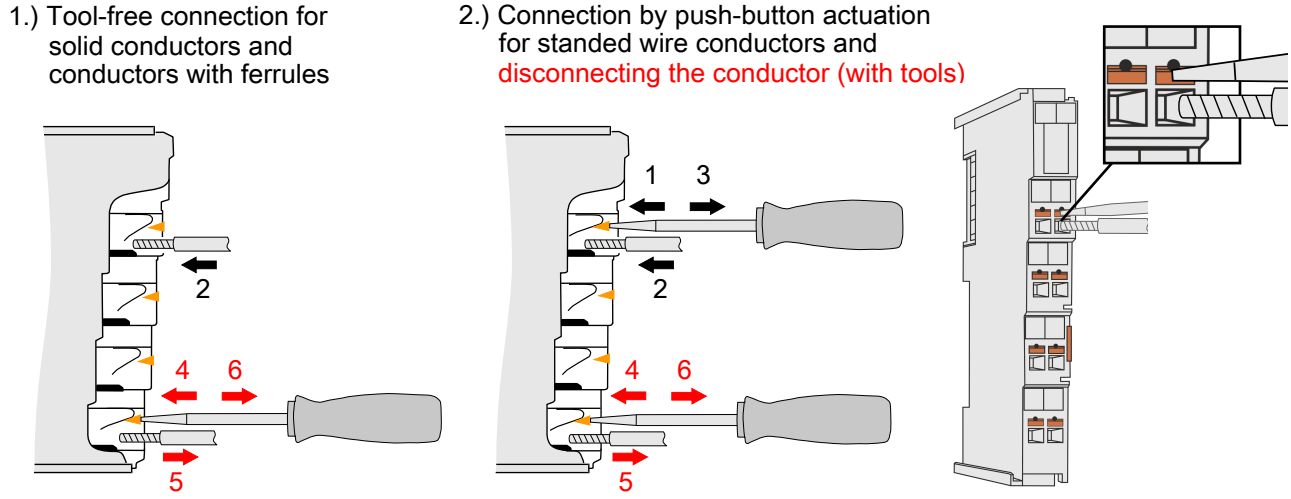


Fig. 14: Connecting and disconnecting the conductors

1. **Connecting solid conductors and conductors with ferrules (Fig. top left, step 2)**
   These can be connected using a direct plug-in technique, without tools.
   - Insert the conductor into the terminal point after stripping the insulation (2)
2. **Connect stranded cables (Fig. top right, steps 1–3)**
   These must be connected via the push-button.
   - Use a screwdriver to press the push-button to open the contact point (1).
   - Then insert the conductor (2).
   - Release the push-button by pulling back the screwdriver to close the terminal point (3).

Disconnect cables (Fig. above steps 4–6)
The release is performed using the push-button for all conductor types.

- Use a screwdriver to press the push-button to unlock the contact (4).
- Then pull out the conductor (5).
- Release the push-button by pulling back the screwdriver to close the terminal point (6).

NOTICE

Observe the permissible connection cross-sections and stripping lengths

The permissible conductor cross-section and stripping length depend on the type of housing, please refer to the housing data in the “Technical data” chapter for the device in question.

3.5.2 Shielding

Shielding

Encoders, analog sensors, and actuators should always be connected with shielded, pair-wise twisted cables.

3.6 Note on power supply

⚠ WARNING

Power supply from SELV / PELV power supply unit!

SELV / PELV circuits (safety extra-low voltage / protective extra-low voltage) according to IEC 61010-2-201 must be used to supply this device.

Notes:

- SELV / PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
- A SELV supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV supply also requires a safe connection to the protective conductor.

3.7 Disposal

Products marked with a crossed-out wheeled bin shall not be discarded with the normal waste stream. The device is considered as waste electrical and electronic equipment. The national regulations for the disposal of waste electrical and electronic equipment must be observed.

4 Commissioning

NOTICE

Further information

For information on commissioning with TwinCAT and EtherCAT basics, please refer to the EtherCAT system documentation.

⚠ CAUTION

Watchdog settings

Changes to the watchdog settings can lead to unwanted system behavior or damage to devices.

- Observe the information in the chapter “Notes on watchdog settings” in the EtherCAT system documentation before making any changes to the watchdog settings!

4.1 IO-Link basics

IO-Link is a communication system for connecting intelligent sensors and actuators to an automation system. The IEC 61131-9 standard specifies IO-Link under the designation "Single-drop digital communication interface for small sensors and actuators" (SDCI).

Both the electrical connection data and the communication protocol are standardized and summarized in the IO-Link Spec.

IO-Link specification

The development of the Beckhoff IO-Link master ED/EJ/EL6224 was subject to the IO-Link specification 1.1. At the time this documentation was created, the IO-Link specification was incorporated into the IEC standardization and adopted as IEC 61131-9 in an extended form. The new designation SDCI is also being introduced at the same time.
As a member of the respective committee, Beckhoff supports the development of IO-Link and reflects changes to the specification in its products.

4.1.1 IO-Link system layout

An IO-Link system consists of an IO-Link master and one or more IO-Link devices, i.e. sensors or actuators. The IO-Link master provides the interface to the higher-level controller and controls communication with the connected IO-Link devices.
The IO-Link masters from Beckhoff have several IO-Link ports, to each of which one IO-Link device can be connected. IO-Link is therefore not a fieldbus, but a point-to-point connection.

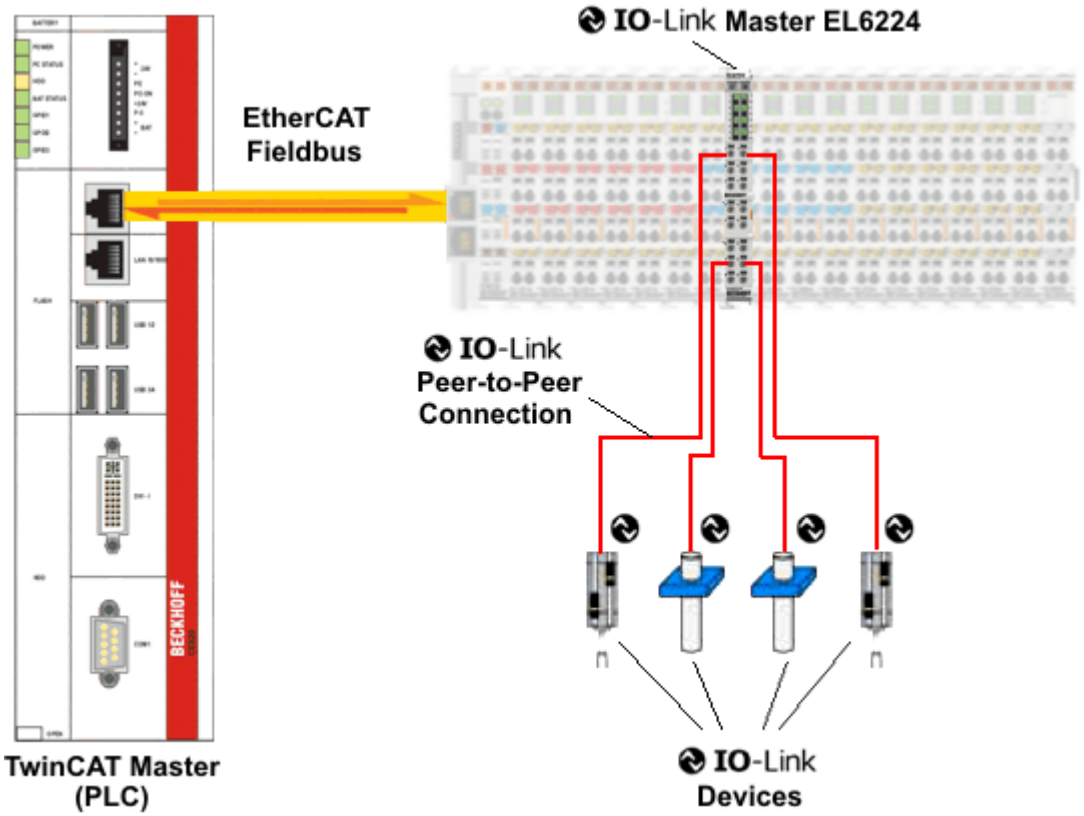


Fig. 15: IO-Link point-to-point communication using the example of the EL6224 IO-Link master

⚠ CAUTION

Risk of device damage!

The IO-Link devices must be powered from the 24 V power supply of the IO-Link master provided for this purpose, otherwise the IO-Link port may be damaged!

IO-Link application example

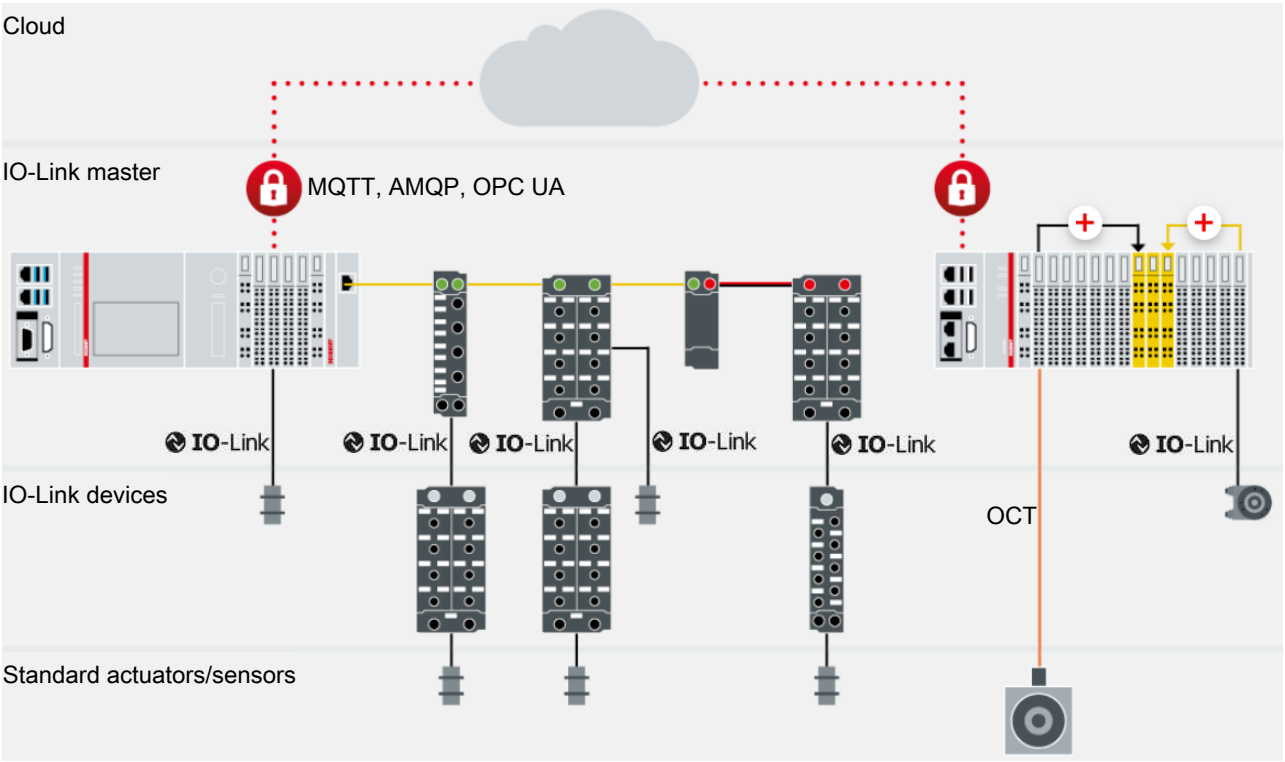


Fig. 16: Example of a control system with IO-Link integration

An overview of Beckhoff IO-Link products can be found on the Beckhoff website https://www.beckhoff.com//io-link

4.1.2 Establishment of IO Link communication

The establishment of the IO-Link communication is illustrated in Fig. *Establishment of IO-Link communication.* This illustrates in particular the sequence when automatically scanning [▶ 39] the IO-Link port.

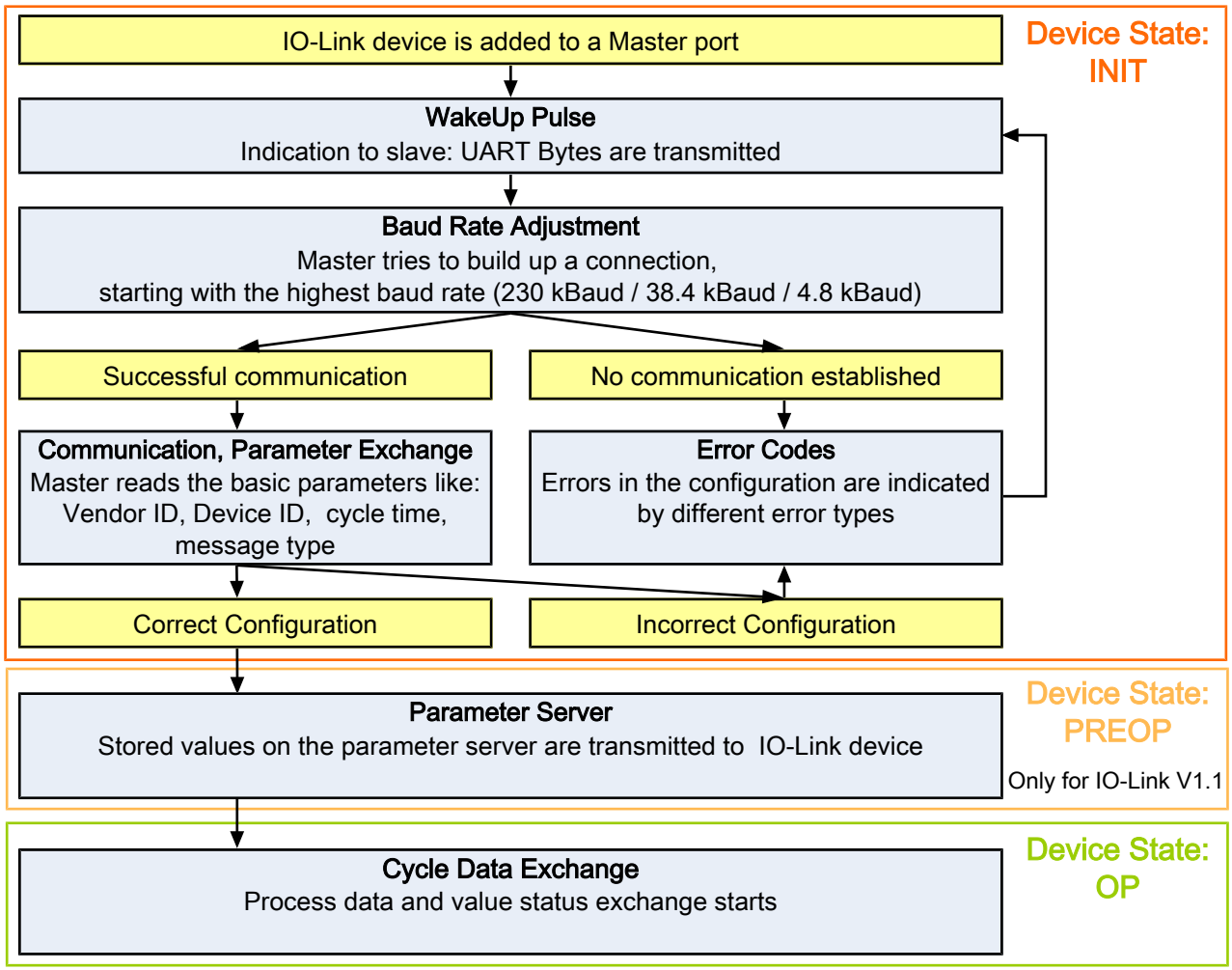


Fig. 17: Establishment of IO Link communication

- If an IO-Link device is connected to a master port, the master attempts to establish communication. A defined signal level, the **wake-up pulse**, signals to the device that UART bytes are to be sent from now on.
  From this point on, all data will be interpreted by the IO-Link device as UART bytes.
- The master runs through all baud rates [▶ 31], starting with the fastest baud rate (COM3 = 230 kbaud). A successful connection has been established when the slave responds to the wake-up pulse.
- First of all the master reads the **basic parameters** (Vendor ID, Device ID, process data length, telegram type and cycle time) and compares them with the existing configuration.
- If no connection could be established to the device, or if the saved parameters differ from those read, the corresponding error is output.
- If the saved parameters differ from those read, the IO-Link device changes to the PREOP state.
  If the IO-Link device specification is V1.1, the parameter server [▶ 30] is now executed. If the IO-Link device specification is V1.0, this step is omitted and the device changes directly to OP.
- Finally the cycle time is written and the device changes to OP. After that the master cyclically exchanges data with the device.

4.1.3 Device description IODD

IO-Link devices possess individual system information in the form of an IO device description (IODD), which contains:

- Communication features
- Device parameters with value range and default values
- Identification, process and diagnostic data
- Device data
- Text description
- Picture of the device
- Vendor's logo

If the IODD is imported, then the device data are automatically detected during automatic scanning [▶ 39] with TwinCAT and adopted in the System Manager.

4.1.4 Parameter server

In order to be able to use the functionality of the parameter server, both the IO-Link master and the IO-Link device must be specified to V1.1. The IO-Link revision of the device can be read for the individual port under Settings [▶ 45]. All IO-Link masters from Beckhoff with current firmware support the IO-Link specification V1.1.

- The parameter server in the IO-Link master contains parameter data that are saved in the IO-Link device. The memory capacity is max. 2 kbyte (including header).
  If the IO-Link device is exchanged, then the data are loaded from the parameter server into the new device. The requirement for this is that the device is of the same type (VendorID and DeviceID must be the same).
- If a new IO-Link device is configured, then the IO-Link master loads the parameters from the IO-Link device into the parameter server when starting for the first time.
  Data from other IO-Link devices that are already configured (VendorID and DeviceID do not correspond to the configured device) are overwritten.
- At each further start the IO-Link master uses a checksum to check whether the data in the parameter server correspond to those on the IO-Link device and if necessary downloads them to the device.
- If the parameters change during the device runtime, this can be reported to the Master via the store button [▶ 53] (ParamDownloadStore [▶ 54]). The master then starts the parameter server with an upload.
- By default the event is not set each time the parameters are written, therefore the end of the parameterization procedure has to be reported to the IO-Link device via the store button [▶ 53] (ParamDownloadStore [▶ 54]).
  The IO-Link device then sends the corresponding event to the master. The data are loaded into the parameter server.
- In the case of a pre-programmed IO-Link device, no download takes place from the parameter server to the device.

4.1.5 Data transfer rate

An IO-Link master according to specification V1.1 supports all three transmission methods and automatically adjusts the baud rate to that of the IO-Link device.
An IO-Link device usually supports only one baud rate. IO-Link devices with different baud rates can be connected to the various ports of the master.

- COM1 = 4.8 kbaud
- COM2 = 38.4 kbaud
- COM3 = 230.4 kbaud

4.2 IO link - Configuration and parameterizing

4.2.1 Configuring the IO-Link master

EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

When adding the IO-Link master in the TwinCAT System Manager (see chapter Setup in TwinCAT System Manager in the EtherCAT system documentation), an additional tab called “IO-Link” is created (see following figure). A more detailed description can be found in chapter Configuring IO-Link devices [▶ 33].

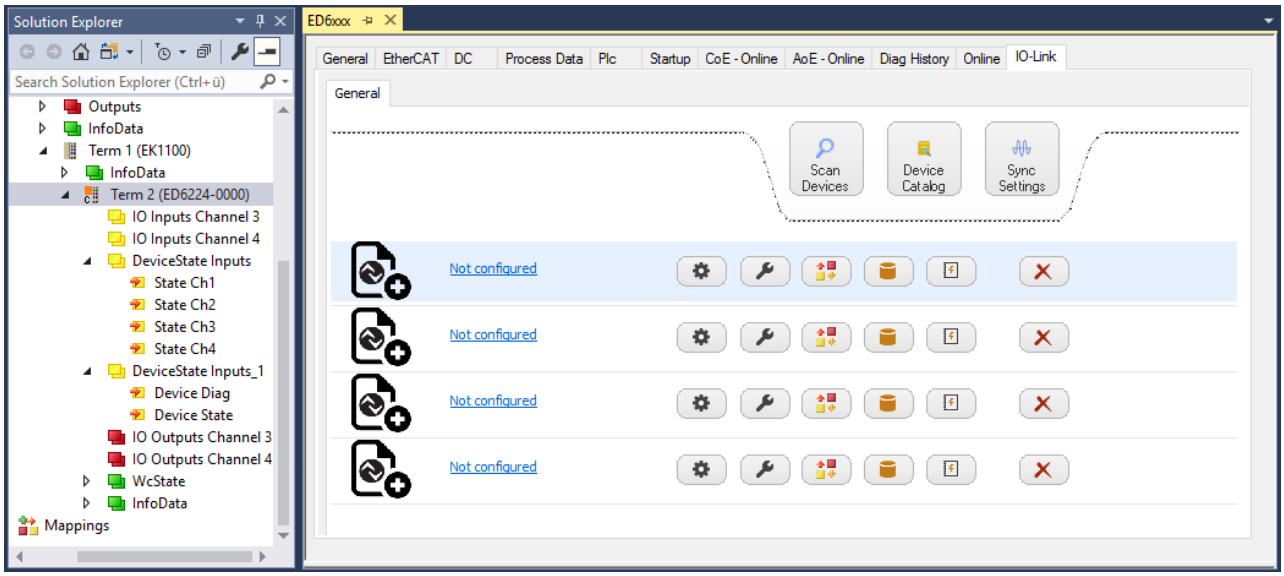


Fig. 18: “IO-Link” tab

IO-Link extension

If the “IO-Link” tab does not appear as shown in the figure above, the corresponding System Manager extension is missing. The System Manager extension is required for TwinCAT version 2.10, build 1325 to 1330.

- If your System Manager version or TwinCAT 3 does not yet have the current System Manager extension, it can be installed later if necessary. Please contact our Support. [▶ 86]

4.2.2 Configuring IO-Link devices

The configuration of the IO link devices is carried out in the IO link configuration tool. Configure the IO link device as described below.

The IO-Link devices are configured using the IO-Link configuration tool. Configure the IO-Link device as described in the following chapters:

✓ Prerequisite: an IO-Link master has been added in the Solution Explorer under the “I/O” entry.

1. Open the IO-Link configuration tool [▶ 33].
2. Include the IODD file of the IO-Link device [▶ 34].
3. Assign the devices to the ports
   ⇨ Assigning a device to a port [▶ 37]
   ⇨ Configuring a port as a digital input/output [▶ 38]
4. Removing an IO-Link device from a port [▶ 44]
5. Activate the IO-Link configuration [▶ 44] for the changes to take effect.

4.2.2.1 Open the IO link configuration tool

✓ Prerequisite: an IO-Link master has been added in the Solution Explorer under the “I/O” entry.

1. Double-click on the IO-Link master.
   ⇨ A device editor for the IO-Link master opens.
2. Click on the “IO-Link” tab.

⇨ The IO-Link configuration tool opens.

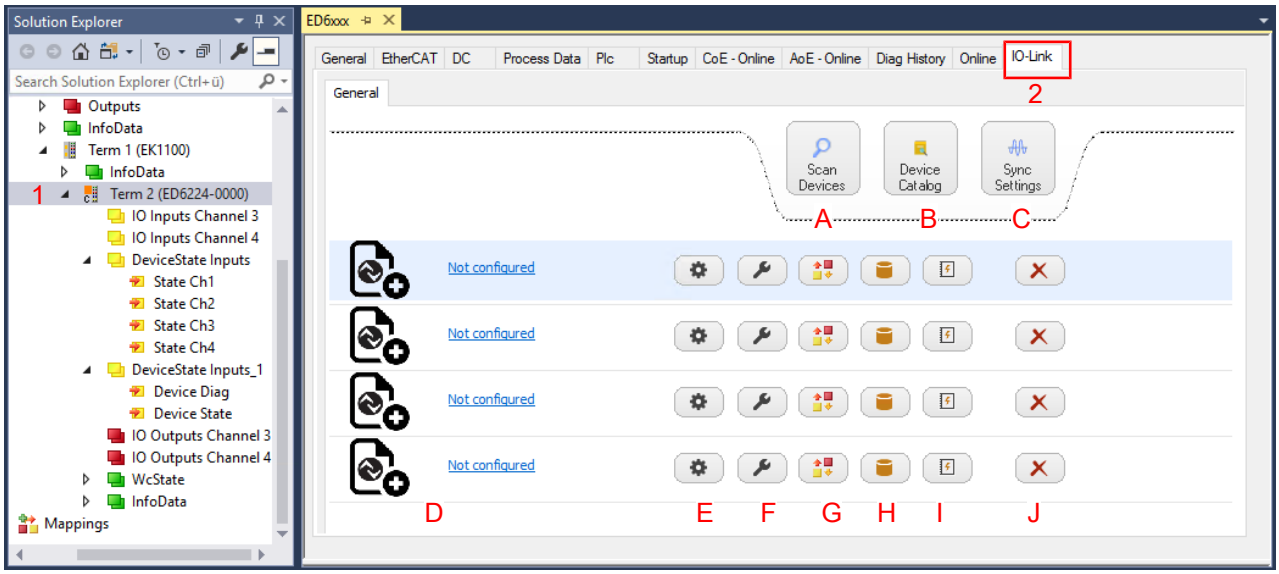


Fig. 19: IO-Link configuration tool

A: „Scan Devices [▶ 39]“	Automatic scanning for connected devices
B: “Device Catalog [▶ 34]”	Searching for and integrating device description (IODD), perform port assignment manually and configure IO-Link devices manually
C: “Sync Settings [▶ 60]”	Setting EtherCAT synchronization
D: Ports of the IO-Link master [▶ 37]	Devices assigned to the ports are displayed with product image and name
E: “Settings [▶ 45]”	Viewing and changing the settings of the IO-Link device
F: “Properties [▶ 47]”	Viewing and changing the parameters of the IO-Link device
G: “Process Data [▶ 58]”	Process data of the IO-Link device
H: “Data Storage [▶ 53]”	Displaying data stored in the parameter server
I: “Event log [▶ 59]”	Displaying errors and warning messages
J: “Remove device from port [▶ 44]”	Removing IO-Link devices from port n

4.2.2.2 Integrating IO-Link devices

The integration of the IODD file should always be the first step, since this enables the breakdown of the individual process data of the IO-Link device as well as the display of the parameters.

There are several ways of integrating an IO-Link device:

1. Open the “Device Management” dialog window via the “Device Catalog” button
2. Importing the IODD file (offline and online) via
   - ⇨ Import Devicedescription [▶ 36] button or
   - ⇨ “Online” tab with the IODD Finder [▶ 35]
3. Use the “Local” tab to select a device and assign it to a port [▶ 37]
4. Automatic scanning of the IO-Link ports (online) via
   - ⇨ Scan Devices [▶ 39] button
5. Manual insertion (offline and online) via
   - ⇨ “Create Device [▶ 43]” tab

Application note

- If the IODD is not available,
  the IO-Link device should be integrated online via “Scan Devices”.
- Manual integration via “Create Device” should only be carried out
  if neither the vendor’s IODD nor the IO-Link device are available at the time the project is created.

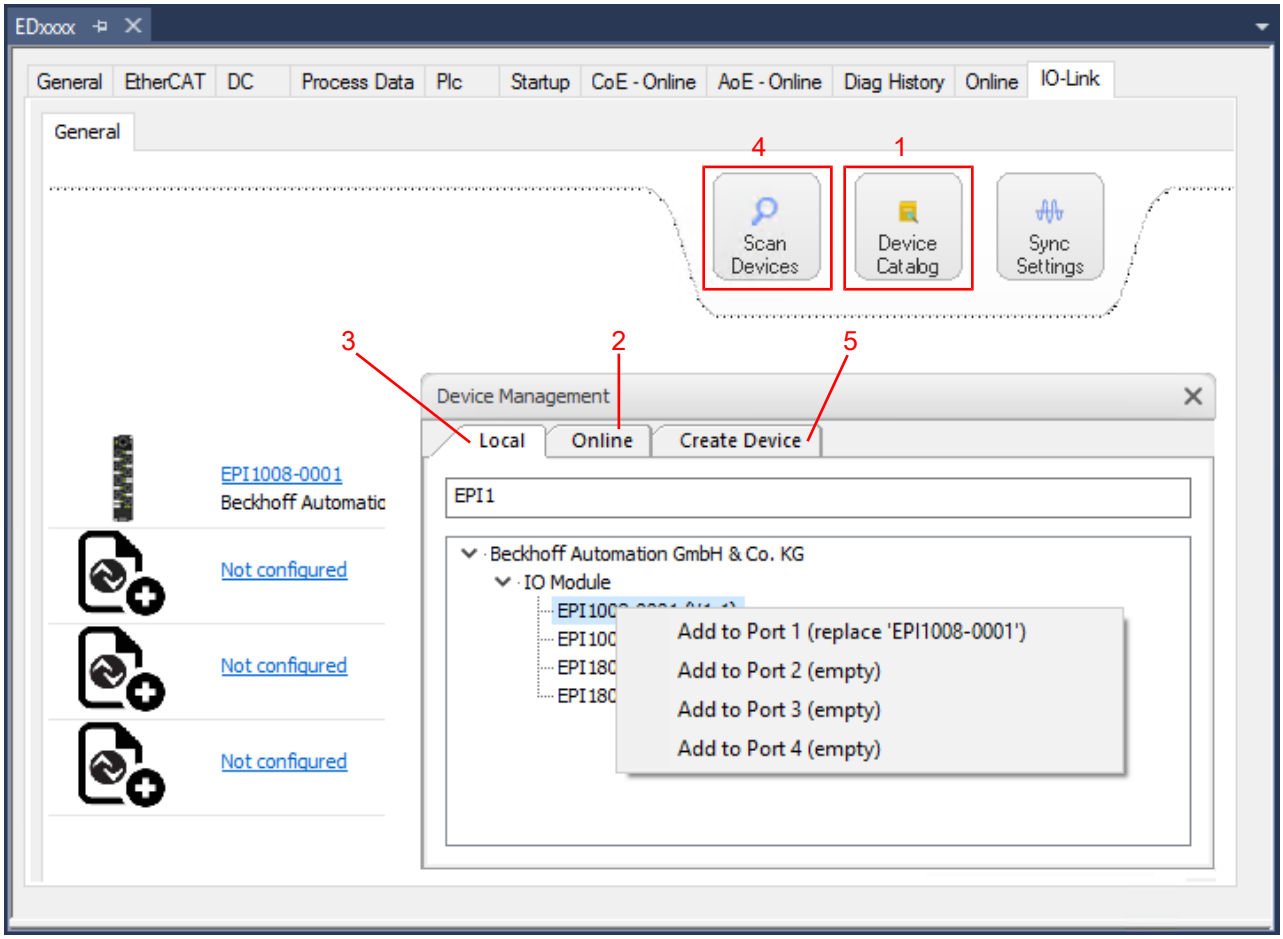


Fig. 20: Creating IO-Link devices

4.2.2.2.1 1. Importing the device description IODD

Importing the device description simplifies the integration of the IO-Link devices. The individual process data are broken down, enabling simple parameterization of the sensor. The IODD only needs to be imported during the initial commissioning of a new IO-Link device. The import is port-independent. Proceed as follows to import the IODD:

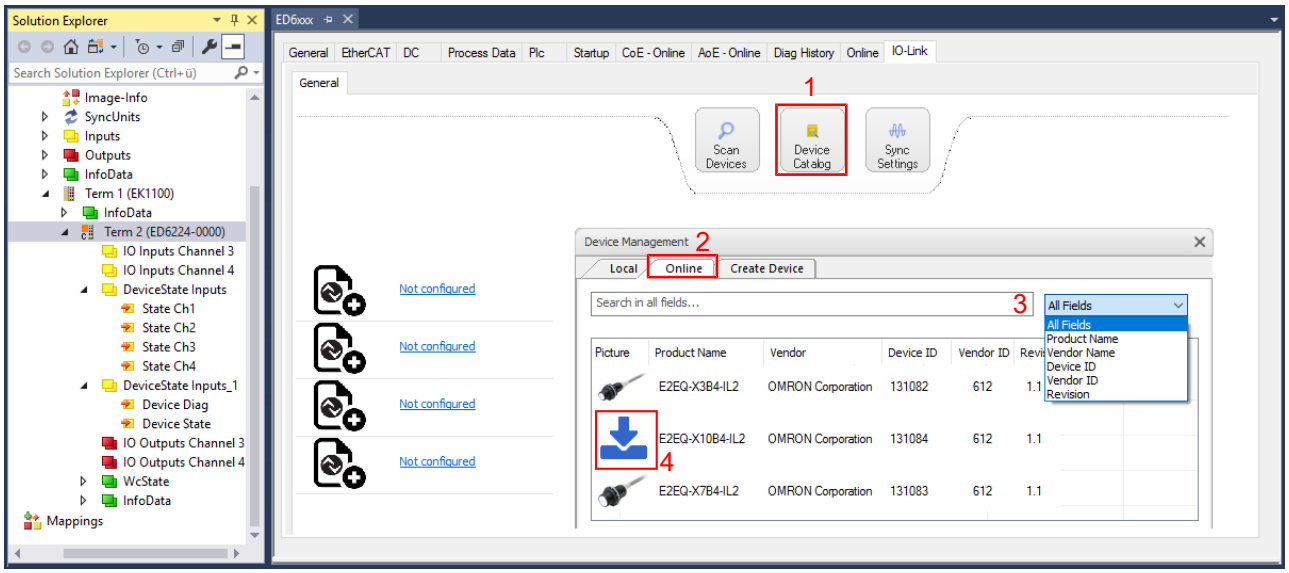


Fig. 21: Import of the IODD device description via “IODD Finder”

Integrating the IODD via the “IODD Finder”

1. Click on the “Device Catalog” button in the “IO-Link” tab (see figure above (1)).
   ⇨ The “Device Management” window will open.
2. Search for the desired IO-Link sensor/device by choosing the search field and entering the search term in the search mask (see figure above (3)).
3. Select the desired IO-Link sensor/device. Move the mouse pointer over the figure of the desired IO-Link sensor/device. A blue download icon appears, see figure above (4).
4. After clicking on the download icon, the .xml file of the selected IO-Link sensor/device is imported and stored in the following folder:
   - for TwinCAT 2.x: \TwinCAT\IO\IOLink
   - for TwinCAT 3.x: \TwinCAT\3.X\Config\IO\IOLink
5. When moving the mouse pointer over the IO-Link sensor/device, a green icon now indicates (see the following figure) that the .xml file already exists.
6. The imported device descriptions are listed in the “Local” tab in the tree structure, sorted by vendor.

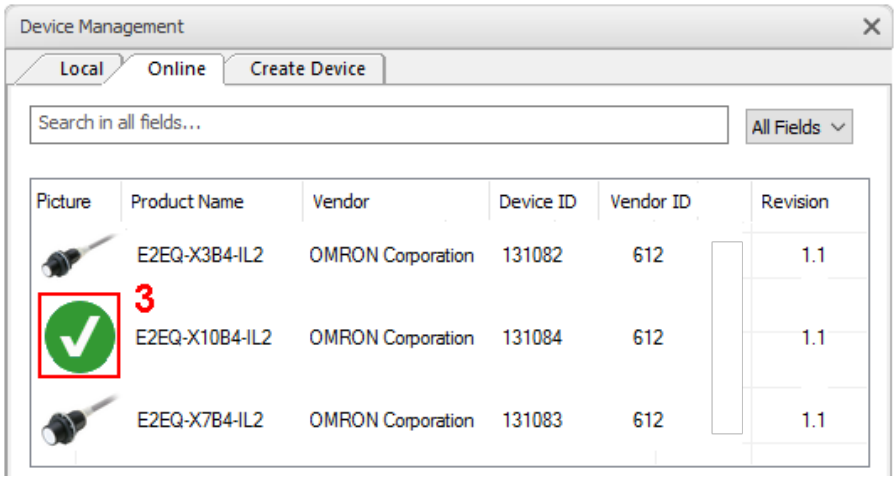


Fig. 22: IODD Finder, display of an already imported device description

Button "Import Devicedescription“

1. Press the “Import Devicedescription” button in the “IO-Link” tab
2. Select the .xml file of the desired sensor.
3. After pressing the Open button, the imported files are stored in the following folder:
   - for TwinCAT 2.x: \TwinCAT\IO\IOLink
   - for TwinCAT 3.x: \TwinCAT\3.X\Config\IO\IOLink.

⇨ The imported device descriptions are listed in a tree structure in the “Catalog” field, sorted by vendor.

No manual copying of the XML files

Do not copy the files directly into the folder; read them in via *Import Devicedescription* instead! Important checks will otherwise be bypassed!

4.2.2.2.2 2. Assigning IO-Link device to port n

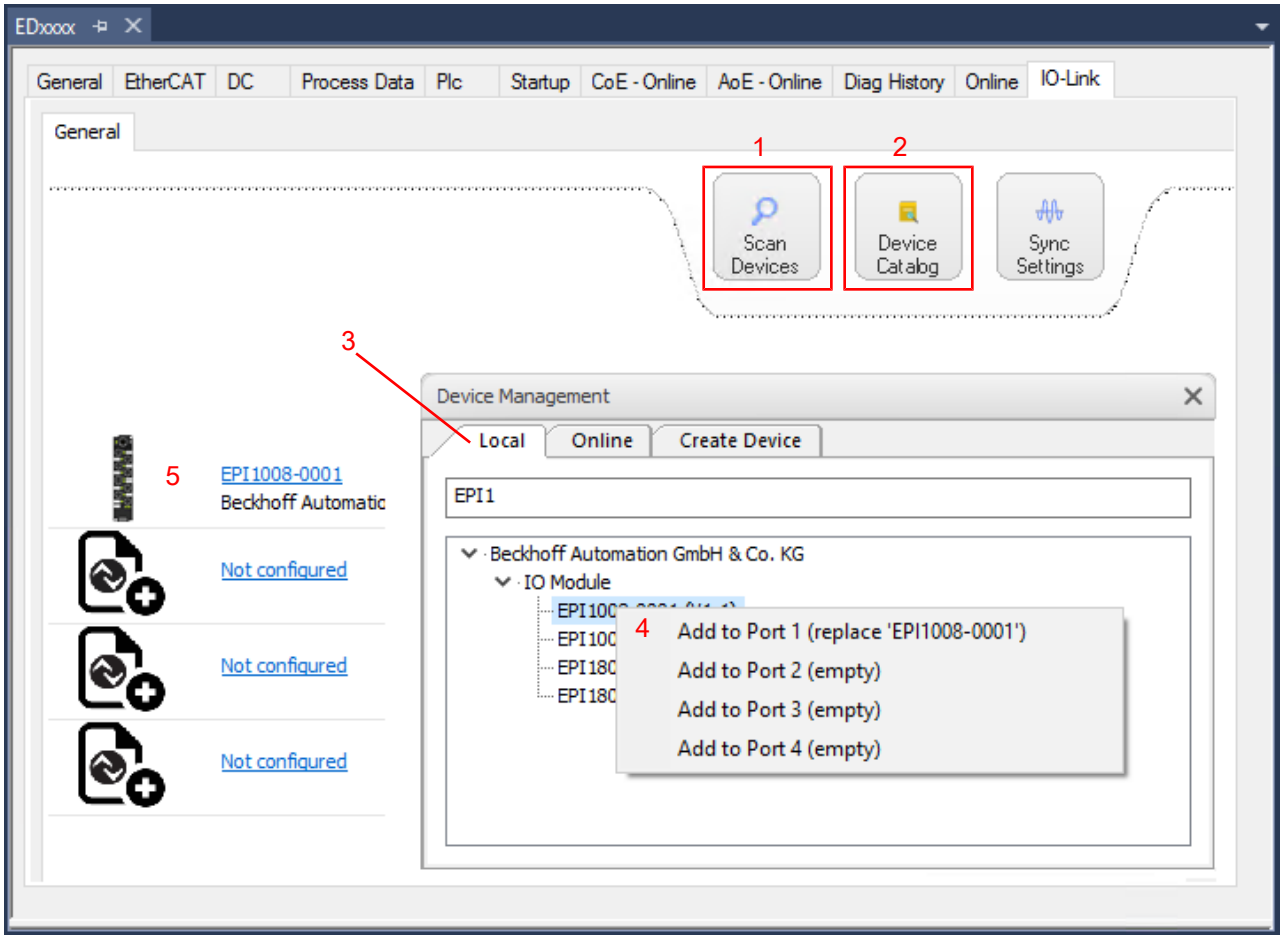


Fig. 23: Configuration using the example of the IO-Link device EPI1008-0001 on port 1

Online configuration

✓ Prerequisite: the IO-Link device is connected.

1. Click the “Scan Devices” button (see chapter Automatic scanning [▶ 39]).

⇨ The device is automatically detected and created with the corresponding parameters. If several devices are stored in the IODD file, the first entry is always selected here. Grouping in the IODD is usually carried out by the vendor if the process data is the same and there are only mechanical differences (e.g. different material).

Offline configuration

1. Click the “Device Catalog” button. The “Device Management” dialog is displayed. The “Local” tab displays the device descriptions that have already been imported, sorted by vendor in a tree structure.
2. Select the desired IO-Link device from the tree structure by
   - drag and drop: drag the device onto the port, or by
   - double-clicking on the device and clicking “Add to Port n”.

Activation of the configuration

3. Activate the IO-Link configuration [▶ 44] for the changes to take effect.

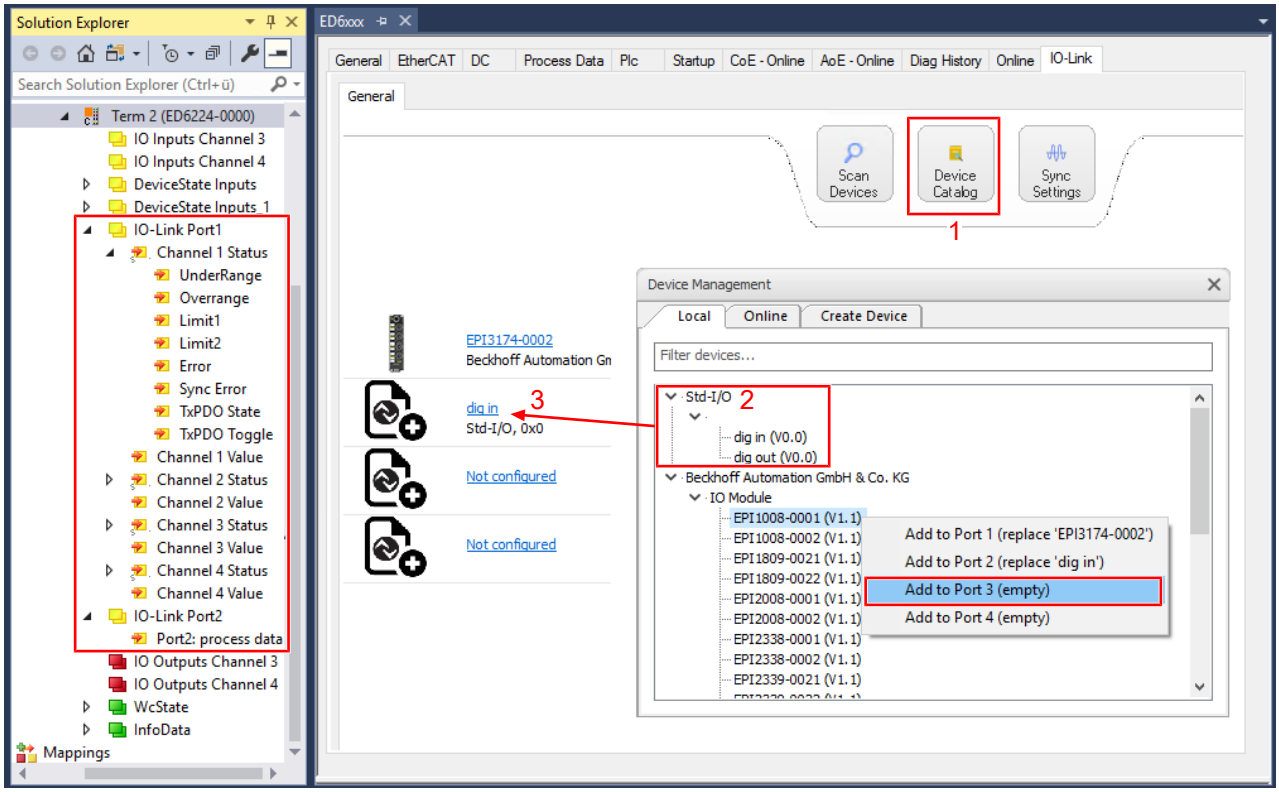
⇨ The IO-Link devices are displayed, and the process data is created. If an error is found when integrating the IO-Link device, e.g. wrong VendorID or no device connected, then this is indicated via the status of the port (object “State Ch.n” 0xF100:0n).

Configuration of the IO-Link ports as digital in- or output

IO-Link ports can also be configured as digital inputs or digital outputs. This allows digital sensors and actuators having no IO-Link functionality to be connected to IO-Link ports.

1. Click the “Device Catalog” button. The “Device Management” dialog is displayed.
2. Expand the “Std I/O” tree node in the “Local” tab.
   ⇨ The operation modes “dig in” and “dig out” appear.
3. Configure the desired port. There are two ways to do this, either by:
   - drag and drop: drag the device onto the port, or by
   - double-clicking on “dig in” / “dig out” and clicking “Add to Port n”.

Example of port assignment on the EL6224 IO-Link master



Port 1:
EPI3174-0002 is assigned

Port 2:
is configured as digital input

Port 3:
EPI1008-0001 is assigned via “Add to Port 3”

The process data for port 1 and port 2 are displayed in the Solution Explorer.

4.2.2.2.3 3. Automatic scanning of the IO-Link ports

This part of the documentation describes the configuration of the physically available IO-Link devices in TwinCAT.

During automatic scanning of the IO-Link ports, the steps “WakeUp pulse”, “Baud rate setting”, “Reading of the communication parameters”, plus “Parameter server” and “Cyclic data exchange”, if applicable, are performed, see Establishing the IO-Link communication [▶ 29]. The corresponding IO-Link device must be connected to the IO-Link port for this.

The connected devices are automatically detected, configured and a search is performed for the associated IODD.

Finding connected IO-Link devices

✓ Prerequisite: the master and the devices are connected and powered.

1. Click on the “Scan Devices” button (see following figure).

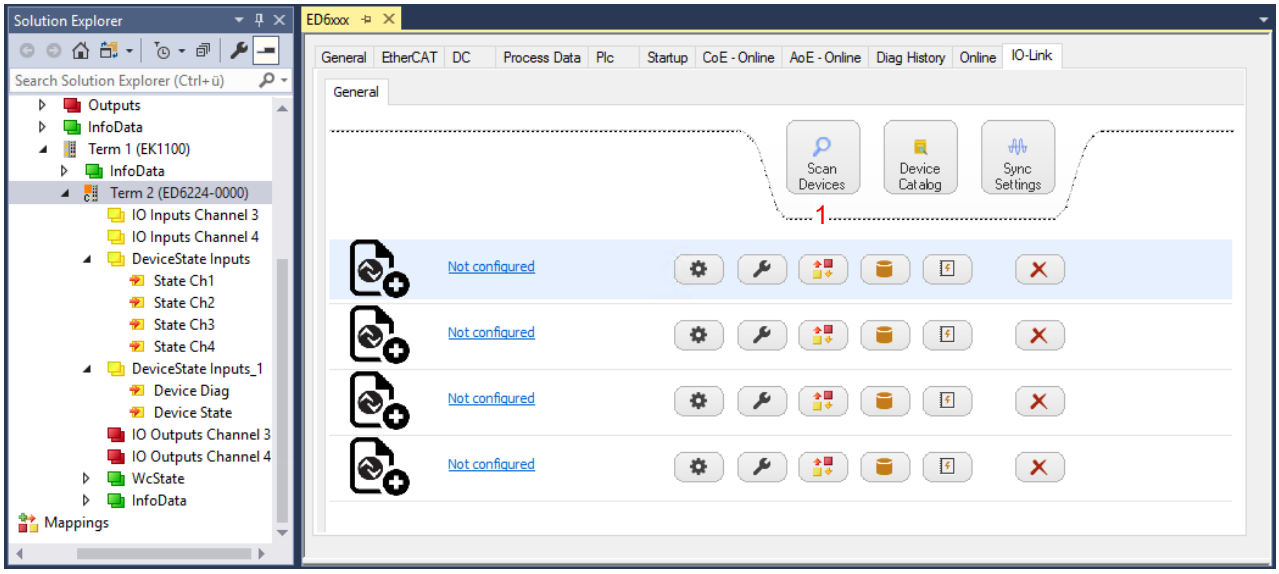


Fig. 24: Scan Devices

⇨ The connected IO-Link devices are found.

⇨ The connected device is listed for each port in the information window.
Only port 2 of the master is assigned an IO-Link device.

⇨ Confirm the information window with the “OK” button.

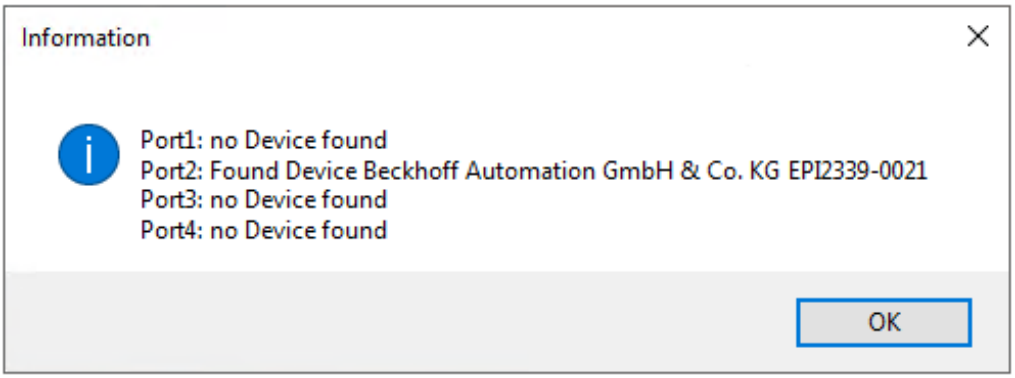


Fig. 25: Information “Scan Devices”

1. To be able to work with the devices, the button “Reload Devices” must be clicked. 

The IO-Link devices are now entered in the “General” tab. To the right of the port display, further tabs can be opened for information and settings for the IO-Link device.

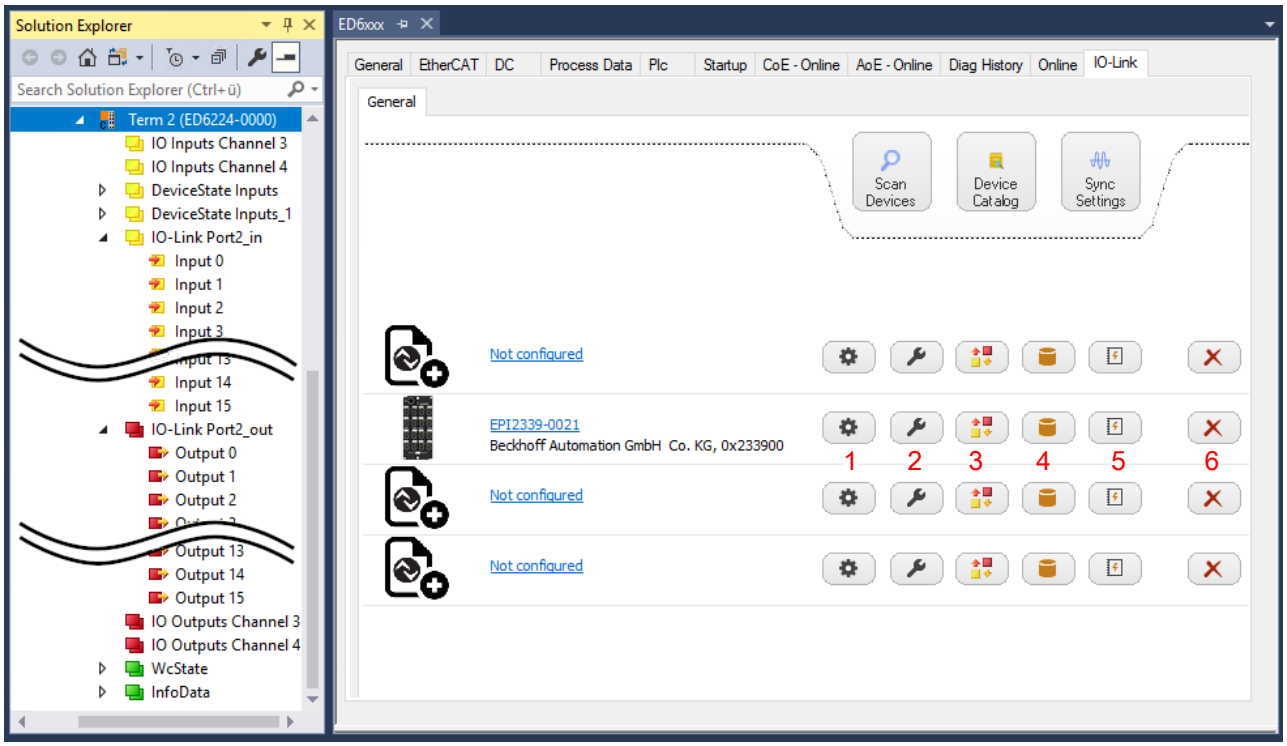


Fig. 26: Device on port 2, settings and information on the IO-Link device via buttons 1 - 6

1: “Settings [▶ 45]”	Viewing and changing the settings of the IO-Link device
2: “Properties [▶ 47]”	Viewing and changing the parameters of the IO-Link device
3: “Process Data [▶ 58]”	Process data of the IO-Link device
4: “Data Storage [▶ 53]”	Displaying data stored in the parameter server
5: “Event log [▶ 59]”	Displaying errors and warning messages
6: “Remove device from port [▶ 44]”	Removing IO-Link devices from port n

Displaying Device Settings

1. Open the “Settings” tab via the “Settings” button.
2. If necessary, change the settings in the “Settings” tab as described in the chapter Settings of the IO-Link devices [▶ 45].

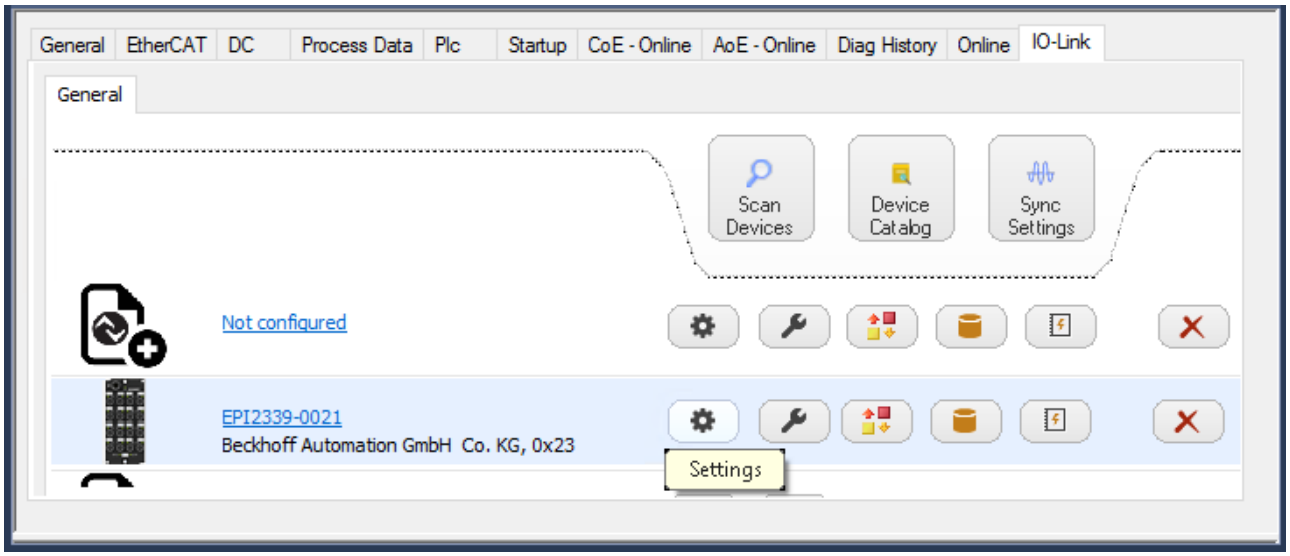


Fig. 27: Open the “Settings” tab for port 2

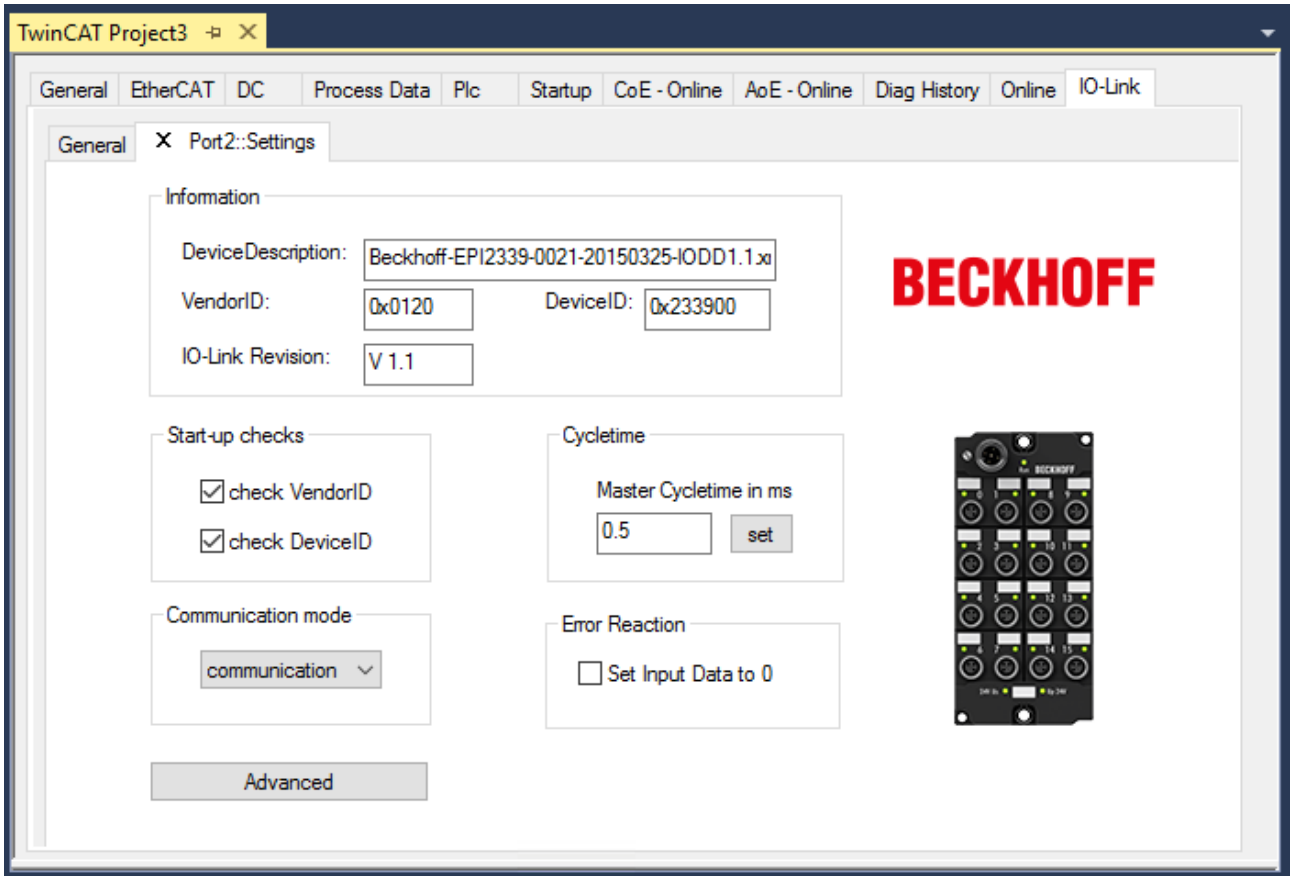


Fig. 28: Device port 2 settings

Displaying Device Parameters

1. Open the “Parameters” tab via the “Properties” button.
   ⇨ The parameters of the respective IO-Link device are listed.
2. Parameterize the device as described in chapter EPIxxxx, ERIxxxx - Setting the IO-Link device parameters [▶ 47].

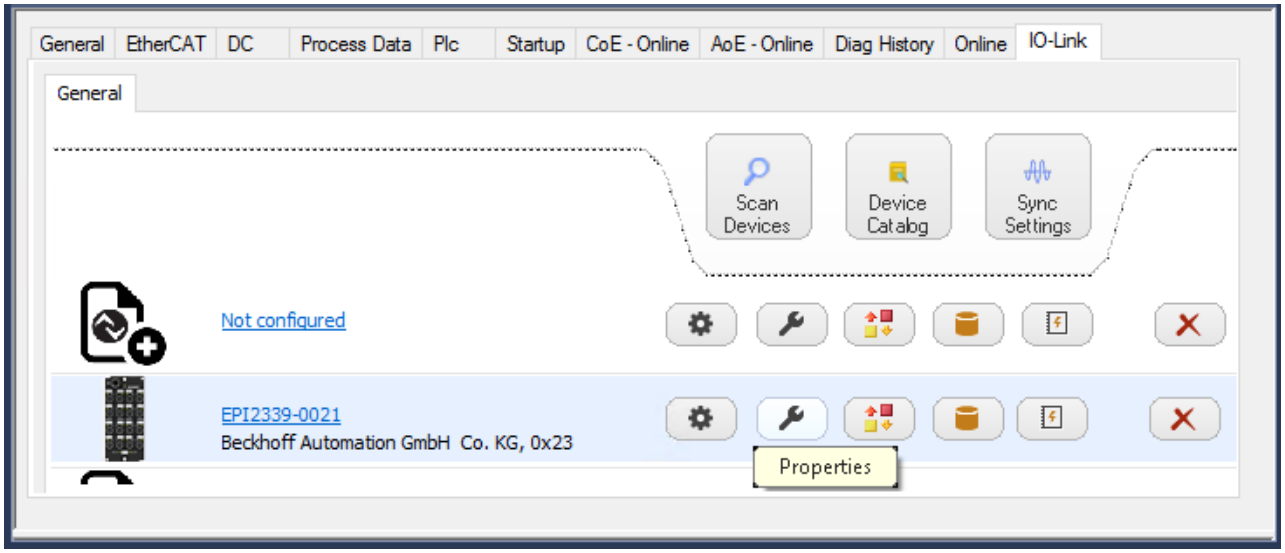


Fig. 29: Open the “Parameters” tab for port 2

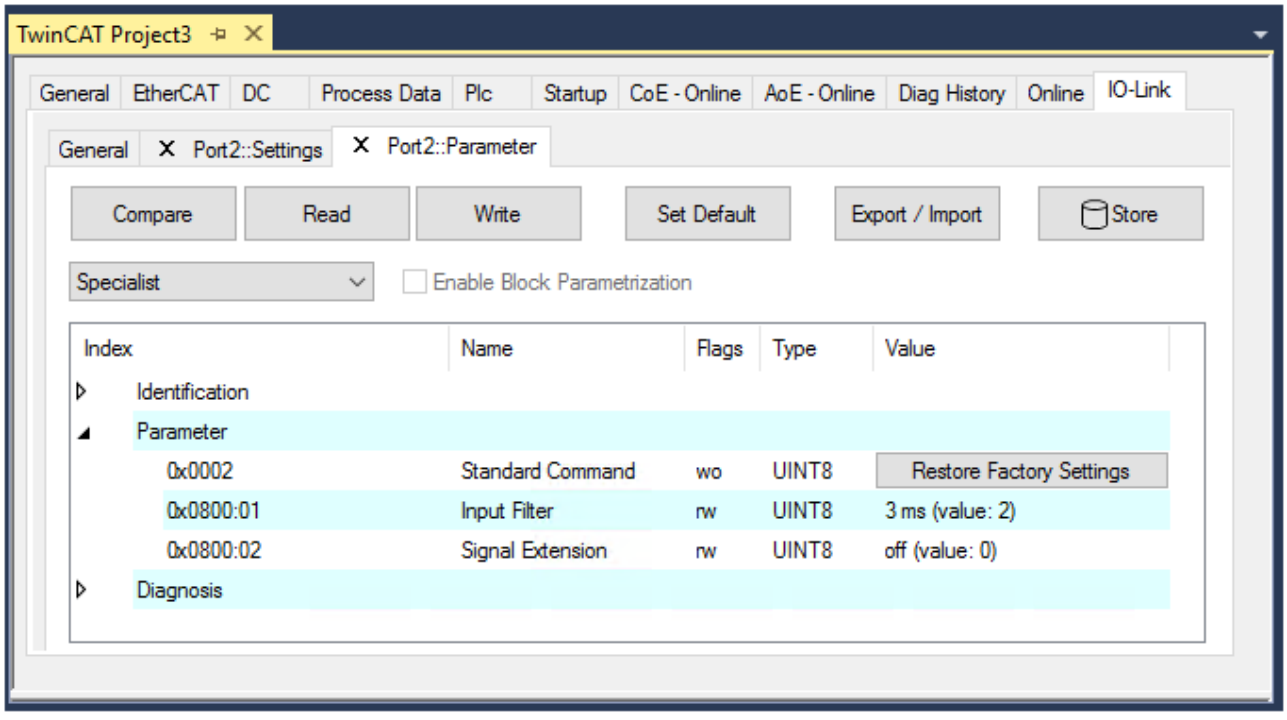


Fig. 30: Parameter device port 2

4.2.2.2.4 4. Manual insertion via Create Device

This part of the documentation describes the manual configuration of the IO-Link devices in TwinCAT.

The manual insertion of the IO-Link device should only be carried out if the IODD from the vendor and the IO-Link device are not available. By saving the project, the settings for the individual ports are saved. The devices created are **not** stored in the tree structure in the “Local” tab. To manually insert the IO-Link devices via the “Create Device” tab, proceed as follows:

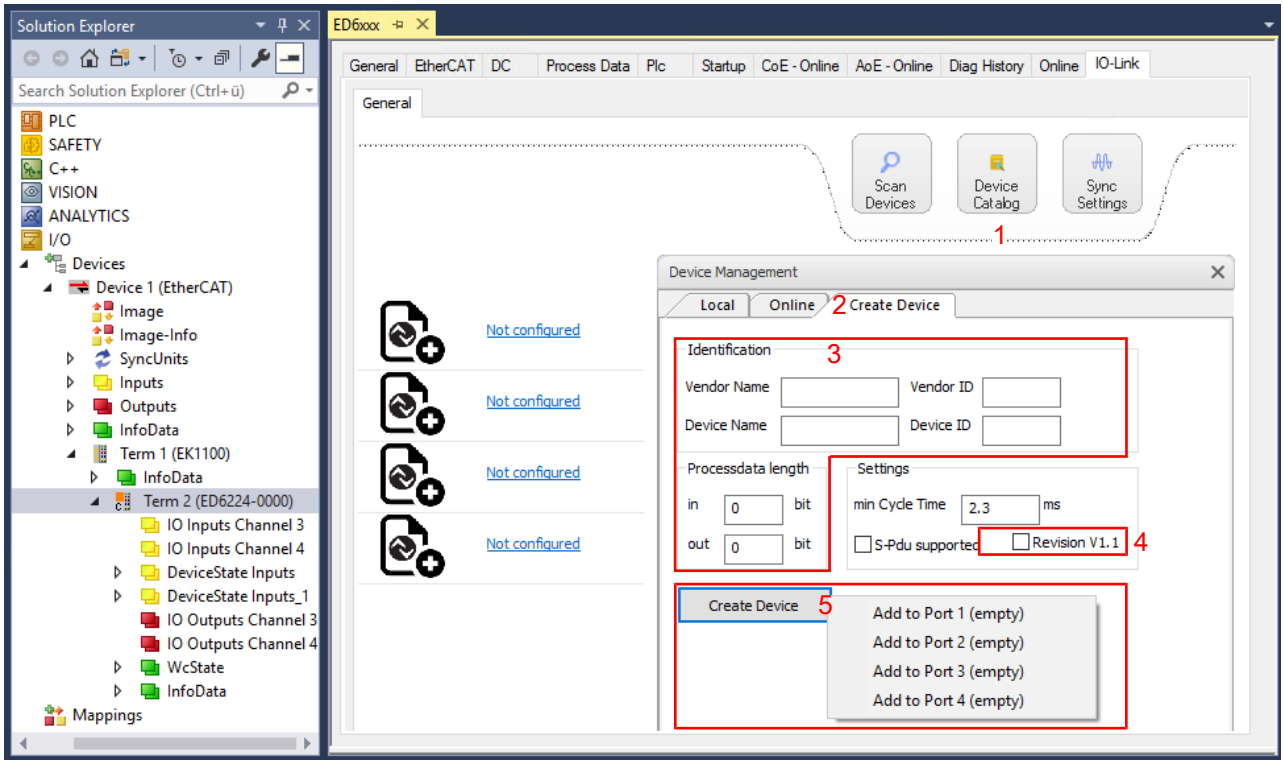


Fig. 31: Manual creation of an IO-Link device via the “Create Device” tab (2)

1. Click on the “Device Catalog” button to open the “Device Management” dialog.
2. Select the “Create Device” tab. This data is **not** stored in the tree structure on the “Local” tab and must be manually entered for each port.
3. Create an IO-Link device with the basic communication parameters in the “Create Device” dialog. Mandatory fields are: “Vendor ID”, “Device ID” and “Processdata length”, see figure above (3). The “Vendor ID” and “Device ID” values can be entered both as hexadecimal numbers (input format: 0xnnnn) and decimal numbers (nnnn).
   The communication parameters to be entered can be found in the information provided by the device vendor.
4. If the IO-Link device version is 1.1, then the parameter server is activated by the selection of the “Revision 1.1” field (see figure above (4)).
5. Press the “Create Device” button and assign the IO-Link device to a port (see figure above (5)).
6. Activate the IO-Link configuration [▶ 44] for the changes to take effect.

Reading the IODD

Even when manually creating and scanning, the IODD should always be read as well in order to display further sensor-specific information.

7. Further settings can be made in the IO-Link device settings as described in the IO-Link device settings [▶ 45] chapter.

4.2.2.3 Removal of IO-Link devices

To remove a device that has already been inserted, proceed as follows.

1. Select the port from which the device should be removed.
2. Click the “Remove Device from port” button.

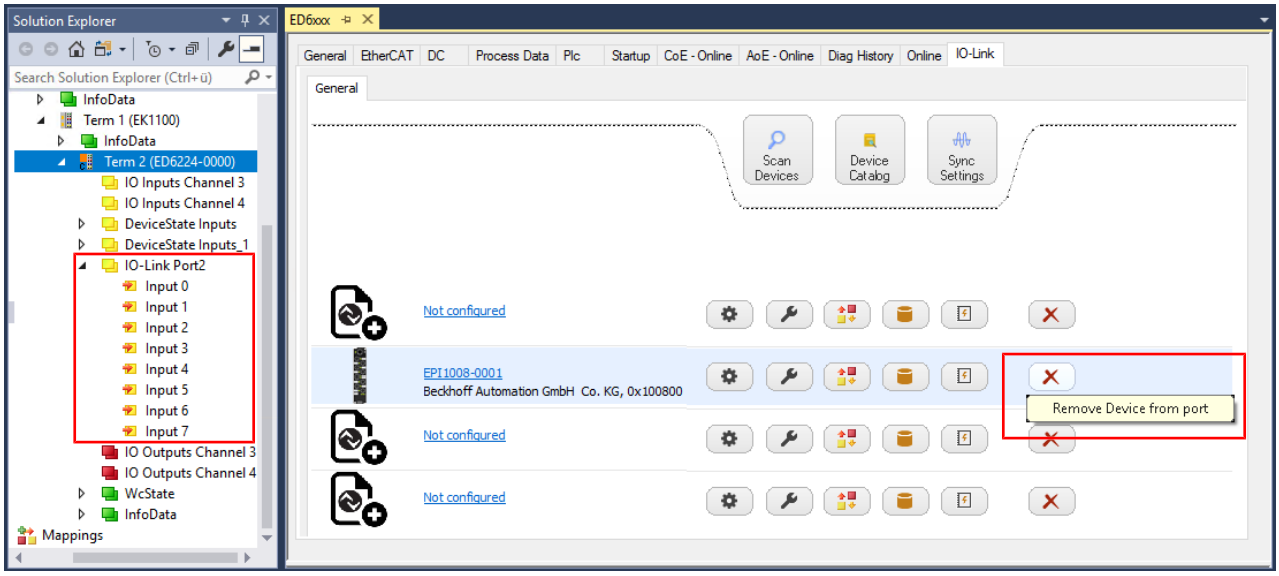


Fig. 32: Removing the device on port 2.

3. Activate the IO-Link configuration [▶ 44] for the changes to take effect.

⇨ The process data that has already been created is removed.

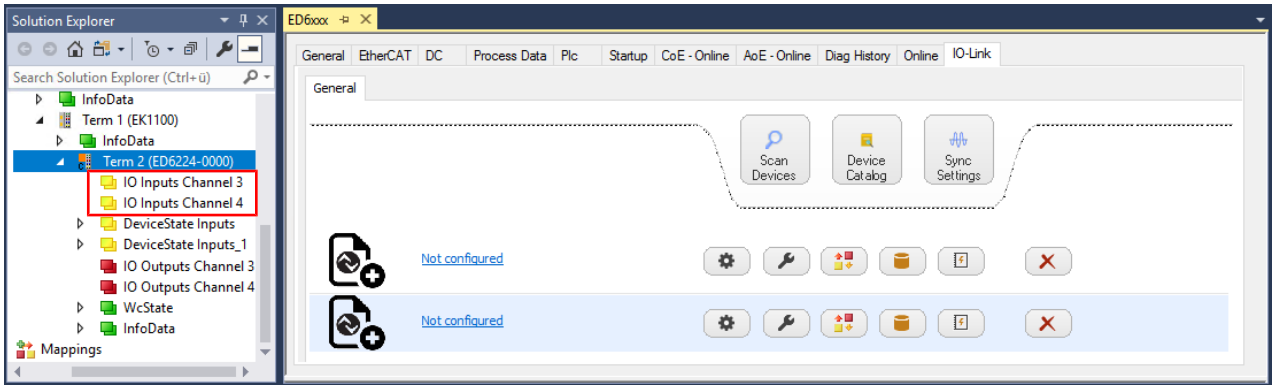


Fig. 33: The device on port 2 has been removed, the process data is no longer displayed in the tree.

4.2.2.4 Activating the configuration

Changes in the IO-Link configuration tool only become effective when you activate the IO-Link configuration.

There are two ways to activate the IO-Link configuration:

- Click on the "Reload Devices" button

- Activate the TwinCAT configuration:
  Click on the "Activate Configuration" button

4.2.3 Settings of the IO-Link devices

To find the basic settings of the devices for each port, proceed as follows.

1. Select the port and click on the “Settings” button to open the “Port n:: Settings” tab, in which the settings described below can be made.

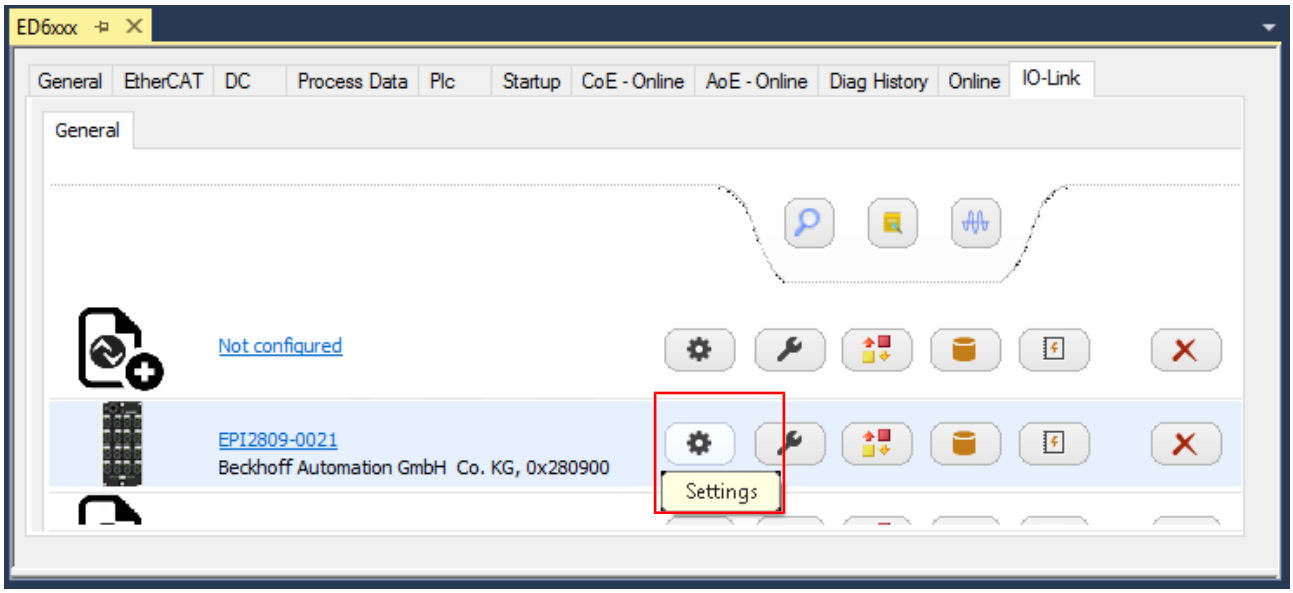


Fig. 34: Open device settings using the example of EPI2809-0021 on port 2

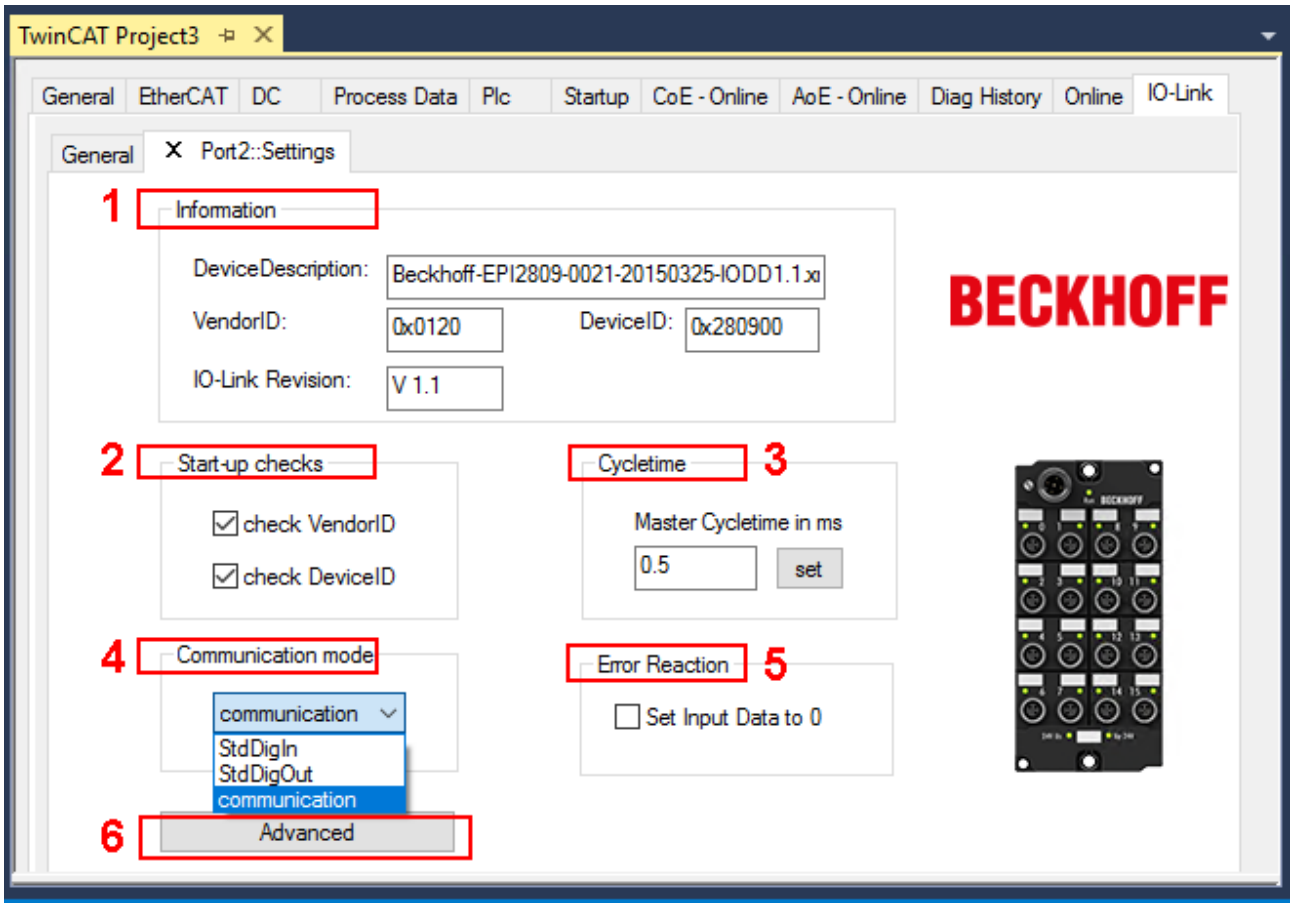
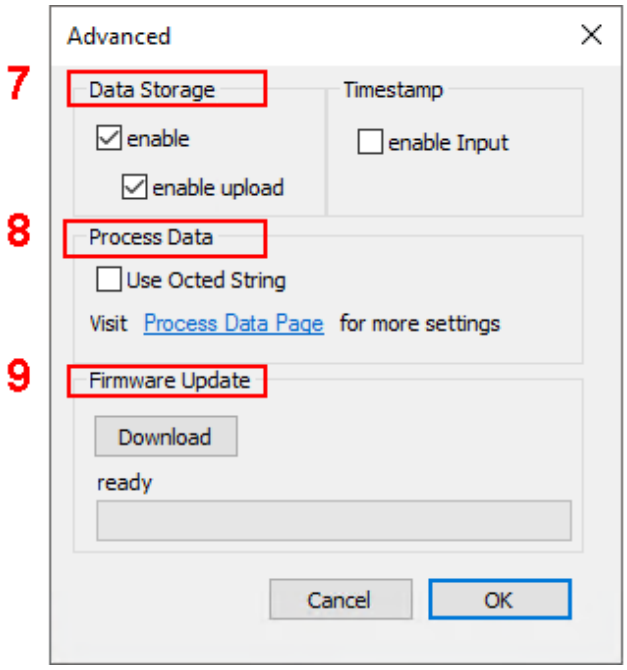


Fig. 35: Settings of the IO-Link devices

2. **Information**
   This field is for information only; the IODD that was read is displayed under “Device Description”. Furthermore, the “”Vendor ID”, “Device ID” and the “IO-Link Revision” (V 1.0 or V 1.1) of the IO-Link devices are displayed. If the device is an IO-Link V 1.1 device, then the parameter server [▶ 30] function is supported.

The following settings can be configured in the “Settings” tab (see figure above):

3. **Start-up checks**
   This parameter can be used to specify that the Vendor ID and Device ID should be checked when the IO-Link device starts up.
   ⇨ This avoids errors when replacing IO-Link devices.
4. **CycleTime**
   Specifies the cycle time of the IO-Link master.
5. **Communication mode**
   Selection of the mode in which the IO-Link port is to be operated.
   ⇨ “Communication”: Default mode for IO-Link devices
   ⇨ “StdDigIn / StdDigOut”: Mode for non-IO-Link devices, automatically selected if the port is configured as a digital input or output [▶ 38]
6. **Error Reaction**
   If the “Set Input Data to 0” field is activated
   ⇨ input data is set to 0 in case of error
   ⇨ status display: “Error”
7. **“Advanced” button**
8. **Data Storage**
   Please note the sensor versions:
   ⇨ V1.0 -> data storage is not supported
   ⇨ V1.1 -> data is stored in the parameter server (preset)
9. **Process Data Format**
   Adaptation of the process data format
   If the “Use Octet String” field is selected,
   ⇨ complex data types (process data) are created as octet strings.
   Advantage: simple further processing in the PLC
10. **Firmware update of Beckhoff IO-Link devices**
    A firmware update of the IO-Link device is possible via the “Download” button. Please refer to the description in the Firmware update of the IO-Link device chapter in the EPIxxxx documentation.



4.2.4 EPIxxxx, ERIxxxx - Setting of the IO-Link device parameters

This chapter explains how to read out and set the IO-Link device parameters.

The number and type of the objects shown on the “Parameters” tab vary according to the type of sensor. The default settings as stored in the IODD can initially be seen.

1. Click on the IO-Link master in the TwinCAT tree structure and click on the “IO-Link” tab.
2. Select the port to which the IO-Link device is connected and click on the “Properties” button to open the “Portn::Parameter“” tab.

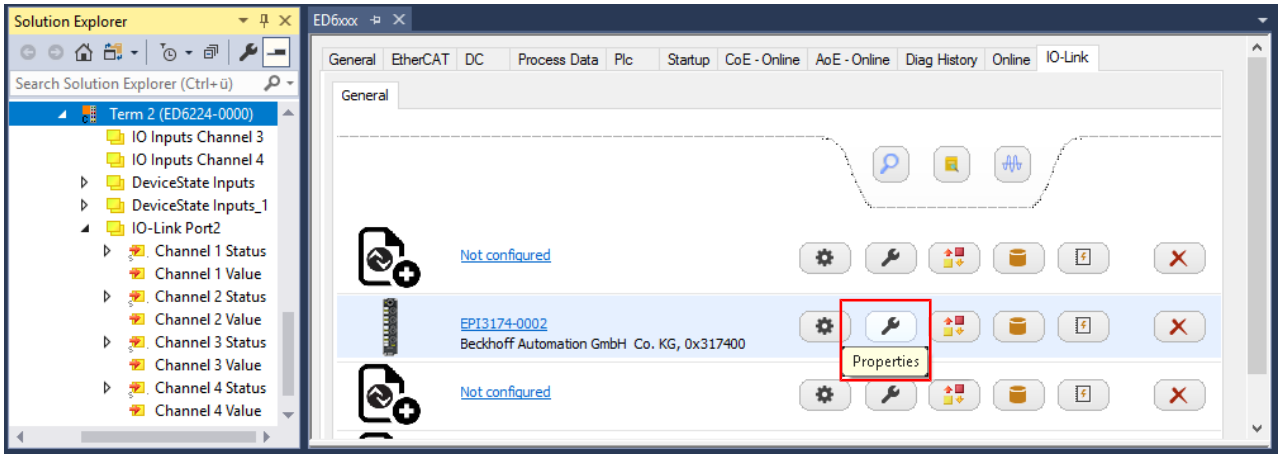


Fig. 36: Open device parameters using the example of EPI3174-0002 on port 2

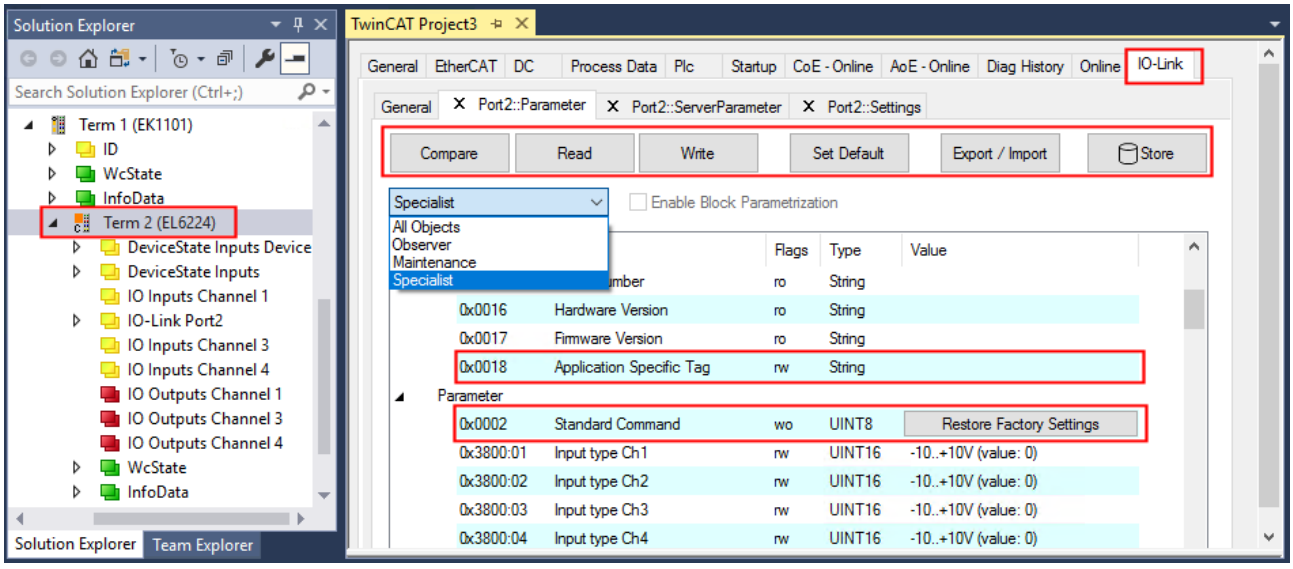


Fig. 37: Parameter des IO-Link Devices

The device parameters are listed in the tab. The buttons Compare [▶ 48], Read, Write [▶ 50], Set Default [▶ 51], Export/Import [▶ 52] and Store [▶ 53] are located at the top of the tab. The “Read”, “Write” and “Store” buttons are used to read out the parameters stored in the IO-Link device, load them and store them in the parameter server of the master.

Different user roles can be selected from the drop-down menu. The default user role is “Specialist”. The parameters are displayed in different representations and scopes.

Restarting the IO link device or restoring of the application parameters is possible via the parameter Standard Command [▶ 56].

Application specific information can be specified in parameter (0x0018) Application Specific Tag [▶ 57].

“Compare” button

1. Press the “Compare” button.
   ⇨ the parameter data of the configuration are compared with the parameter sets in the sensor.

⇨ The result is displayed in the “Parameter” tab see following figures.

Conformity of configuration and sensor data

The match is confirmed by a green tick in front of the index. Matching values are displayed in the “Value” field (see index 0x0018 “Application Specific Tag”).

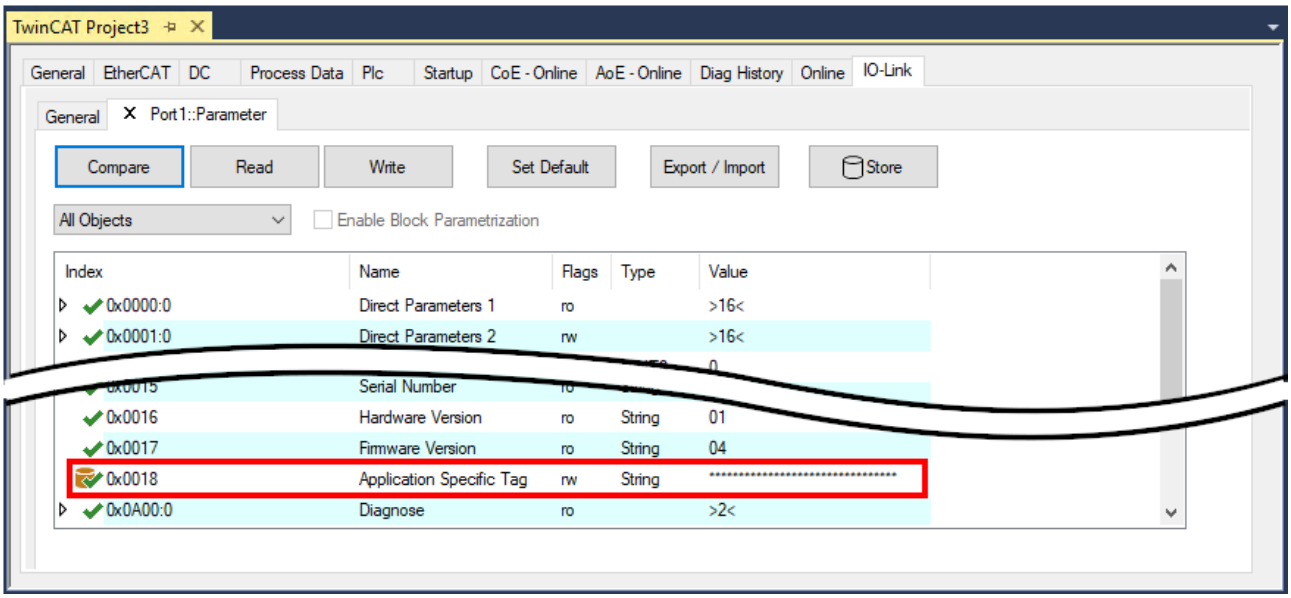


Fig. 38: Display of matching data in the “Parameter” tab

Deviations between configuration and sensor data

Deviations are indicated by a pen-symbol in front of the index. If there are different values in the “Value” field, the value “Compare” is displayed (see Index 0x0018 “Application Specific Tag”).

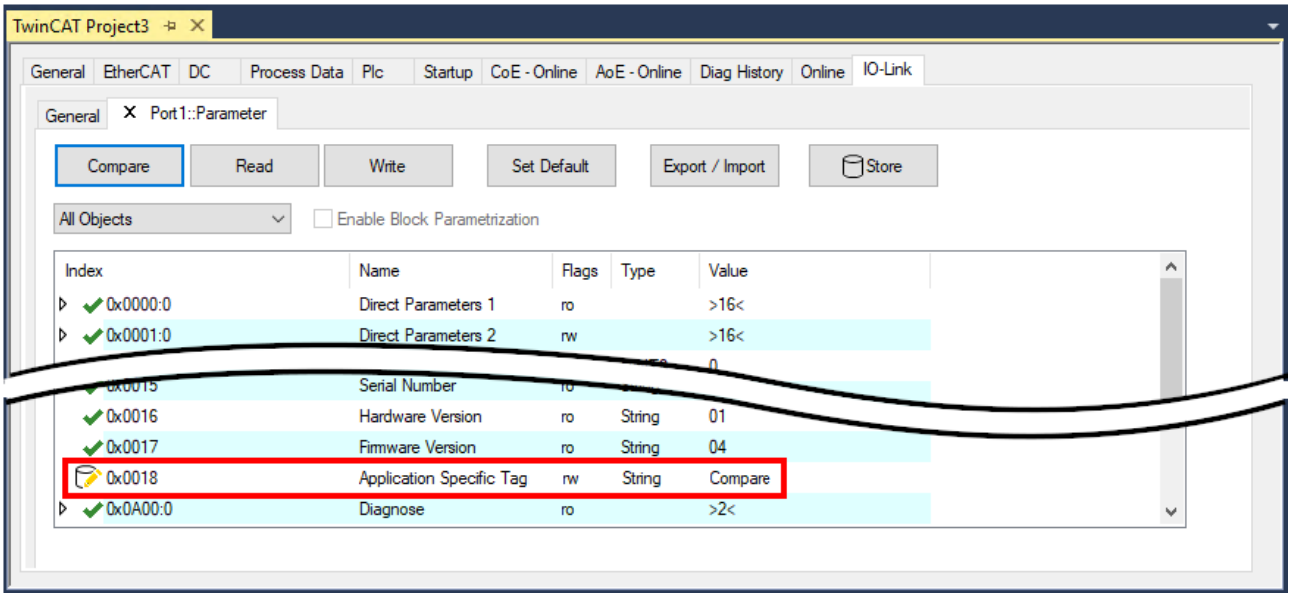


Fig. 39: Display of deviating data in the “Parameter” tab

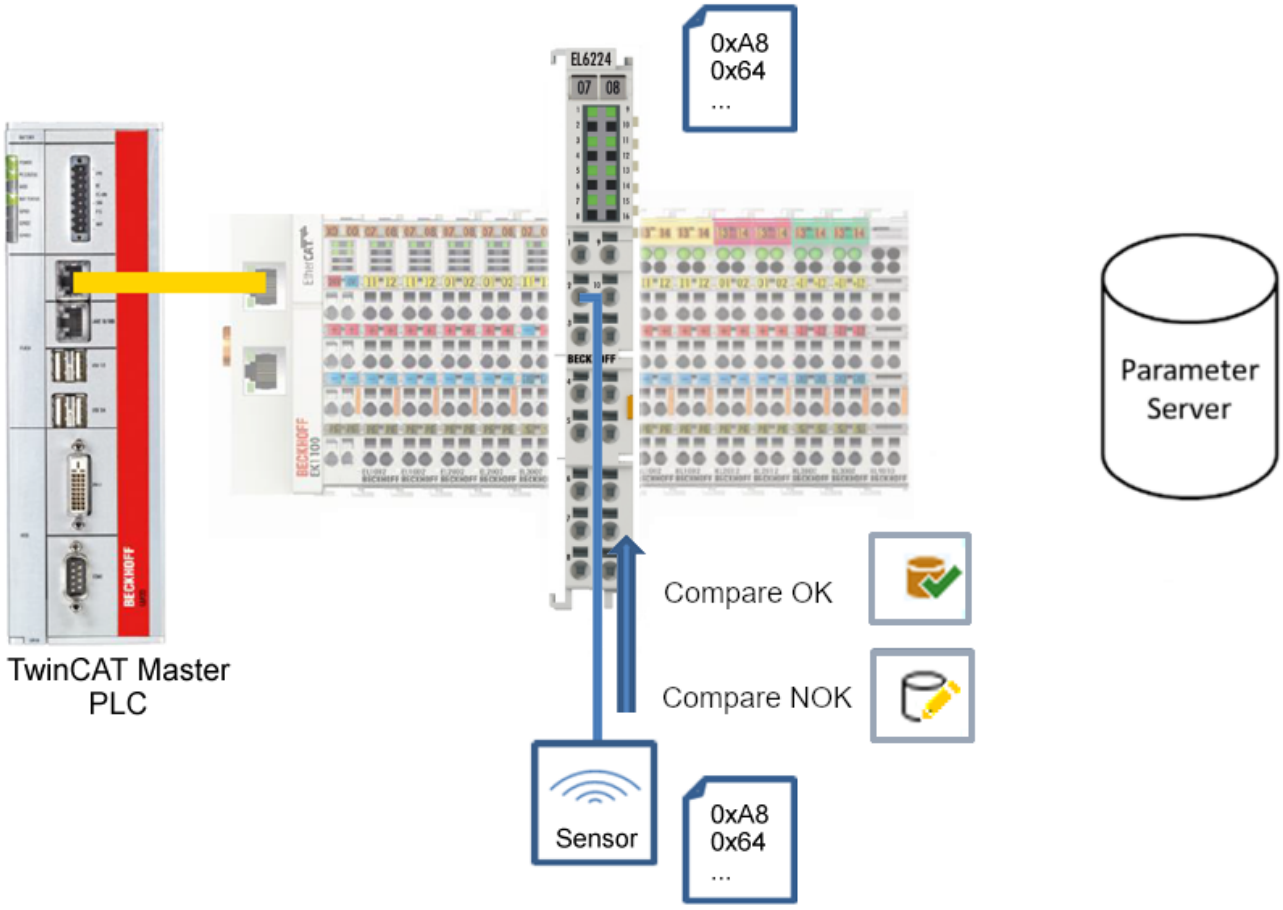


Fig. 40: Compare configuration and sensor data

“Read” button

The default values from the IODD file are always preset

1. Press the “Read” button

⇨ The current parameter values of the sensor are read. The successful reading of the data is confirmed with a green tick in front of the index.

“Write” button

The default values from the IODD file are always preset

1. Enter the desired value under “Value”.
2. Press the Enter key.
   ⇨ The values are accepted.
3. Press the “Write” button.

⇨ The data is written to the device (offline configuration is possible). The successful writing process is confirmed via a storing symbol in front of the index.

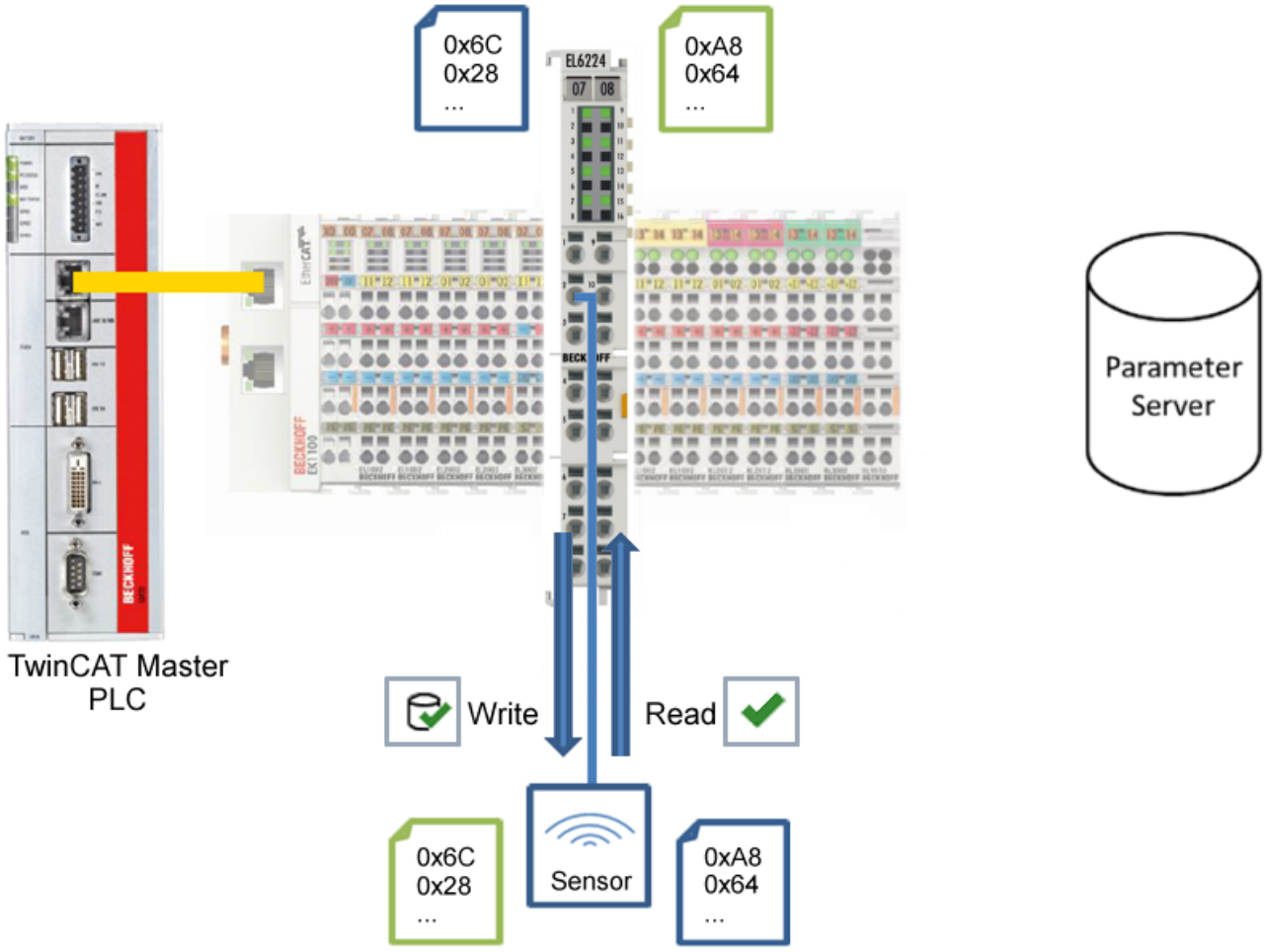


Fig. 41: Write parameter data to the sensor, read parameter data from the sensor.

“Set Default” button

1. Press the “Set Default” button.

⇨ All parameter values are set to the default settings.

Write default-values to the sensor

Note that the default-values must also be written to the device via the “Write” button.

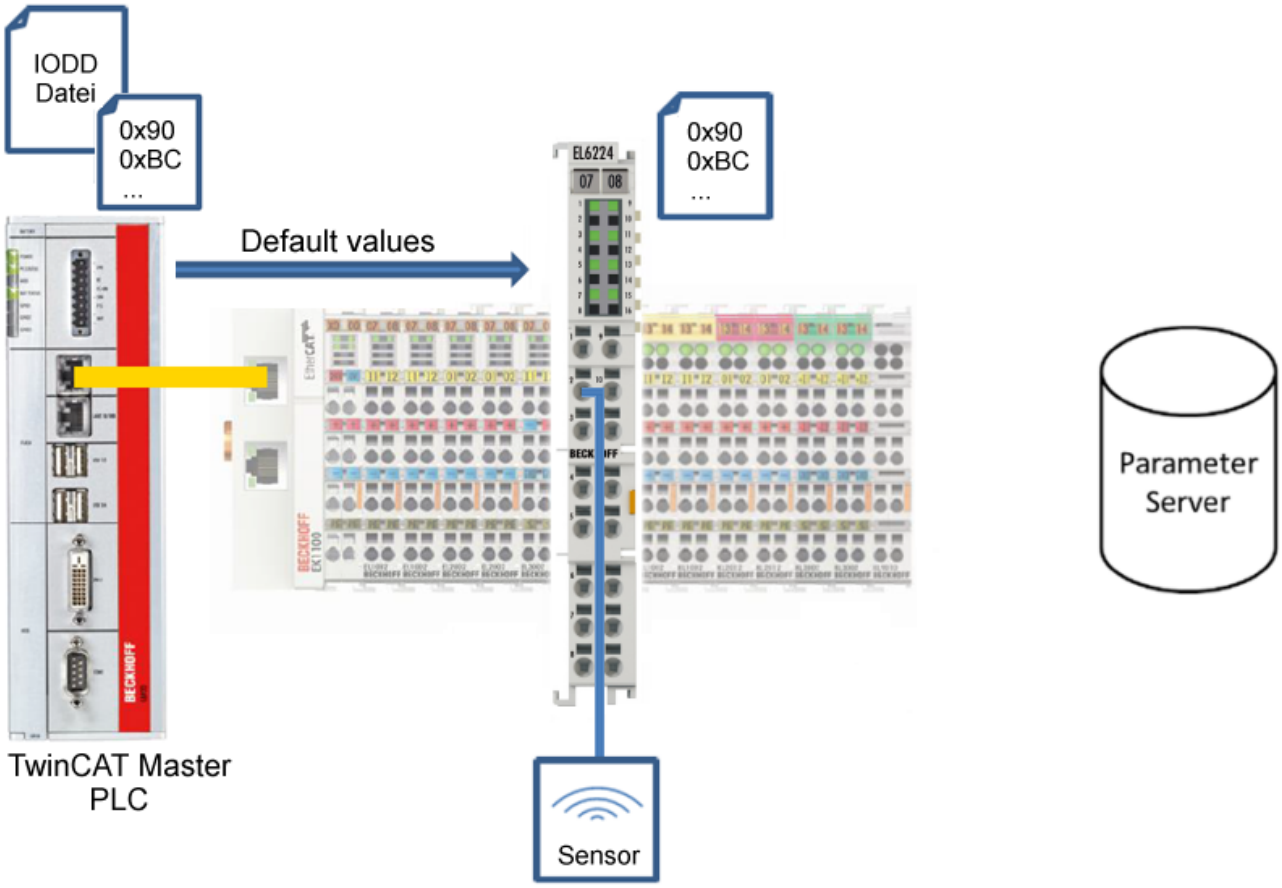


Fig. 42: Reset parameter values to default

“Export / Import” button

The set parameter values can be exported as a .vbs file and restored later via Import.

1. Press the “Export / Import” button (see the diagram below (1)).
   ⇨ The Import / Export dialog is opened
2. Specify the path under which you want to export or import the .vbs file, see fig. (2) below and confirm with the “Open” button, see fig. (4) below.
3. In addition, the export options “Attach Store Command” and “Enable Block Parametrization” can be selected as shown in fig. (3) below.
   ⇨ “Attach Store Command”: The parameters are loaded into the parameter server after the script has written all values.
   ⇨ “Enable Block Parametrization”: Block parameterization is enabled. For some sensors, writing is only possible when block parameterization is enabled.
4. Press the “Export” or “Import” button
   ⇨ The parameters are adopted from the imported file. The change of parameters is marked with a pencil symbol.
5. Write the new parameter values to the sensor via “Write” button.

⇨ The data is written to the device (offline configuration is possible). The successful writing process is confirmed via a storing symbol in front of the index.

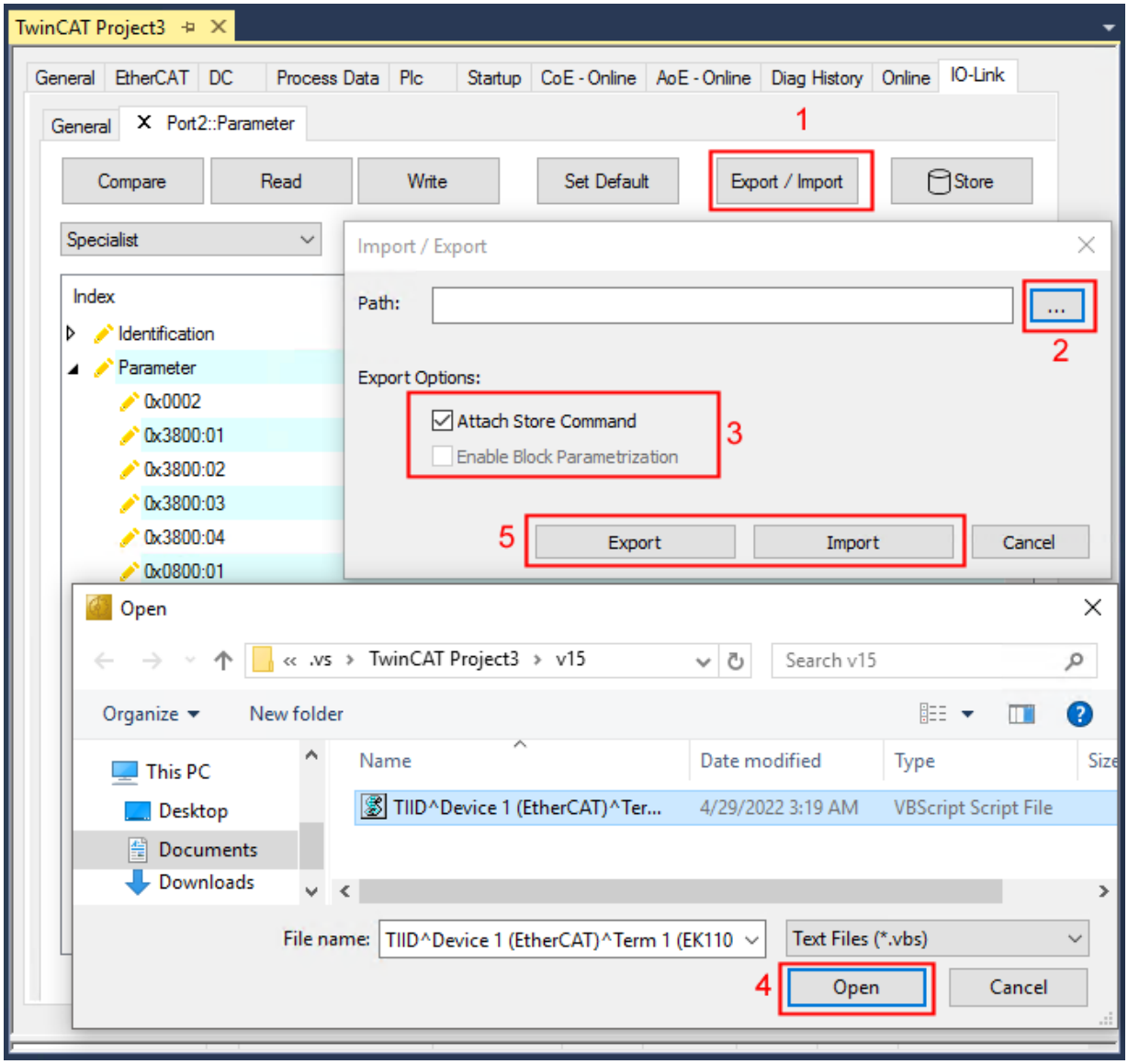


Fig. 43: Parameterization IO-Link device - Export / Import

“Store” button

1. Click “Store” (data storage):

   ⇨ The Beckhoff IO-Link master stores sensor-dependent-data, e. g. the following parameters
   (0x0018) “Application-Specific Tag”,
   (0x08n0) “Settings” and
   0x3800 “Range Settings”.
   The success of storing process is marked with the storing symbol.

   ⇨ If the IO-Link device is exchanged for a similar module, the device can be restored.

The stored values are displayed in the “ServerParameter” tab

1. Right-click on the device and select “Parameter Server” from the menu.

⇨ The stored values are displayed.

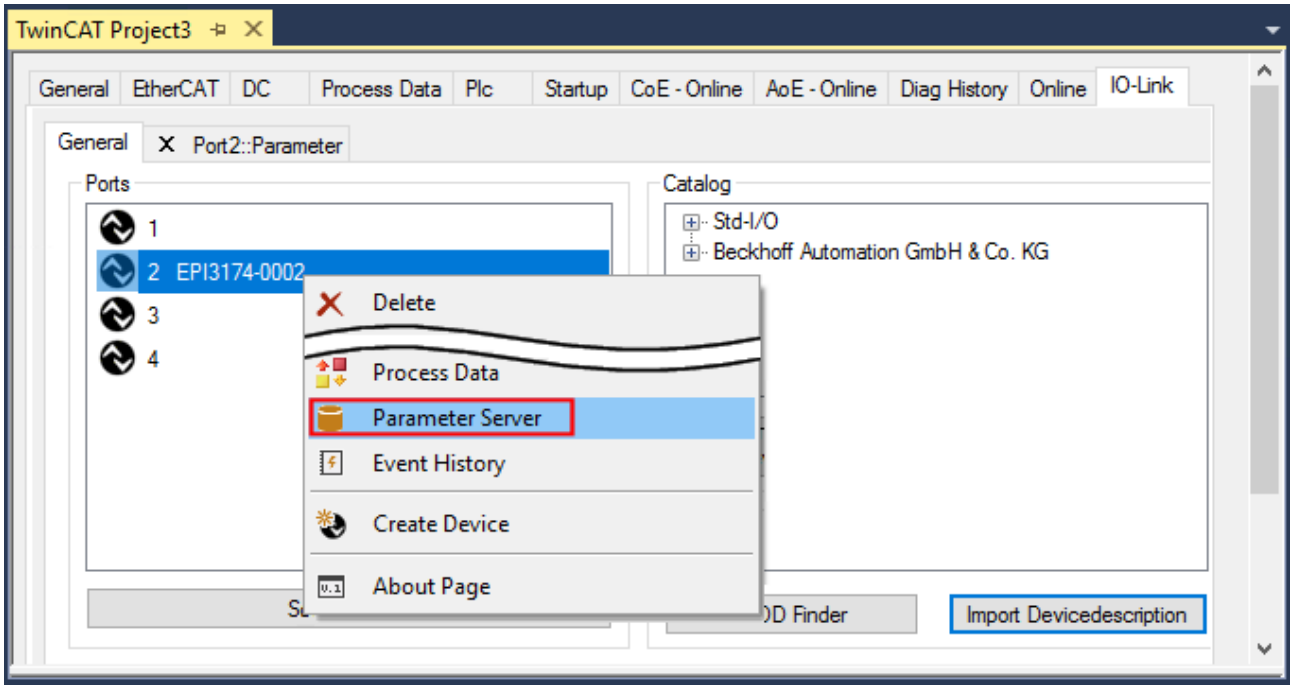


Fig. 44: Open the “ServerParameter” tab

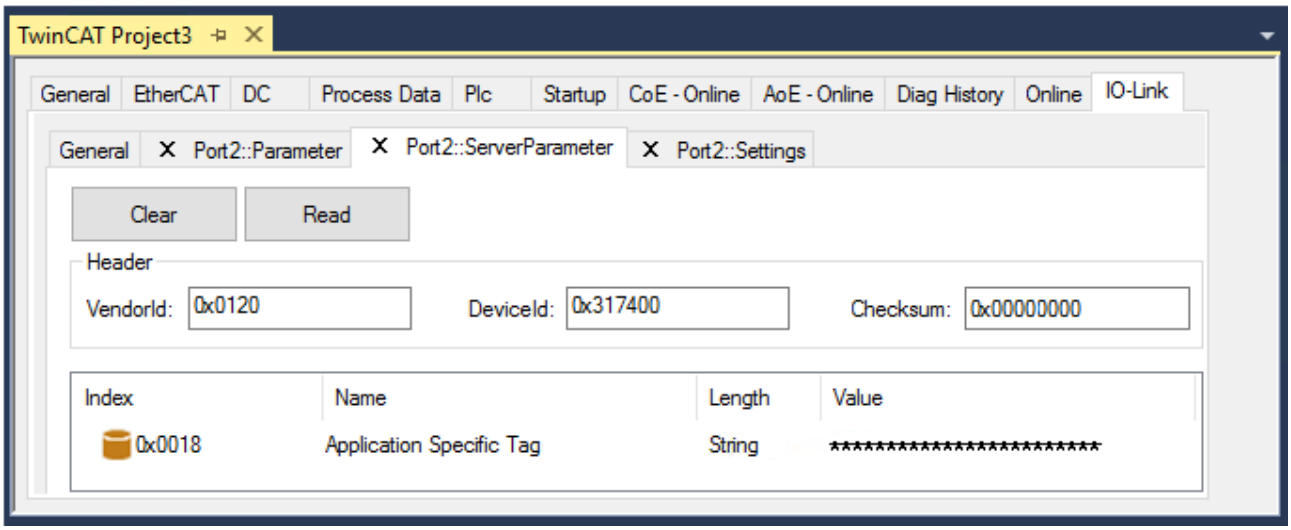


Fig. 45: „ServerParameter“ tab

Activate store button via PLC

As for CoE, the Indexgroup of an ADS command is specified as **0xF302** for the IO link data channel.

According to the IO-Link specification devices with ISDU support shall use index **0x0002** to receive the SystemCommand. The following list displays coding examples for system commands (ISDU), the complete table “Coding of SystemCommand (ISDU)” can be found in the IO-Link specification.

Command (hex)	Command (dec)	Name of the command	Definition
….			
0x01	1	ParamUploadStart	Start Parameter Upload
0x02	2	ParamUploadEnd	Stop Parameter Upload
0x03	3	ParamDownloadStart	Start Parameter Download
0x04	4	ParamDownloadEnd	Stop Parameter Download
0x05	**5**	**ParamDownloadStore**	**Finalize parameterization and start Data Storage**
0x06	6	ParamBreak	Cancel all Param commands
….			

Use an ADS Write function block for activating the store-function via the plc. The following figure shows a sample code for activation of the store-Button (command 0x05 “ParamDownloadStore”)

```
Case_Write:
    AdsWrite_EL6224( WRITE := FALSE );
    AdsWrite_EL6224.IDXGRP   := EL6224_Ch_iGrp;
    AdsWrite_EL6224.IDXOFFS  := EL6224_Ch_iOffWri;
    AdsWrite_EL6224.LEN      := SIZEOF(EL6224_bywrite);
    AdsWrite_EL6224.SRCADDR  := ADR(EL6224_bywrite);
    AdsWrite_EL6224( Write := TRUE);
    eSwitch1 := Case_WriBu;

EL6224_AoePortCh   : UINT  := 16#1001;
EL6224_Ch_iGrp     : UDINT := 16#F302;
EL6224_Ch_iOffManu : UDINT := 16#00100000;
EL6224_Ch_iOffPro  : UDINT := 16#00140000;
EL6224_Ch_iOffWri  : UDINT := 16#00020000;
EL6224_sManu       : STRING;
EL6224_sPro        : STRING;
EL6224_bywrite      : BYTE := 16#5;
```

Fig. 46: Sample code for activation of the store-function via the plc

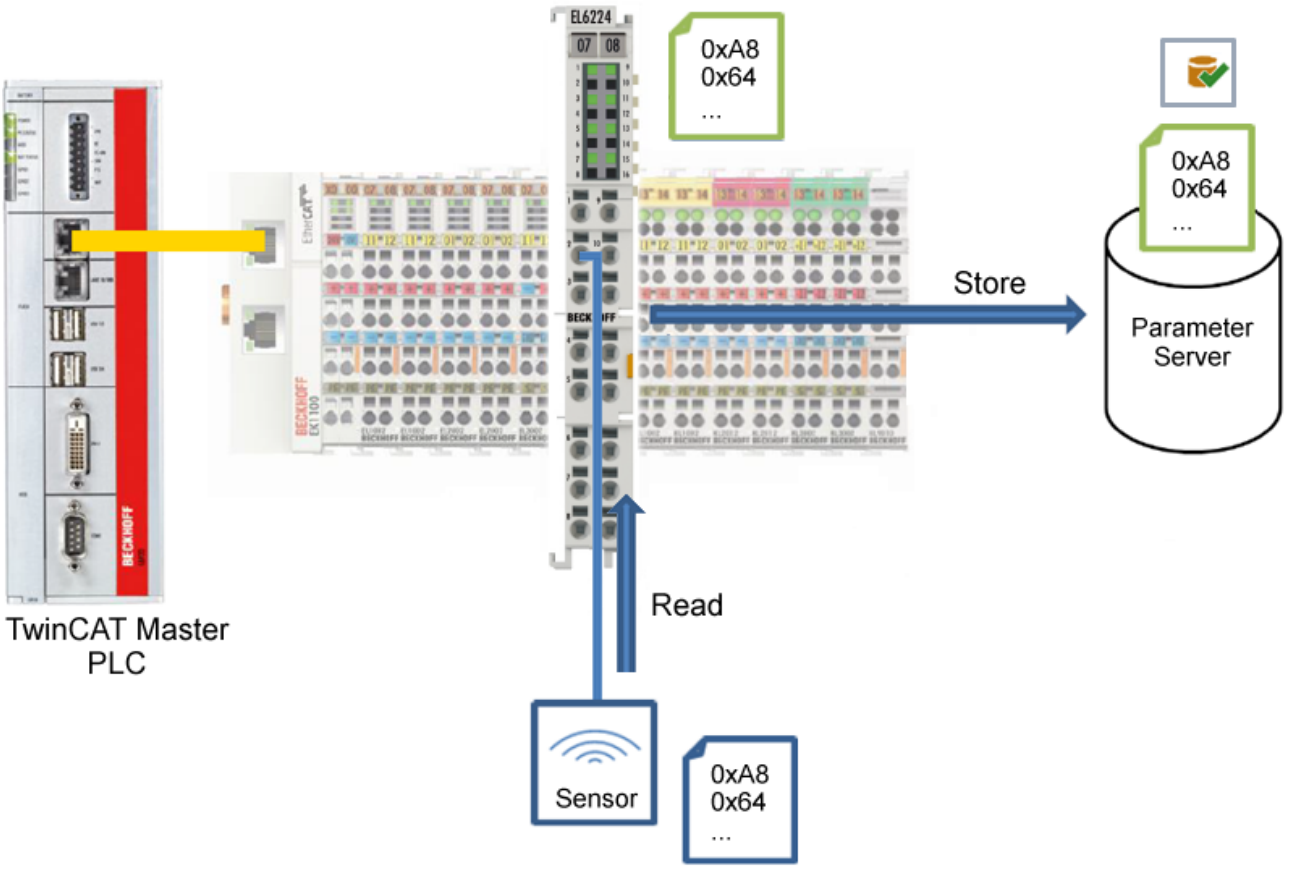


Fig. 47: Store parameters

Standard Command (Index 0x0002)

The IO-Link master writes various IO-Link-specific commands to the “Standard Command” during startup. Some of these commands are available in the TwinCAT interface (see figure below).

1. Click “Standard Command” in the parameter list of the “All Objects” user role, then double-click “Standard Command” in the right-hand field.
2. Select the desired value from the list of different options and
   - “Device Reset”: Restarts the IO-Link device.
   - “Application Reset”: No function.
   - “Restore Factory Settings”: Restoring the application parameters, i.e. the Settings parameter (0x0800).
3. Use the “Write” button (as described above [▶ 50]).

⇨ The data is written to the device (offline configuration is possible). The successful writing process is confirmed via a storing symbol in front of the index.

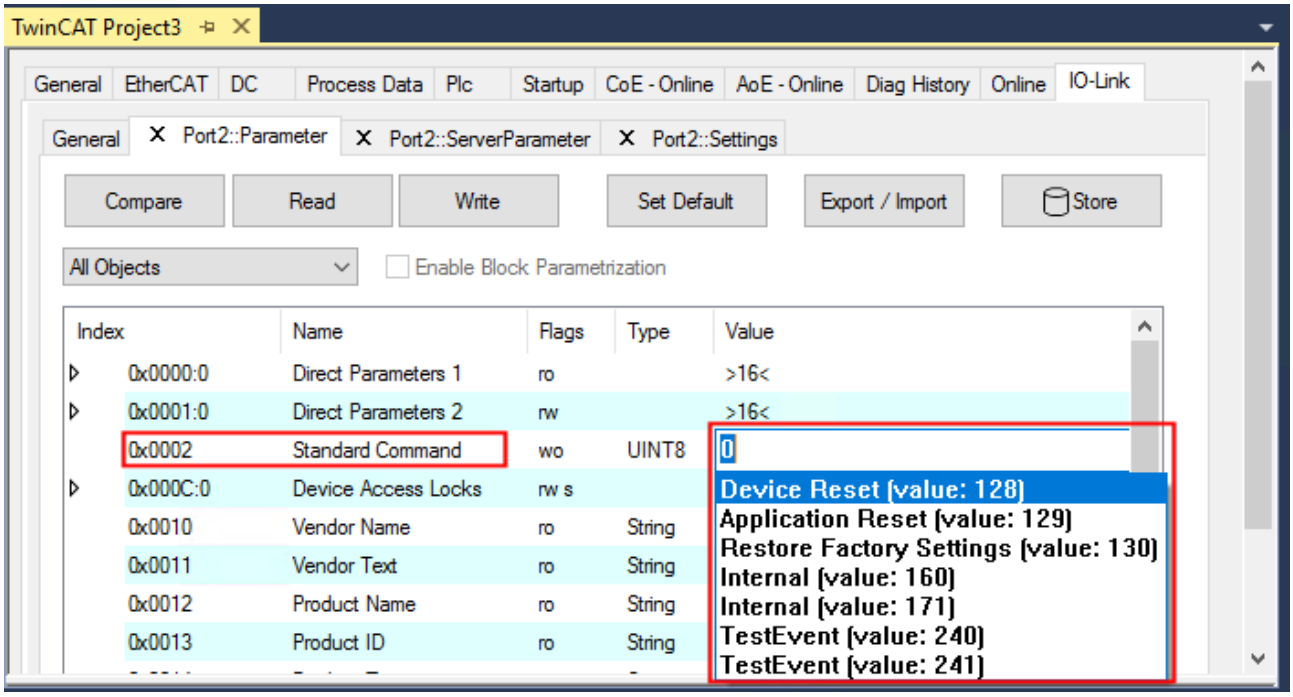


Fig. 48: Parameters IO-Link device “Standard Command”

“Application Specific Tag” (Index 0x0018)

Application-specific information can be entered and stored here.

1. Click “Application-Specific Tag” in the parameter list, then double-click “Application-Specific Tag” in the right-hand field.
2. Enter application-specific information and confirm with the Enter key.
3. Use the Write [▶ 50] button and the Store [▶ 53] button, if required (as described above).

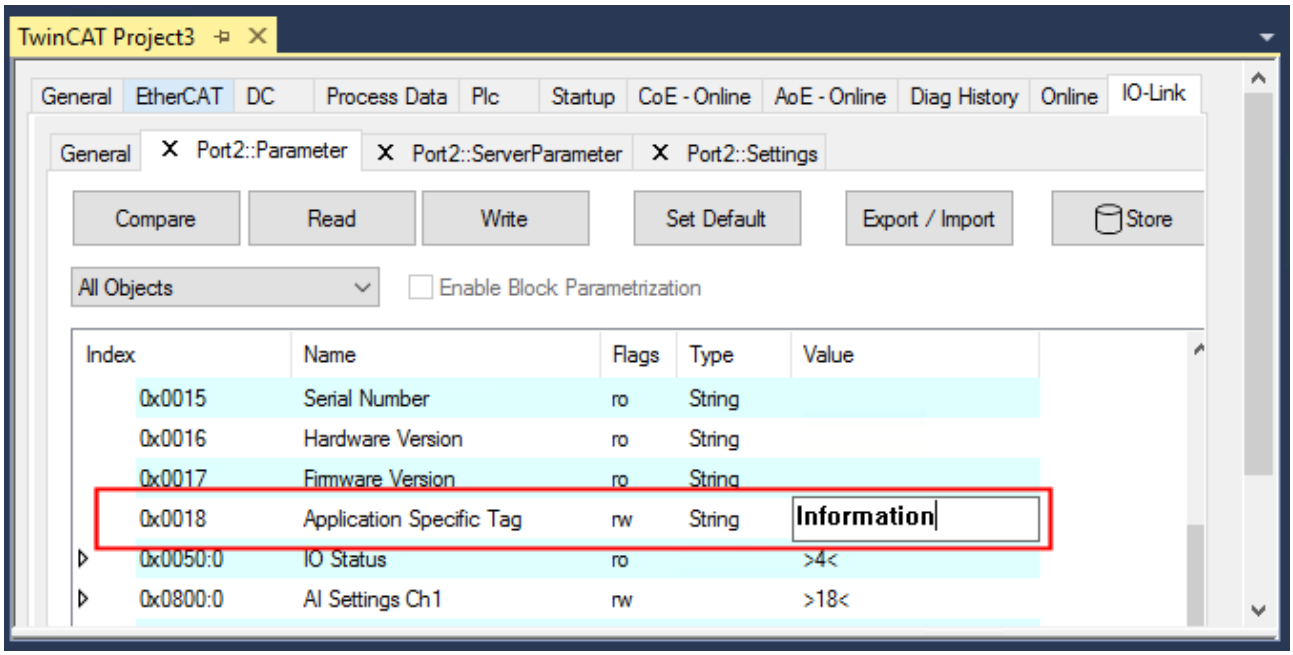


Fig. 49: Parameters IO-Link device: “Application Specific Tag”

4.2.5 Process data of the IO-Link device

The process data is displayed in the “Portn::ProcessDataPage” tab.

1. Click on the IO-Link master in the TwinCAT tree structure and click on the “IO-Link” tab.
2. Select the port to which the IO-Link device is connected and click on the “Process Data” button to open the “Portn::Parameter“” tab.

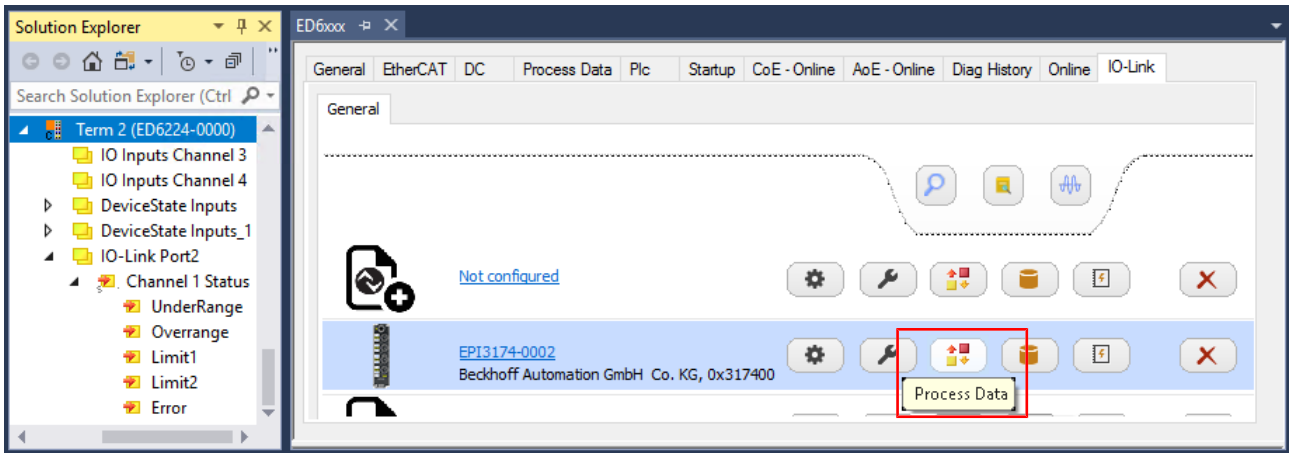


Fig. 50: Display process data for the IO-Link device on port 2

The “Portn::ProcessDataPage” tab shows the bit meaning, position, and length, as well as the value of the process data of the IO-Link device in the online display. The different colors can be used to quickly identify the location and length of the individual process data.

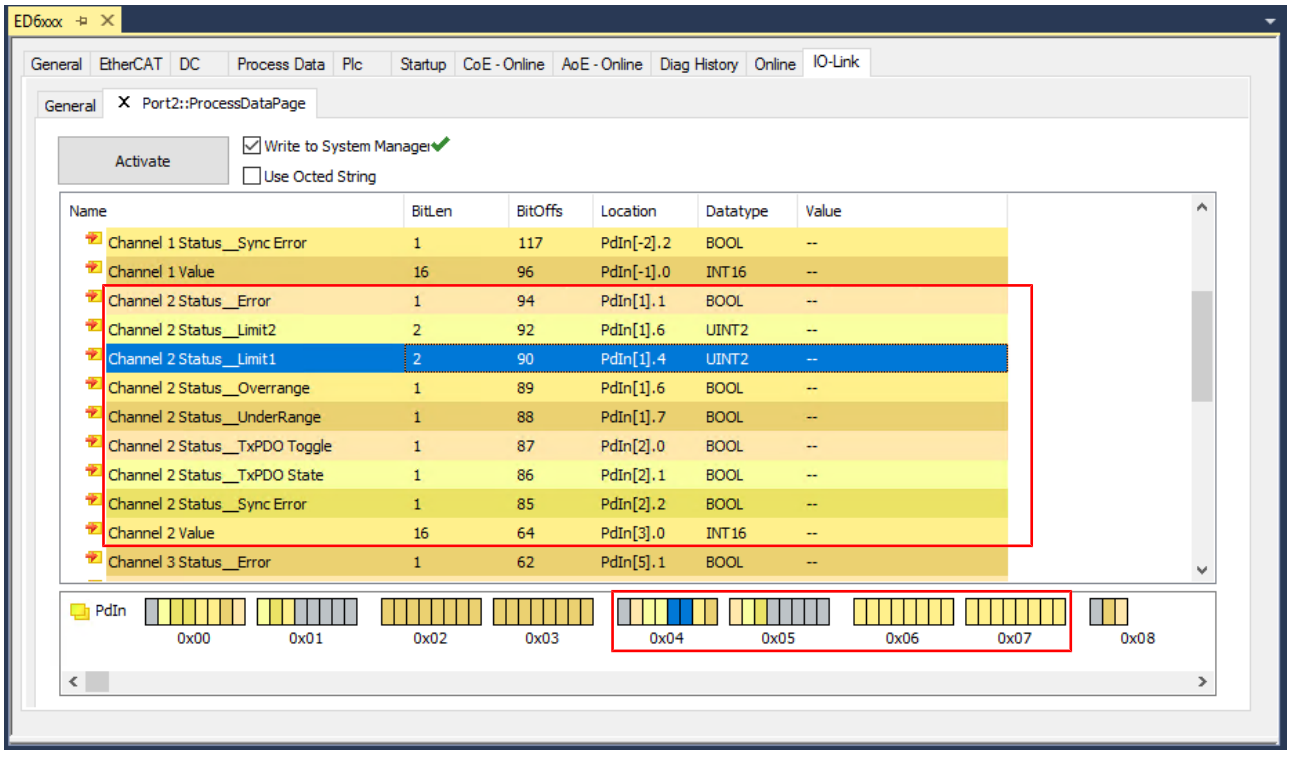


Fig. 51: Display of the process data of the IO-Link device on port 2

Hues				
Meaning	Unassigned bits	Marked process data	Input process data	Output process data

4.2.6 Event directory of the IO-Link device (event log)

The events of the IO-Link device are displayed in the “Portn::Event History” tab.

1. Click on the IO-Link master in the TwinCAT tree structure and click on the “IO-Link” tab.
2. Select the port to which the IO-Link device is connected and click on the “Event log” button to open the “Portn::Event History“” tab.

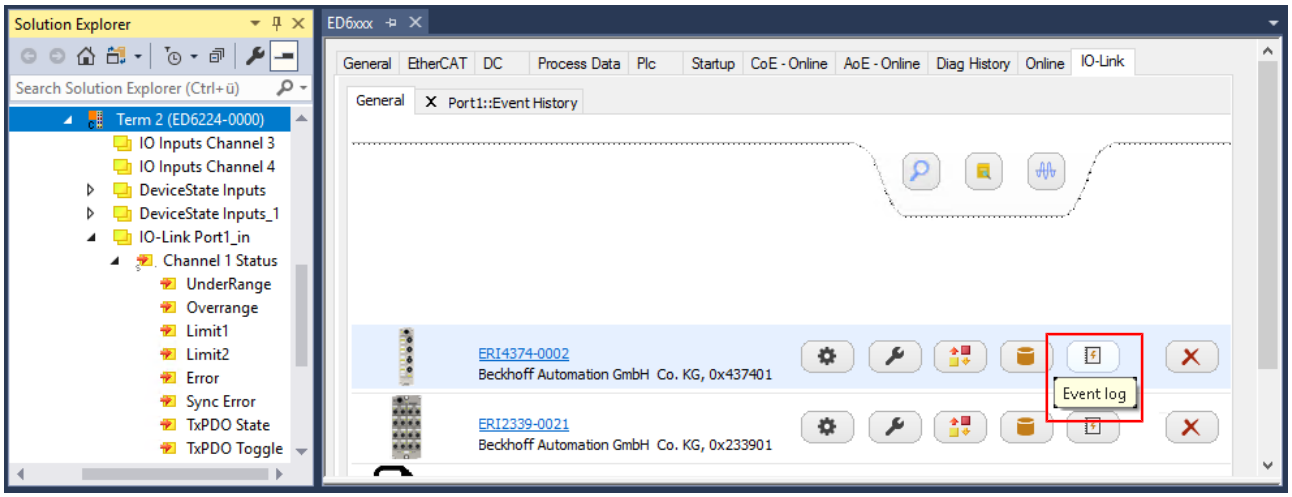


Fig. 52: Open the event directory for the IO-Link device on port 1

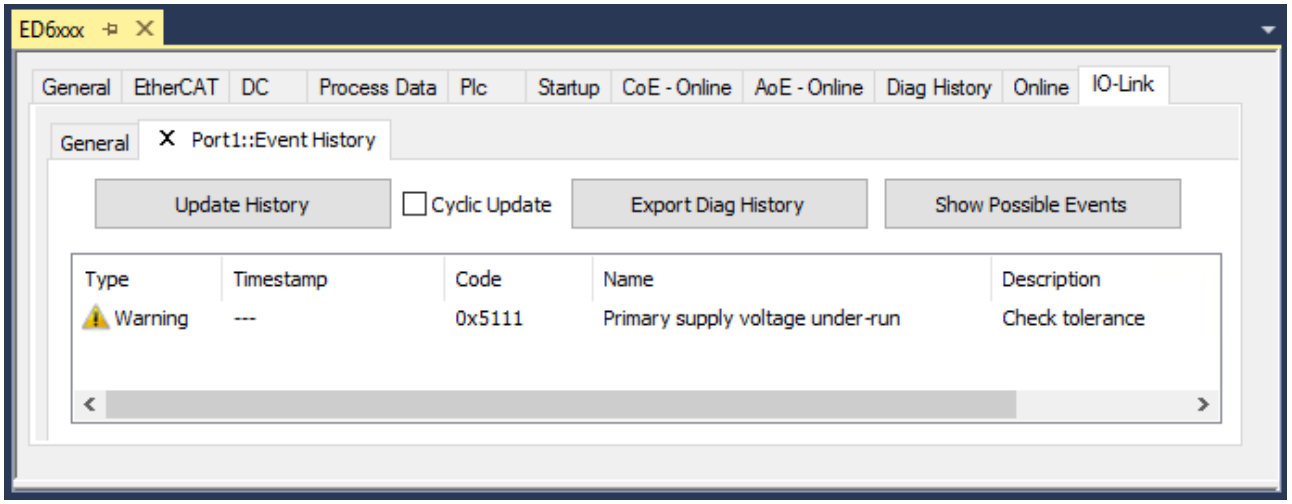


Fig. 53: Display of possible events “Show Possible Events” of the IO-Link device on port 1

- “Update History”:
  Update of the event history for the IO-Link device
- “Cyclic Update”:
  When the checkbox is activated, the event history of the IO-Link device is updated cyclically
- “Export Diag History”:
  Export of the event history
- “Show possible Events”:
  Shows the possible events of the IO-Link device

4.2.7 Synchronization “Sync Settings”

The Sync0 event is specified synchronously for the entire EtherCAT network via the distributed clocks function (for details, see EtherCAT system documentation chapter “I/O synchronization via distributed clocks”). The IO-Link start and cycle time can be specified depending on the Sync0 pulse. This means that the IO-Link data can be provided synchronously across the entire EtherCAT network.

The start time for the IO-Link telegram (offset) and the IO-Link cycle can be defined in the “IO-Link” tab.

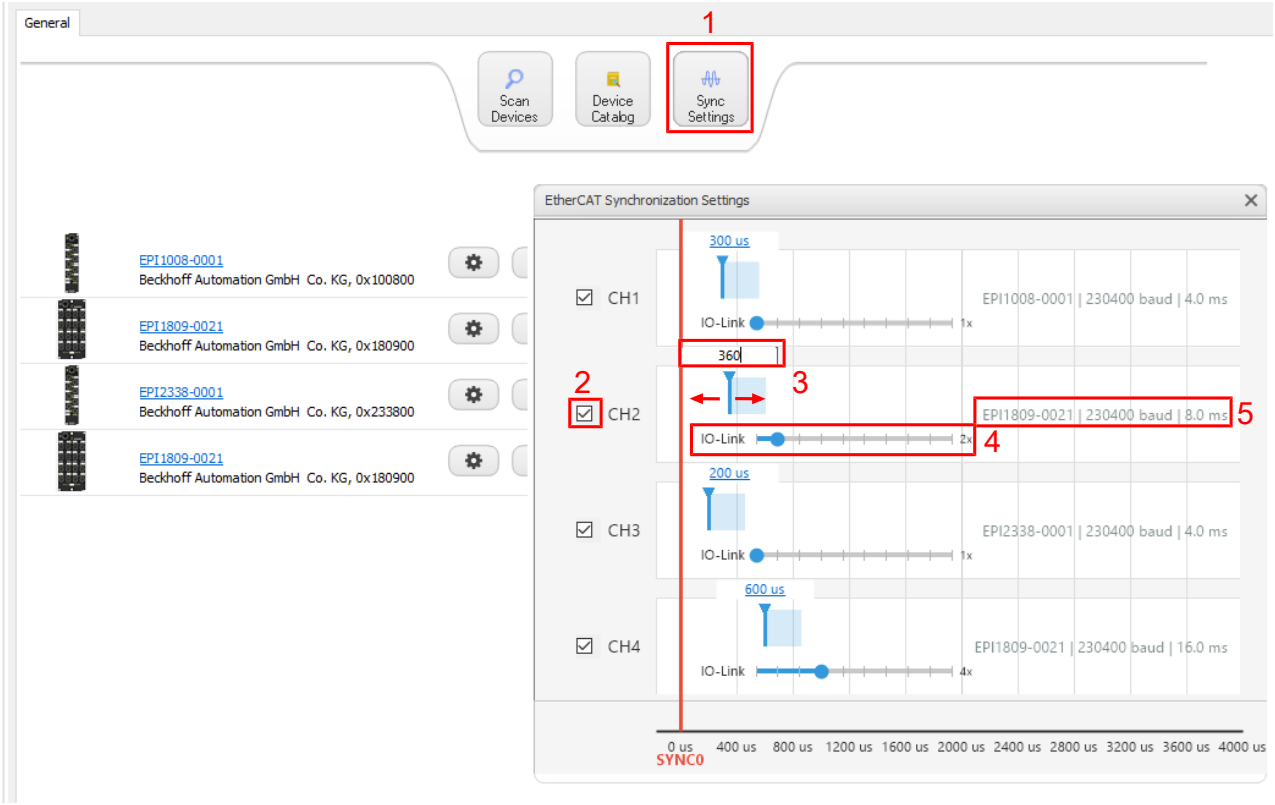


Fig. 54: Settings for the IO-Link start and cycle time via “EtherCAT Synchronization Settings”

1. Click the “Sync Settings” button. The “EtherCAT-Synchronization Settings” dialog will open.
2. Activate the desired channel via the checkbox.
3. Specify the start time (offset) for the IO-Link telegram in relation to the Sync0 event using the slider, or enter a value in the text box.
   ⇨ Equal offset times for all channels are suitable
   for actuator applications such as drives, solenoid valves…
   ⇨ Offset times for the individual channels are suitable
   , for example, for ultrasonic sensors to avoid mutual interference from superimposed signals.
4. Use the slider to define the IO-Link cycle depending on the Sync0 cycle (see figure above (4)). Note the EtherCAT cycle time and the cycle time of the sensor (see figure above, sensor data (5)).
   ⇨ 1x: New IO-Link data is provided after every Sync0 event
   2x: New IO-Link data is provided after every 2nd Sync event
   …

Note on setting the IO-Link cycle time

- The IO-Link cycle time must not be shorter than the EtherCAT cycle time.
  If an IO-Link cycle time < EtherCAT cycle time is selected, an error message is issued (see following figure).

⇨ Change the IO-Link cycle time!

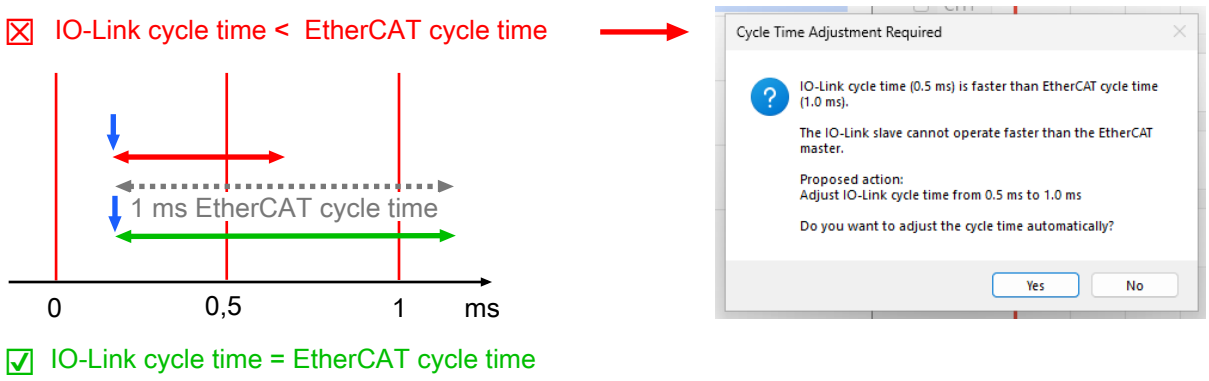


Fig. 55: Error message for IO-Link cycle time < EtherCAT cycle time

5. The IO-Link data is provided according to the selected settings depending on the Sync0 event.

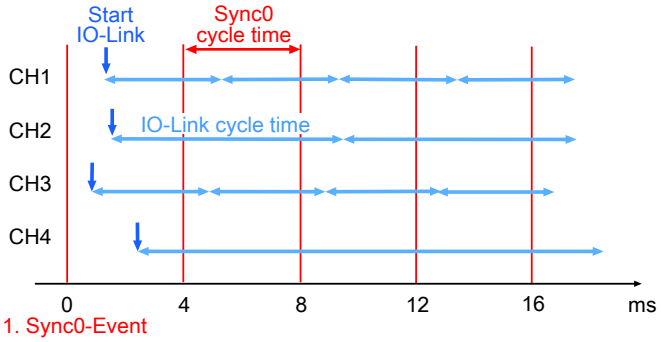


Fig. 56: Schematic representation based on the settings as in the previous graphic

Sync0 event settings

The settings for the Sync0 event can be adjusted via the “DC” and “Advanced Settings” tabs.

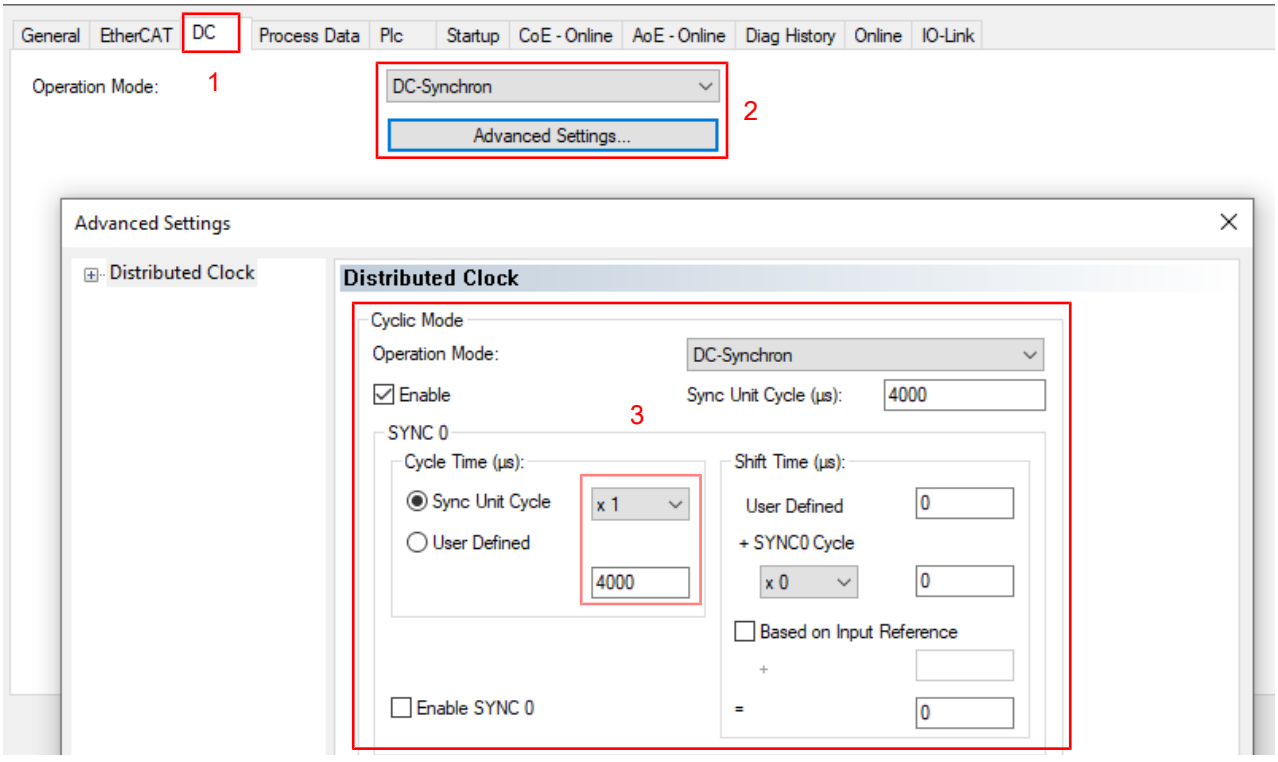


Fig. 57: Specification of Sync0 cycle via the “DC” tab under “Advanced Settings”

4.2.8 Object description

EtherCAT XML Device Description

The display matches that of the CoE objects from the EtherCAT XML Device Description. We recommend downloading the latest XML file from the download area of the Beckhoff website and installing it according to installation instructions.

NOTICE

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE - Online tab (with a double click on the respective object) or via the Process Data tab (assignment of PDOs). A detailed description can be found in the EtherCAT System-Documentation in chapter "EtherCAT subscriber configuration"

Please note the general CoE notes in the EtherCAT System Documentation in chapter "CoE-interface" when using/manipulating the CoE parameters:
- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary,
- existence of current XML description
- use "CoE reload" for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization during commissioning:
  - Restore Object [▶ 62] Index 0x1011
  - Configuration data [▶ 63] Index 0x80n0, 0x80n8
- Objects intended for regular operation, e.g. through ADS access.
- Profile-specific objects:
  - Input data [▶ 64] Index 0x60n0
  - Output data [▶ 64] Index 0x70n0
  - Information and diagnostic data [▶ 65] Index 0x90n0, 0xA0n0, 0xF000 - 0xF920
- Standard objects [▶ 68]

The following section first describes the objects required for normal operation, followed by a complete overview of missing objects.

4.2.8.1 Restore object

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default parameters	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to **"0x64616F6C"** in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0_{dec})

4.2.8.2 Configuration data (0x80n0)

Index 80n0 IO Settings Ch. 1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	IO Settings	Max. Subindex	UINT8	RO	0x28 (40_{dec})
80n0:04	Device ID	The device ID is used for validating the IO-Link device.	UINT32	RW	0x00000000 (0_{dec})
80n0:05	VendorID	The Vendor ID is used for validating the vendor of the IO-Link device.	UINT32	RW	0x00000000 (0_{dec})
80n0:20	IO-Link Revision	ID of the specification version based on which the IO-Link device communicates. Bit 0 - 3: MinorRev Bit 4 - 7: MajorRev	UINT8	RW	0x00 (0_{dec})
80n0:21	Frame capability	The Frame Capability indicates certain functionalities of the IO-Link device (e.g. ISDU supported). Bit 0: ISDU Bit 1: Type1 Bit 7: PHY1	UINT8	RW	0x00 (0_{dec})
80n0:22	Min cycle time	The cycle time refers to the communication between the IO-Link master and the IO-Link device. This value is transmitted in the IO-Link format for Min Cycle Time. Bit 6 and 7: time base Bit 0 to 5: multiplier (see following table)	UINT8	RW	0x00 (0_{dec})

Time base	Time base meaning	Calculation	Min. Cycle Time
00_{bin}	0.1 ms	multiplier x time base	0.0 - 6.3 ms
01_{bin}	0.4 ms	6.4 ms + multiplier x time base	6.4 - 31.6 ms
10_{bin}	1.6 ms	32.0 ms + multiplier x time base	32.0 - 132.8 ms
11_{bin}	6.4 ms	134.4 ms + multiplier x time base	134.4 - 537.6 ms

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:23	Offset time	Reserved	UINT8	RW	0x00 (0_{dec})
80n0:24	Process data in length	These parameters are transmitted in the IO-Link format for “Process data in length”. Bit 7: BYTE indicates whether the value in LENGTH is interpreted • as bit length [bit not set] or • as byte length + 1 [bit set] Bit 6: SIO indicates whether the device supports the standard I/O mode [bit set] Bit 0 to 4: LENGTH Length of the process data	UINT8	RW	0x00 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:25	Process data out length	These parameters are transmitted in the IO-Link format for “Process data out length”. Bit 7: BYTE indicates whether the value in LENGTH is interpreted • as bit length [bit not set] or • as byte length + 1 [bit set] Bit 6: SIO indicates whether the device supports the standard I/O mode [bit set] Bit 0 to 4: LENGTH Length of the process data	UINT8	RW	0x00 (0_{dec})
80n0:26	Compatible ID	reserved	UINT16	RW	0x0000 (0_{dec})
80n0:27	Reserved	reserved	UINT16	RW	0x0000 (0_{dec})
80n0:28	Master Control	Controls the IO-Link master port and defines the various operation modes of the IO-Link master. Bits 0...3 0: IO-Link port inactive 1: IO-Link port as digital input port 2: IO-Link port as digital output port 3: IO-Link port in communication via the IO-Link protocol 4: IO-Link port in communication via the IO-Link protocol. IO-Link state is ComStop (no cyclic communication, data is only exchanged when required). Bits 4...15 2: DataStorage active 4: DataStorage upload inactive	UINT16	RW	0x0000 (0_{dec})

Index 80n8 IO Sync Settings Ch. 1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n8:0	IO Sync Settings	Max. Subindex	UINT8	RO	0x02 (2_{dec})
80n8:01	Ratio		UINT16	RW	0x0000 (0_{dec})
80n8:05	Offset		UINT16	RW	0x0000 (0_{dec})

4.2.8.3 Input data (0x60n0)

Index 60n0 IO Inputs Ch. 1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	IO Inputs Ch.1 - 4	Max. Subindex	UINT8	RO	0x00 (0_{dec})
60n0:01	Subindex 001	IO-Link input process data	-	RO	-
60n0:10	Subindex 016	IO-Link input process data	-	RO	-

4.2.8.4 Output data (0x70x)

Index 70n0 IO Outputs Ch. 1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
70n0:0	IO Outputs Ch.1 - 4	Max. Subindex	UINT8	RO	0x00 (0_{dec})
70n0:01	Subindex 001	IO-Link output process data	-	RO	-
70n0:10	Subindex 016	IO-Link output process data	-	RO	-

4.2.8.5 Information and diagnostic data (0x900n0, 0xA0n0, 0xF000 - 0xF920)

Index 90n0 IO Info data Ch. 1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:0	IO Info data	Max. Subindex	UINT8	RO	0x27 (39_{dec})
90n0:04	Device ID	The device ID is used for validating the IO-Link device.	UINT32	RO	0x00000000 (0_{dec})
90n0:05	VendorID	The Vendor ID is used for validating the vendor of the IO-Link device.	UINT32	RO	0x00000000 (0_{dec})
90n0:20	IO-Link revision	ID of the specification version based on which the IO-Link device communicates. Bit 0 - 3: MinorRev Bit 4 - 7: MajorRev	UINT8	RO	0x00 (0_{dec})
90n0:21	Frame capability	The “Frame capability” indicates certain functionalities of the IO-Link device (e.g. ISDU supported). Bit 0: ISDU Bit 1: Type1 Bit 7: PHY1	UINT8	RO	0x00 (0_{dec})
90n0:22	Min cycle time	The cycle time corresponds to the interval at which the IO-Link master should communicate with the IO-Link device. This value is transmitted in the IO-Link format for Min Cycle Time. Bit 6 and 7: time base Bit 0 to 5: multiplier (see following Table)	UINT8	RO	0x00 (0_{dec})
90n0:23	Offset time	reserved	UINT8	RO	0x00 (0_{dec})
90n0:24	Process data in length	These parameters are transmitted in the IO-Link format for “Process data in length”. Bit 7: BYTE indicates whether the value in LENGTH is interpreted • as bit length [bit not set] or • as byte length + 1 [bit set] Bit 6: SIO indicates whether the device supports the standard I/O mode [bit set] Bit 0 to 4: LENGTH Length of the process data	UINT8	RO	0x00 (0_{dec})
90n0:25	Process data out length	These parameters are transmitted in the IO-Link format for “Process data out length”. Bit 7: BYTE indicates whether the value in LENGTH is interpreted • as bit length [bit not set] or • as byte length + 1 [bit set] Bit 6: SIO indicates whether the device supports the standard I/O mode [bit set] Bit 0 to 4: LENGTH Length of the process data	UINT8	RO	0x00 (0_{dec})
90n0:26	Reserved	reserved	UINT16	RO	0x0000 (0_{dec})
90n0:27	Reserved2	reserved	UINT16	RO	0x0000 (0_{dec})

Time base	Time base meaning	Calculation	Min. Cycle Time
00b	0.100 ms	multiplier x time base	0.000 - 6.300 ms
01b	0.400 ms	6.400 ms + multiplier x time base	6.400 - 31.600 ms
10b	1.600 ms	32.000 ms + multiplier x time base	32.000 - 132.800 ms
11b	6.400 ms	134.400 ms + multiplier x time base	134.400 - 537.600 ms

Index A0n0 IO Diag data Ch. 1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
A0n0:0	IO Diag data Ch.1 - 4	Max. Subindex	UINT8	RO	0x02 (2_{dec})
A0n0:01	IO-Link State	The value of the IO-Link state corresponds to a state from the IO-Link master state machine 0x00: MASTER_STATE_INACTIV 0x01: MASTER_STATE_DIGIN 0x02: MASTER_STATE_DIGOUT 0x03: MASTER_STATE_COMESTABLISH 0x04: MASTER_STATE_INITMASTER 0x05: MASTER_STATE_INITSLAVE 0x07: MASTER_STATE_PREOPERATE 0x08: MASTER_STATE_OPERATE 0x09: MASTER_STATE_STOP	UINT8	RO	0x00 (0_{dec})
A0n0:02	Lost Frames	This parameter counts the number of lost IO-Link telegrams. This value is deleted whenever IO-Link starts up, otherwise it is incremented continuously.	UINT8	RO	0x00 (0_{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the Modular Device Profile	UINT8	RO	0x02 (2_{dec})
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0004 (4_{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default
F008:0	Code word	reserved	UINT32	RW	0x00000000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Max. Subindex	UINT8	RW	0x04 (4_{dec})
F010:01	SubIndex 001	-	UINT32	RW	0x0000184C (6220_{dez})
F010:02	SubIndex 002	-	UINT32	RW	0x0000184C (6220_{dez})
F010:03	SubIndex 003	-	UINT32	RW	0x0000184C (6220_{dez})
F010:04	SubIndex 004	-	UINT32	RW	0x0000184C (6220_{dez})

Index F100 Diagnosis Status data

Index (hex)	Name	Meaning		Data type	Flags	Default
F100:0	Diagnosis Status data	Max. Subindex		UINT8	RO	0x04 (4_{dec})
F100:01	State Ch1	Status byte Ch. 1	See table “Meaning of status byte Ch. 1 - Ch. 4”	UINT8	RO	0x00 (0_{dec})
F100:02	State Ch2	Status byte Ch. 2		UINT8	RO	0x00 (0_{dec})
F100:03	State Ch3	Status byte Ch. 3		UINT8	RO	0x00 (0_{dec})
F100:04	State Ch4	Status byte Ch. 4		UINT8	RO	0x00 (0_{dec})

The status bytes are divided into two nibbles.

Meaning Status byte Ch. 1 - Ch. 4
Low nibble: 0x_0 = Port disabled 0x_1 = Port in std dig in 0x_2 = Port in std dig out 0x_3 = Port in communication OP 0x_4 = Port in communication COMSTOP / dig in Bit (only in std. IO Mode) 0x_5 = not defined 0x_6 = not defined 0x_7 = not defined 0x_8 = Process Data Invalid Bit Combinations are possible and are displayed as addition of the values (s. note)
Higher nibble: 0x1_ = Watchdog detected 0x2_ = internal Error 0x3_ = invalid Device ID 0x4_ = invalid Vendor ID 0x5_ = invalid IO-Link Version 0x6_ = invalid Frame Capability 0x7_ = invalid Cycle Time 0x8_ = invalid PD in length 0x9_ = invalid PD out length 0xA_ = no Device detected 0xB_ = error PreOP/Data storage Combinations are possible and are displayed as addition of the values (s. note)

Addition of the values in case of simultaneously occurring diagnostic messages

If messages occur simultaneously, the value is displayed as a sum in the Status byte of the relevant channel.

- Often for example 0x03 “Port in communication OP” and 0x08 “Process Data Invalid Bit” occur simultaneously:
  0x03 + 0x08 = 0x0B (11_{dec})

⇨ The value 0x0B (11_{dec}) is displayed in the Status byte.

Index F101 DeviceState Status data

Index (hex)	Name	Meaning	Data type	Flags	Default
F101:0	DeviceState status data	Max. Subindex	UINT8	RO	0x10 (16_{dec})
F101:0D	Device diag	TRUE: A new diagnostic message is in the DiagHistory	BOOL	RO	FALSE
F101:10	Device state	TRUE: Collective message if at least one device reports an error	BOOL	RO	FALSE

Index F820 ADS Server Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
F820:0	ADS Server Settings	Max. Subindex	UINT8	RW	0x02 (2_{dec})
F820:01	Net ID	NetId and port to which the DiagHistory messages can be sent via emergency	UINT16	RW	0x0000 (0_{dec})
F820:02	Port		UINT16	RW	0x0000 (0_{dec})

Index F900 Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
F900:0	Info data	Max. Subindex	UINT8	RO	0x09 (9_{dec})
F900:01	IO-Link Version	-	UINT8	RO	0x10 (16_{dec})

Index F915 LED Status

Index (hex)	Name	Meaning	Data type	Flags	Default
F915:0	LED Status	Max. Subindex	UINT8	RO	0x08 (8_{dec})
F915:01	Run		UINT32	RO	0x0000 0000 (0_{dec})
F915:02	LED2		UINT32	RO	0x0000 0000 (0_{dec})
F915:03	LED3		UINT32	RO	0x0000 0000 (0_{dec})
F915:04	LED4		UINT32	RO	0x0000 0000 (0_{dec})
F915:05	CH.1 status		UINT32	RO	0x0000 0000 (0_{dec})
F915:06	CH.2 status		UINT32	RO	0x0000 0000 (0_{dec})
F915:07	CH.3 status		UINT32	RO	0x0000 0000 (0_{dec})
F915:08	CH.4 status		UINT32	RO	0x0000 0000 (0_{dec})

Index F920 AoE Net Id

Index (hex)	Name	Meaning	Data type	Flags	Default
F920:0	AoE Net Id	Max. Subindex	UINT8	RO	0x09 (9_{dec})
F920:01	Net ID			RO	0x00 00 00 00 00 00 (0_{dec})

4.2.8.6 Standard objects

Standard objects (0x1000-0x1FFF)

The standard objects have the same meaning for all EtherCAT slaves.

4.2.8.6.1 1000 - 10F8 | Device

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: • The low word contains the CoE profile used. • The high word contains the module profile according to the modular device profile.	UINT32	RO	0x0184C1389 (407638921_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	ED6224

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	-

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	00

Index 100B Bootloader version

Index (hex)	Name	Meaning	Data type	Flags	Default
100B:0	Bootloader version	Bootloader version	STRING	RO	n/a

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x18501052 (407900242_{dec})
1018:03	Revision	Revision number of the EtherCAT slave, • The low word (bit 0-15) identifies the special terminal number. • The high word (bit 16-31) refers to the device description.	UINT32	RO	0x00000000 (0_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave, • Low word ◦ The low byte (bit 0-7) of the low word contains the year of production. ◦ The high byte (bit 8-15) of the low word contains the week of production. • The high word (bit 16-31) is 0.	UINT32	RO	0x00000000 (0_{dec})

Index 10E2 Manufacturer-specific Identification Code*)

Index (hex)	Name	Meaning	Data type	Flags	Default
10E2:0	Manufacturer-specific Identification Code	Manufacturer-specific Identification Code	UINT8	RO	0x01 (1_{dec})
10E2:01	SubIndex 001	reserved	STRING	RO	

Index 10F0 Backup parameter handling

Index (hex)	Name	Meaning	Data type	Flags	Default
10F0:0	Backup parameter handling	Information for standardized loading and saving of backup entries	UINT8	RO	0x01 (1_{dec})
10F0:01	Checksum	Checksum across all backup entries of the EtherCAT slave	UINT32	RO	0x00000000 (0_{dec})

Index 10F2 Backup parameter storage

Index (hex)	Name	Meaning	Data type	Flags	Default
10F2:0	Backup parameter storage	EtherCAT backup object	OCTET-STRING[4]	RW	{0}

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Maximum subindex	UINT8	RO	0x15 (21_{dec})
10F3:01	Maximum Messages	Maximum number of stored messages A maximum of 16 messages can be stored.	UINT8	RO	0x00 (0_{dec})
10F3:02	Newest Message	Subindex of the latest message	UINT8	RO	0x00 (0_{dec})
10F3:03	Newest Acknowledged Message	Subindex of the last confirmed message	UINT8	RW	0x00 (0_{dec})
10F3:04	New Messages Available	Indicates that a new message is available.	BOOLEAN	RO	0x00 (0_{dec})
10F3:05	Flags	not used	UINT16	RW	0x0000 (0_{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[24]	RO	{0}
…	…	….	…	…	…
10F3:15	Diagnosis Message 016	Message 16	OCTET-STRING[24]	RO	{0}

Index 10F8 Timestamp Object

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Timestamp Object	Timestamp Object [ns] For SM synchronous operation: Time since power-on For DC synchronous operation: copy of the DC time Time can be used by the device e.g. for timestamps of the DiagMessage	UINT64	RO	

4.2.8.6.2 160n | Mapping RxPDO

Index 1600 IO RxPDOPDO-Map Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	IO RxPDOPDO-Map Ch.1	PDO Mapping RxPDO 1	UINT8	RW	0x01 (1_{dec})
1600:01	SubIndex 001	1st PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8

Index 1601 IO RxPDOPDO-Map Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	IO RxPDOPDO-Map Ch.2	PDO Mapping RxPDO 2	UINT8	RW	0x01 (1_{dec})
1601:01	SubIndex 001	1st PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8

Index 1602 IO RxPDOPDO-Map Ch.3

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	IO RxPDOPDO-Map Ch.3	PDO Mapping RxPDO 3	UINT8	RW	0x01 (1_{dec})
1602:01	SubIndex 001	1st PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8

Index 1603 IO RxPDOPDO-Map Ch.4

Index (hex)	Name	Meaning	Data type	Flags	Default
1603:0	IO RxPDOPDO-Map Ch.4	PDO Mapping RxPDO 4	UINT8	RW	0x01 (1_{dec})
1603:01	SubIndex 001	1st PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8

4.2.8.6.3 1A0n, 1A80, 1A81 | Mapping TxPDO

Index 1A00 IO TxPDOPDO-Map Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	IO TxPDOPDO-Map Ch.1	PDO Mapping TxPDO 1	UINT8	RW	0x01 (1_{dec})
1A00:01	SubIndex 001	1st PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8

Index 1A01 IO TxPDOPDO-Map Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	IO TxPDOPDO-Map Ch.2	PDO Mapping TxPDO 2	UINT8	RW	0x01 (1_{dec})
1A01:01	SubIndex 001	1st PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8

Index 1A02 IO TxPDOPDO-Map Ch.3

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	IO TxPDOPDO-Map Ch.3	PDO Mapping TxPDO 3	UINT8	RW	0x01 (1_{dec})
1A02:01	SubIndex 001	1st PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8

Index 1A03 IO TxPDOPDO-Map Ch.4

Index (hex)	Name	Meaning	Data type	Flags	Default
1A03:0	IO TxPDOPDO-Map Ch.4	PDO Mapping TxPDO 4	UINT8	RW	0x01 (1_{dec})
1A03:01	SubIndex 001	1st PDO Mapping entry (8 bits align)	UINT32	RW	0x0000:00, 8

Index 1A80 DeviceState TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A80:0	DeviceState TxPDO-Map Inputs	PDO Mapping TxPDO 5	UINT8	RW	0x04 (4_{dec})
1A80:01	SubIndex 001	1st PDO Mapping entry (object 0xF100 (Diagnosis Status data), entry 0x01 (State Ch1))	UINT32	RW	0xF100:01, 8
1A80:02	SubIndex 002	2nd PDO Mapping entry (object 0xF100 (Diagnosis Status data), entry 0x02 (State Ch2))	UINT32	RW	0xF100:02, 8
1A80:03	SubIndex 003	3rd PDO Mapping entry (object 0xF100 (Diagnosis Status data), entry 0x03 (State Ch3))	UINT32	RW	0xF100:03, 8
1A80:04	SubIndex 004	4th PDO Mapping entry (object 0xF100 (Diagnosis Status data), entry 0x04 (State Ch4))	UINT32	RW	0xF100:04, 8

Index 1A81 DeviceState TxPDO-Map Inputs Device

Index (hex)	Name	Meaning	Data type	Flags	Default
1A81:0	DeviceState TxPDO-Map Inputs Device	PDO Mapping TxPDO 6	UINT8	RW	0x04 (4_{dec})
1A81:01	SubIndex 001	1st PDO Mapping entry (12 bits align)	UINT32	RW	0x0000:00, 12
1A81:02	SubIndex 002	2nd PDO Mapping entry (object 0xF101 (DeviceState Status data), entry 0x0D (Device Diag))	UINT32	RW	0xF101:0D, 1
1A81:03	SubIndex 003	3rd PDO Mapping entry (2 bits align)	UINT32	RW	0x0000:00, 2
1A81:04	SubIndex 004	5th PDO Mapping entry (object 0xF101 (DeviceState Status data), entry 0x0D (Device State))	UINT32	RW	0xF101:10, 1

4.2.8.6.4 1C00 - 1C33 | Device - System Manager

Index 1C00 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default
1C00:0	Sync manager type	Using the Sync Managers	UINT8	RO	0x04 (4_{dec})
1C00:01	SubIndex 001	Sync Manager Type Channel 1: Mailbox Write	UINT8	RO	0x01 (1_{dec})
1C00:02	SubIndex 002	Sync Manager Type Channel 2: Mailbox Read	UINT8	RO	0x02 (2_{dec})
1C00:03	SubIndex 003	Sync Manager Type Channel 3: Process Data Write (Outputs)	UINT8	RO	0x03 (3_{dec})
1C00:04	SubIndex 004	Sync Manager Type Channel 4: Process Data Read (Inputs)	UINT8	RO	0x04 (4_{dec})

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x04 (4_{dec})
1C12:01	Subindex 001	1st assigned RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:02	Subindex 002	2nd assigned RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1601 (5633_{dec})
1C12:03	Subindex 003	3rd assigned RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1602 (5634_{dec})
1C12:04	Subindex 004	4th assigned RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1603 (5635_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x06 (6_{dec})
1C13:01	Subindex 001	1st assigned TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	Subindex 002	2nd assigned TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A01 (6657_{dec})
1C13:03	Subindex 003	3rd assigned TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A02 (6658_{dec})
1C13:04	Subindex 004	4th assigned TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A03 (6659_{dec})
1C13:05	Subindex 005	5th assigned TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A80 (6784_{dec})
1C13:06	Subindex 006	6th assigned TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A81 (6785_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchronous with SM 2 event • 2: DC mode - synchronous with SYNC0 event • 3: DC mode - synchronous with SYNC1 event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • Synchronous with SM 2 event: cycle time of the master • DC Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0007A120 (500000_{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: Free Run is supported • Bit 1 = 1: Synchronization with SM 2 event is supported • Bit 2-3 = 01: DC-Mode is supported • Bit4 -5 = 10: Output Shift with SYNC1 event (DC mode only) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08)	UINT16	RO	0x0805 (2053_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0007A120 (500000_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time	Minimum time between SYNC1 event and output of the outputs (in ns)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Get cycle time	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03, 0x1C32:05, 0x1C32:06, 0x1C32:09, 0x1C33:03, 0x1C33:06, 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RO	0x0000 (0_{dec})
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of times that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC Mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32_{dec})
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchronous with SM 3 Event (no outputs available) • 2: DC - synchronous with SYNC0 event • 3: DC - synchronous with SYNC1 event • 34: Synchronous with SM 2 event (outputs available)	UINT16	RW	0x0000 (0_{dec})
1C33:02	Cycle time	Cycle time (in ns): • Free Run: cycle time of the local timer • Synchronous with SM 2 event: cycle time of the master • DC Mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0007A120 (500000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, DC Mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: Free Run is supported • Bit 1: Synchronization with SM 2 event is supported (outputs available) • Bit 1: Synchronization with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC-Mode is supported • Bit 4-5 = 01: Input shift through local event (outputs available) • Bit 4-5 = 10: Input shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 or 0x1C33:08)	UINT16	RO	0x0005 (5_{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0007A120 (500000_{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and the inputs being available for the master (in ns, DC mode only)	UINT32	RO	0x0007A120 (500000_{dec})
1C33:07	Minimum delay time	Min. time between SYNC1 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:08	Get cycle Time	This entry allows to measure the real required process data provision time. • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C33:03, 0x1C33:06, 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RO	0x0000 (0_{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C33:0D	Shift too short counter	Number of times that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC Mode only)	BOOLEAN	RO	0x00 (0_{dec})

4.3 Access to IO-Link data

4.3.1 IO-Link system communication

The Beckhoff IO-Link masters are divided into two services.

- On the one hand, they act as an IO-Link master for the connected IO-Link devices;
- on the other hand, they act as an EtherCAT slave in relation to the PLC TwinCAT master.

The system communication is shown in the following illustration.

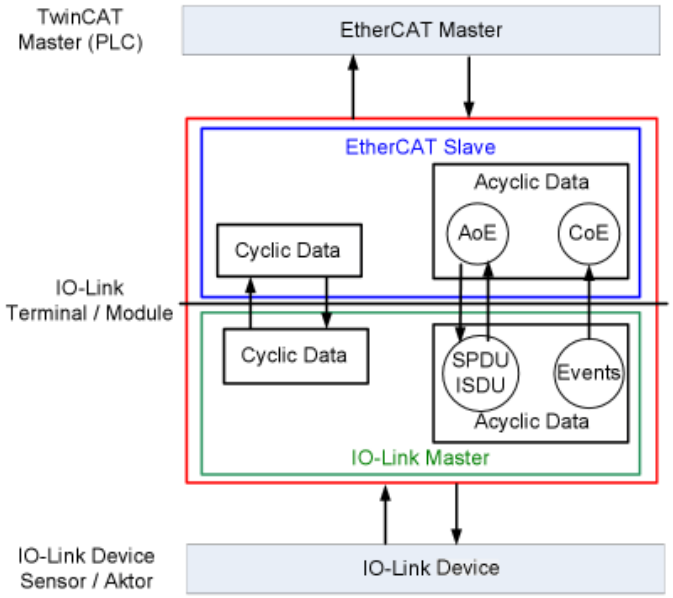


Fig. 58: Illustration of the system communication of an EtherCAT master

In principle cyclic and acyclic data are exchanged. The cyclic process data can be accessed via the PDOs [▶ 76], the acyclic data via AoE [▶ 77]. The events are additionally displayed under Diag History [▶ 80] in the System Manager.

- **Cyclic data:**
  - Process data
  - Value status
- **Acyclic data:**
  - Device data
  - Events

4.3.2 PDO Assignment

The scope of the process data offered varies depending on the configured IO-Link ports.
“DeviceState Inputs Device” and “DeviceState Inputs” are selected by default. Device-specific PDOs (0x1A0n “Port (n-1) Process Data”) are only displayed after configuring on the respective port and restarting the EtherCAT system or reloading the configuration in Config mode; see Activating the configuration [▶ 44].

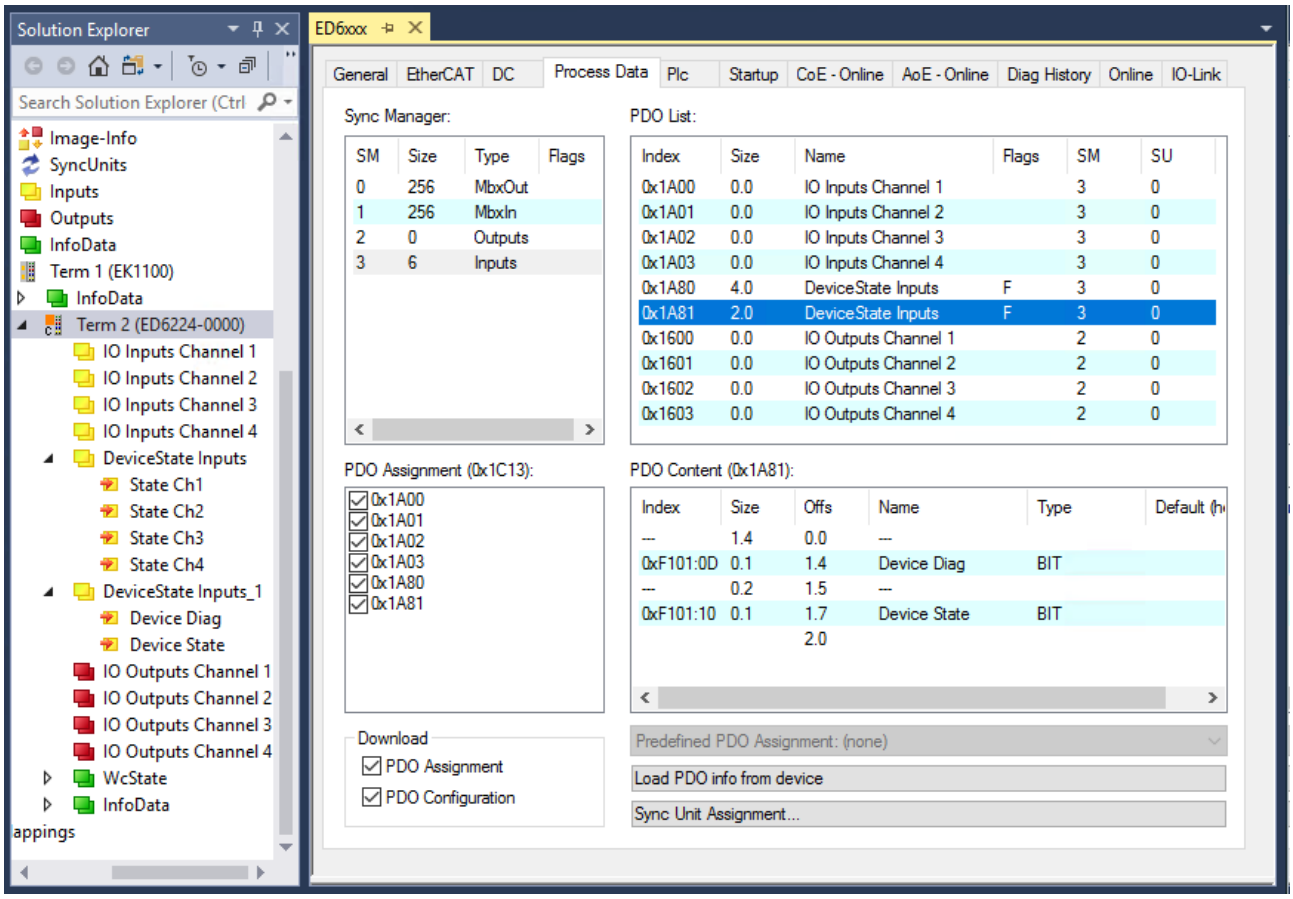


Fig. 59: Illustration of the process data allocation, SM3 inputs using the ED6224 as an example

SM3, PDO Assignment 0x1C13			
Index	**Size (byte.bit)**	**Name**	**PDO content**
0x1A81	2.0	DeviceState Inputs Device	Index 0xF101:0D - Device Diag Index 0xF101:10 - Device State
0x1A80	4.0	DeviceState Inputs	Index 0xF100:01 - State Ch1 Index 0xF100:02 - State Ch2 Index 0xF100:03 - State Ch3 Index 0xF100:04 - State Ch4
0x1A00	0.0 - 32.0	Port 1 Process Data	IO-Link device-specific / only active after configuration
0x1A01	0.0 - 32.0	Port 2 Process Data	IO-Link device-specific / only active after configuration
0x1A02	0.0 - 32.0	Port 3 Process Data	IO-Link device-specific / only active after configuration
0x1A03	0.0 - 32.0	Port 4 Process Data	IO-Link device-specific / only active after configuration

Process data representation

If data types are used that don’t conform to IEC61131-3, they are represented as octet strings.

The status of the IO-Link ports [▶ 81] 1 to 4 is displayed via index 0xF100:0n.
The indexes 0xF101:xx provide general diagnosis data.

Index	Size (byte.bit)	Name	Meaning
0xF101:0D	0.1	Device Diag	Occurrence of events (on the device side) is reported via a status bit
0xF101:10	0.1	Device State	Interruption of communication with one of the slaves is reported via a status bit
0xF100:01	1.0	State Ch.1	See table “Meaning of status byte Ch. 1 - Ch. 4”
0xF100:02	1.0	State Ch.2	
0xF100:03	1.0	State Ch.3	
0xF100:04	1.0	State Ch.4	

The status bytes are divided into two nibbles.

Meaning Status byte Ch. 1 - Ch. 4
Low nibble: 0x_0 = Port disabled 0x_1 = Port in std dig in 0x_2 = Port in std dig out 0x_3 = Port in communication OP 0x_4 = Port in communication COMSTOP / dig in Bit (only in std. IO Mode) 0x_5 = not defined 0x_6 = not defined 0x_7 = not defined 0x_8 = Process Data Invalid Bit Combinations are possible and are displayed as addition of the values (s. note)
Higher nibble: 0x1_ = Watchdog detected 0x2_ = internal Error 0x3_ = invalid Device ID 0x4_ = invalid Vendor ID 0x5_ = invalid IO-Link Version 0x6_ = invalid Frame Capability 0x7_ = invalid Cycle Time 0x8_ = invalid PD in length 0x9_ = invalid PD out length 0xA_ = no Device detected 0xB_ = error PreOP/Data storage Combinations are possible and are displayed as addition of the values (s. note)

Addition of the values in case of simultaneously occurring diagnostic messages

If messages occur simultaneously, the value is displayed as a sum in the Status byte of the relevant channel.

- Often for example 0x03 “Port in communication OP” and 0x08 “Process Data Invalid Bit” occur simultaneously:
  0x03 + 0x08 = 0x0B (11_{dec})

⇨ The value 0x0B (11_{dec}) is displayed in the Status byte.

4.3.3 Accessing IO-Link parameters

The exchange of the acyclic data takes place via a specified index and subindex range that is device-specific and can be read about in the corresponding vendor documentation.

4.3.4 Parameter data exchange

An intelligent IO-Link sensor/actuator can support parameterization by ISDUs (Indexed Service Data Unit). The PLC must explicitly query or, when marked as such, send these acyclic service data.

ISDU access

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:

Designation	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 to 0xFFFF is reserved

The use and implementation of these ranges is the responsibility of the sensor/actuator manufacturer. For clarification, just a few of the possible indices are listed here. Please take a look at the relevant chapter “Object description and parameterization”.

Index	Name
0010	Vendor Name
0011	Vendor Text
0012	Product Name
0013	Product ID
0015	Serial Number
0016	Hardware Revision
0017	Firmware Revision
...	...

IO-Link operating modes

The IO-Link ports on the IO-Link master can be operated in the following nine modes (see Object description and parameterization - IO-Link State, Index 0xA0n0 [▶ 66]:01):

- INACTIVE: Statemachine is inaktive
- DIGINPUT: The port behaves like a digital input
- DIGOUTPUT: The port behaves like a digital output
- ESTABLISHCOMM: The IO-Link wakeup sequence is running
- INITMASTER: Readout the IO-Link device and check the communication parameters
- INITDEVICE: Initialization of the IO-Link device
- PREOPERATE: Parameter server is running
- OPERATE: The port is used for IO-Link communication
- STOP: Communication is stopped (COM-stop)

4.3.5 ADS

Communication relating to IO-Link demand data is initiated via an ADS command. An ADS address always consists of a NetID and PortNo. An ADS command is forwarded from TwinCAT to the ED/EJ/EL6224-00x0 via AoE (**A**DS **o**ver **E**therCAT). From there, the command is relayed to the IO-Link master section and therefore to the demand data channel.

AoE NetID

The ED/EJ/EL6224-00x0 is assigned a dedicated AoE-NetID for communication with the IO-Link master section. This is assigned by the configuration tool (see the following figure).
NetID at “ED/EJ/EL6224-00x0” -> “EtherCAT” -> “Advanced Settings” -> “Mailbox” -> “AoE”

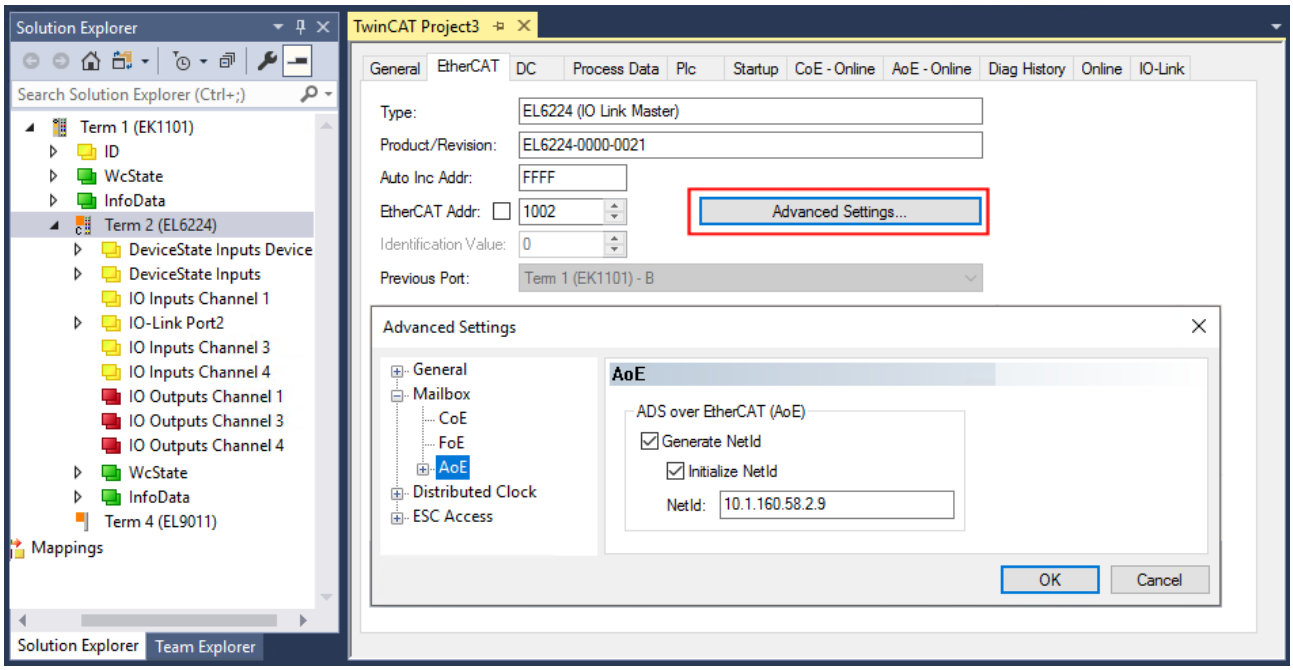


Fig. 60: Assignment of AoE-NetID using the example of EL6224

PortNo

The individual IO-Link ports for the master are allocated via the port number. The port numbers are allocated in ascending order from 0x1000. This means that IO-Link port 1 corresponds to PortNo 0x1000 and IO-Link Portn corresponds to PortNo 0x1000 + n-1.
The following specification applies to Beckhoff’s 4-port IO-Link masters:

- IO-Link port 1 corresponds to PortNo 0x1000
- IO-Link port 2 corresponds to PortNo 0x1001
- IO-Link port 3 corresponds to PortNo 0x1002
- IO-Link port 4 corresponds to PortNo 0x1003

ADS Indexgroup

As for CoE, the Indexgroup of an ADS command is specified as 0xF302 for the IO link data channel.

ADS Indexoffset

The IO link addressing with index and subindex is coded in the Indexoffset. The Indexoffset has a size of 4 bytes and is subdivided as follows: 2-byte index, 1-byte reserve, 1-byte subindex.

- Example: Indexoffset 0x12340056 corresponds to index 0x1234 and subindex 56

4.3.6 Access to events

Some of the IO-Link sensors forward events that occur to the master. These events may be items of information, warnings or error messages, e.g. short circuit or overheating.
The IO-Link master reports these events by setting the "Device Diag [▶ 76]" bit. Further information on the events can be read via the CoE directory or the "Diag History" tab.

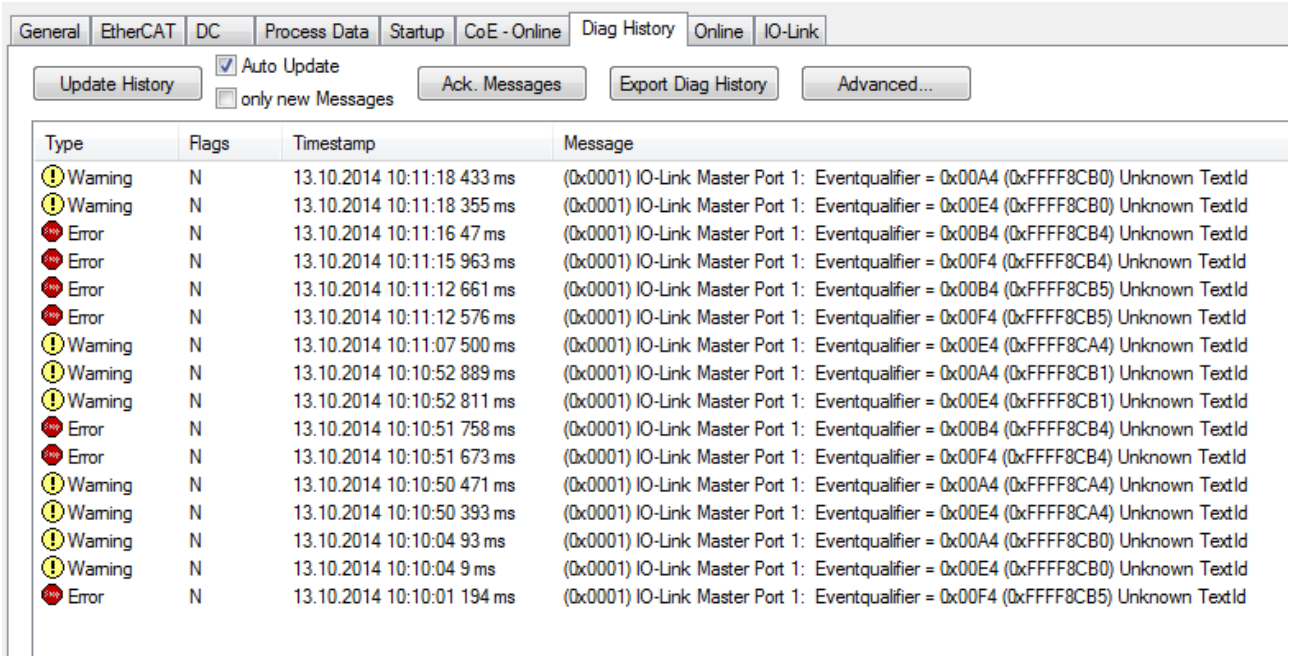


Fig. 61: Diag History tab

The events are arranged according to type (information, warning, error), flag, occurrence of the event (time stamp) and message (port number and event code).
The meaning of the individual messages can be taken from the vendor documentation. The IO-Link device can be directly allocated on the basis of the port number. The events occurring can be managed using the various buttons.

- **Update History:** if the "Auto Update" field is not selected, then the current events can be displayed via the "Update History" button
- **Auto Update:** if this field is selected, then the list of events occurring is automatically updated
- **Only new Messages:** if this field is selected, then only those messages that have not yet been confirmed are displayed.
- **Ack. Messages:** an event that occurs is reported via the “Device Diag [▶ 76]”. Confirming the message will reset the bit to 0.
- **Export Diag History:** the events that have occurred can be exported as a "txt" file and thus archived.
- **Advanced:** this field currently (as at 3rd quarter 2015) has no function.

4.3.7 PLC library: Tc3_IoLink

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.

See software documentation in the Beckhoff Information System:

TwinCAT 3 | PLC Library: Tc3_IoLink

4.4 Diagnostics

4.4.1 State of the IO-Link Ports

There is a status byte for each IO-Link port. You can find the status bytes at the following locations:

- In the process data object “DeviceState Inputs”
- In the CoE object 0xF100 [▶ 66]

The meaning of the status byte is displayed in the TwinCAT System Manager in the “Comment” field of the “Variable” tab (see following figure).

Please also note the information on the addition of the values in case of simultaneously occurring diagnosis messages in the PDO assignment [▶ 77] chapter.

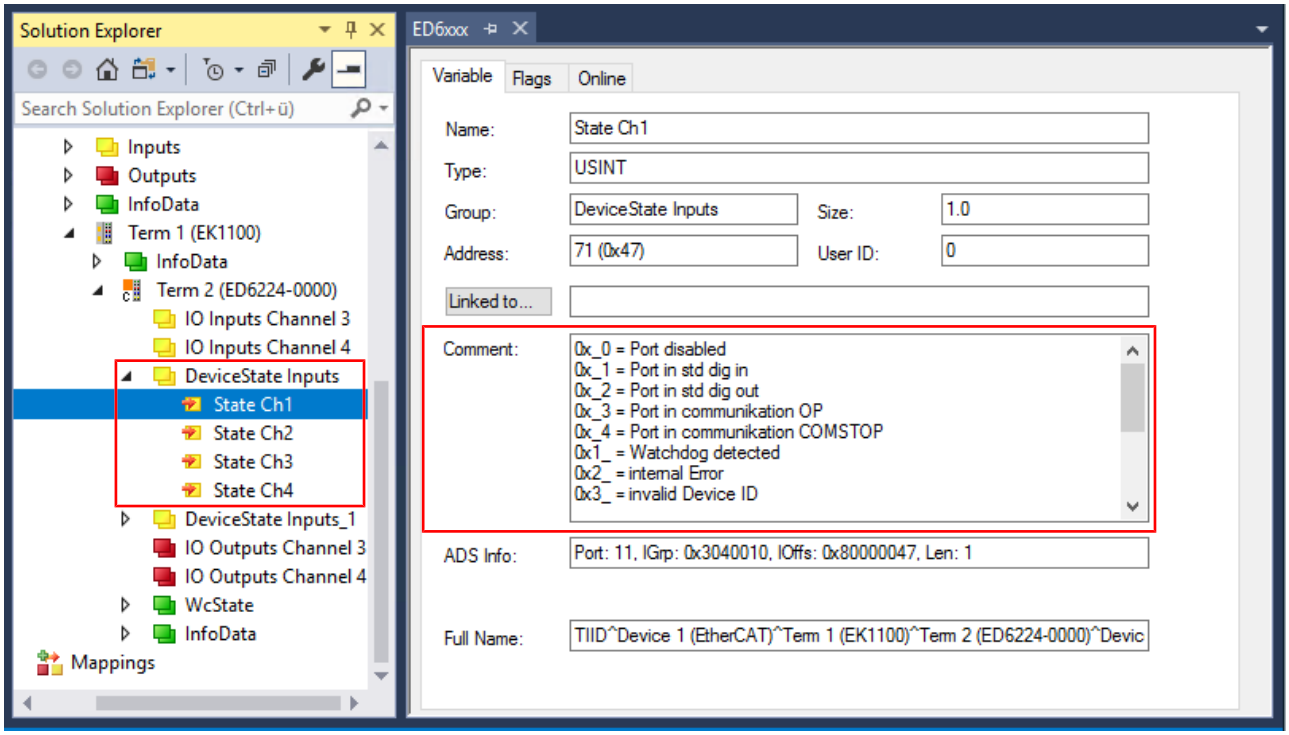


Fig. 62: Display process data object “DeviceState Inputs” and meaning of the status byte in the “Comment” field

4.4.2 ADS 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 tables C.1 and C.2.

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	

Table C.1 ErrorTypes, IO-Link Spec

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

Table C.2 Derived ErrorTypes, IO-Link Spec

4.4.3 Further error diagnostics

Device State Inputs Device (0x1A81)

- The “Device Diag [▶ 76]” PDO (0xF101:0D) indicates that at least one event has occurred in the “Diag History”.
- “Device State [▶ 76]” 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 (0x1A80)

The status of the IO-Link devices is displayed at the respective port (see “Comment [▶ 81]” field in the System Manager).

Nominal/actual comparison of the parameter objects

The indices 0x90n0 (Info data) can be referred to for validation of the configuration indices 0x80n0 of the connected IO-Link device.
In case of 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 [▶ 66] 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.

5 Appendix

5.1 Firmware compatibility

Beckhoff EtherCAT devices are delivered with the latest available firmware version. Firmware and hardware must be compatible; not every combination ensures compatibility. The overview below shows the hardware versions on which a firmware can be operated.

Note

- The newest possible firmware for the respective hardware should be implemented
- Beckhoff is not under any obligation to provide customers with free firmware updates for delivered products.

NOTICE

Risk of damage to the device!

Observe the notes on the firmware update in the EtherCAT System Manual in chapter “Firmware Update”.
If a device is set to BOOTSTRAP mode for firmware update, it may not check whether the new firmware is suitable during download.
This may result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

ED6224			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00*	01*	ED6224-0000-0016	2026/02

*) This is the current compatible firmware/hardware version at the time of the preparing this documentation. Check on the Beckhoff website whether more up-to-date documentation is available.

5.2 Documentation issue status

Version	Comment
1.1.0	• Update chapter “Technical data” • Update structure
1.0.0	• First release for ED6224
0.1	• Preliminary documentation for ED6224

5.3 Support and Service

Beckhoff and their partners around the world offer comprehensive support and service, making available fast and competent assistance with all questions related to Beckhoff products and system solutions.

Beckhoff's branch offices and representatives

Please contact your Beckhoff branch office or representative for local support and service on Beckhoff products!

The addresses of Beckhoff's branch offices and representatives round the world can be found on her internet pages: www.beckhoff.com

You will also find further documentation for Beckhoff components there.

Support

The Beckhoff Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
e-mail:	support@beckhoff.com
web:	www.beckhoff.com/support

Service

The Beckhoff Service Center supports you in all matters of after-sales service:

- on-site service
- repair service
- spare parts service
- hotline service

Hotline:	+49 5246 963 460
e-mail:	service@beckhoff.com
web:	www.beckhoff.com/service

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

Trademark statements

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

Third-party trademark statements

IO-Link is a registered trademark of PROFIBUS Nutzerorganisation e.V.

More Information:
www.beckhoff.com/ed6224

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