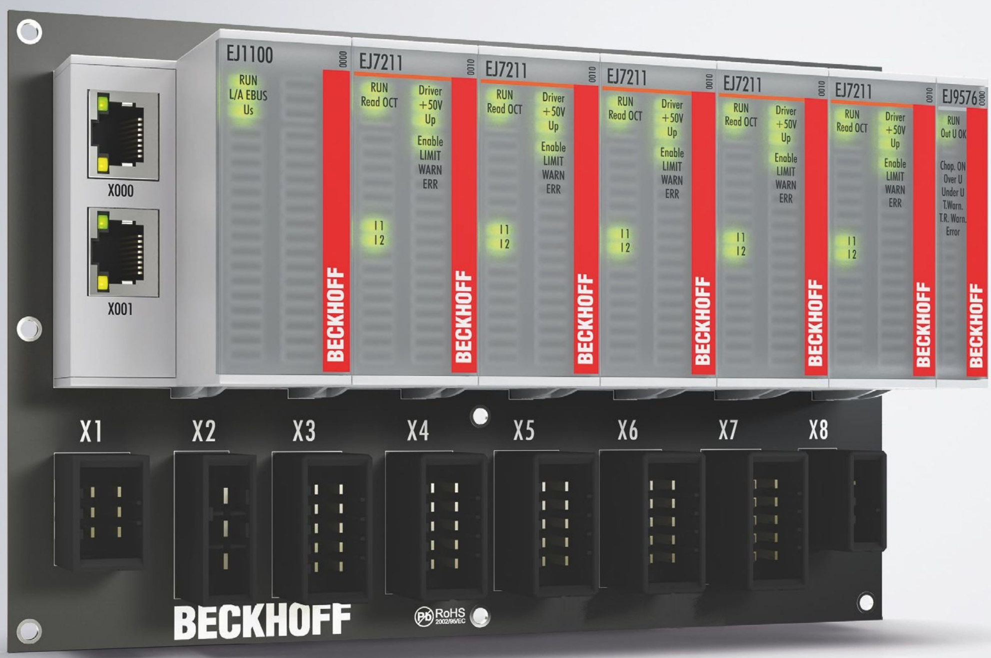
BECKHOFF New Automation Technology

Documentation | EN

EJ6224, EJ6224-0090

IO-Link Master



2023-02-16 | Version: 1.9

Table of contents

1 Foreword

1.1 Product overview IO-Link master

EJ6224 [▶ 17]	IO-Link master
EJ6224-0090 [▶ 17]	IO-Link master with TwinSAFE SC

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.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

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

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

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

Trademarks

Beckhoff®, TwinCAT®, TwinCAT/BSD®, TC/BSD®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® and XPlanar® are registered trademarks of and licensed by Beckhoff Automation GmbH. Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

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

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

Copyright

1.3 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

In this documentation the following instructions are used.
These instructions must be read carefully and followed without fail!

⚠ DANGER

Serious risk of injury!

Failure to follow this safety instruction directly endangers the life and health of persons.

⚠ WARNING

Risk of injury!

Failure to follow this safety instruction endangers the life and health of persons.

⚠ CAUTION

Personal injuries!

Failure to follow this safety instruction can lead to injuries to persons.

NOTE

Damage to environment/equipment or data loss

Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.4 Intended use

⚠ WARNING

Caution - Risk of injury!

EJ components may only be used for the purposes described below!

1.5 Signal distribution board

NOTE

Signal distribution board

Make sure that the EtherCAT plug-in modules are used only on a signal distribution board that has been developed and manufactured in accordance with the Design Guide.

1.6 Documentation issue status

Version	Comment
1.9	• Update chapter *Pinout*
1.8	• Chapter *PLC library: Tc3_IoLink* added
1.7	• Update chapter *Product description* • Update chapter *Installation of EJ-Modules* • Added chapters: ◦ *IO-Link basics* ◦ *IO-Link Configuration and parameterizing* ◦ *Access to IO-Link data* ◦ *Diagnostics* • Structural update
1.6	• Structural update
1.5	• Update chapter *Marking of EtherCAT plug-in modules*
1.4	• New title page • Update chapter *Product description* • Chapters *Basics communication, TwinCAT Quick Start* , *TwinCAT development environment* and *General Notes - EtherCAT Slave Application* replaced by references in the chapter *Guide through the documentation* • Chapter *Disposal* added • Chapter *Object description and parameterization* added • Update revision status • Structural update
1.3	• Update chapter *EJ6224-00x0-Pinout*
1.2	• Note *Signal distribution board* added • Chapter *Version identification of EtherCAT devices* replaced by chapter *Marking of EtherCAT plug-in modules* • Update Technical data • Update chapter *EJ6224-00x0-Pinout*
1.1	• Chapter *TwinSAFE SC* added
1.0	• First publication EJ6224, EJ6224-0090

1.7 Guide through documentation

NOTE

Further components of documentation

The documentations named in the following table are further components of the complete documentation. These documentations are required for the use of EtherCAT plug-in modules.

No.	Title	Description
[1]	EtherCAT System Documentation	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • Distributed Clocks • Configuration of EtherCAT-Components
[2]	Infrastructure for EtherCAT/Ethernet	• Technical recommendations and notes for design, implementation an testing
[3]	Design GuideSignal-Distribution-Board for standard EtherCAT plug-in modules	Requirements for the design of a Signal-Distribution-Board for standard EtherCAT plug-in modules • Backplane mounting guidelines • Module placement • Routing guidelines

1.8 Marking of EtherCAT plug-in modules

Designation

A Beckhoff EtherCAT device has a 14-digit **technical designation**, made up as follows (e.g. EJ1008-0000-0017)

- **Order identifier**
  - family key: EJ
  - product designation: The first digit of product designation is used for assignment to a product group (e.g. EJ2xxx = digital output module).
  - Version number: The four digit version number identifies different product variants.
- **Revision number:**
  It is incremented when changes are made to the product.

The Order identifier and the revision number are printed on the side of EtherCAT plug-in modules (s. following illustration (A and B).

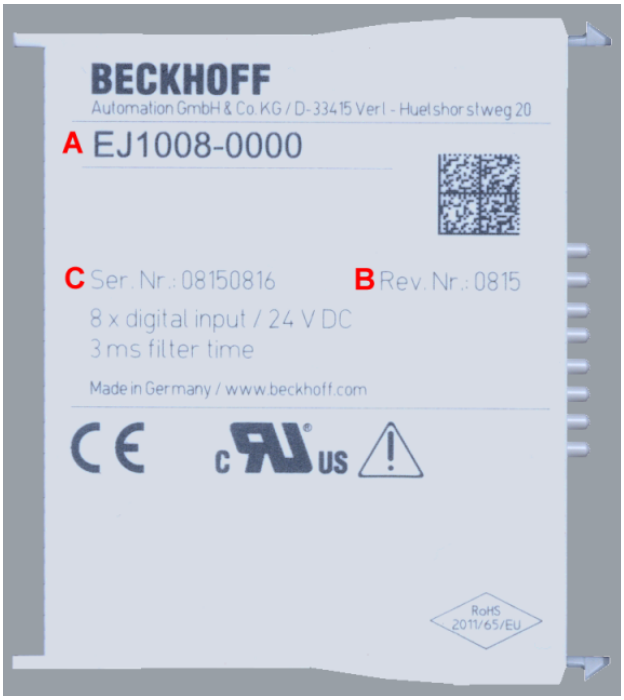


Fig. 1: Order identifier (A), Revision number (B) and serial number (C) using the example of EJ1008

Product group	Example		
	Product designation	Version	Revision
EtherCAT Coupler EJ11xx	EJ1101	-0022 (Coupler with external connectors, power supply module and optional ID switches	-0016
Digital input modules EJ1xxx	EJ1008 8-channel	-0000 (basic type)	-0017
Digital output modules EJ2xxx	EJ2521 1-channel	-0224 (2 x 24 V outputs)	-0016
Analog input modules EJ3xxx	EJ3318 8-channel thermocouple	-0000 (basic type)	-0017
Analog output modules EJ4xxx	EJ4134 4-channel	-0000 (basic type)	-0019
Special function modules EJ5xxx, EJ6xxx	EJ6224 IO-Link master	-0090 (with TwinSAFE SC)	-0016
Motion modules EJ7xxx	EJ7211 servomotor	-9414 (with ECT, STO and TwinSAFE SC)	-0029

Notes

- The elements mentioned above result in the **technical designation**. EJ1008-0000-0017 is used in the example below.
- EJ1008-0000 is the **order identifier**, in the case of “-0000” usually abbreviated to EJ1008.
- The **revision** -0017 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
- The product designation, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

Serial number

The serial number for EtherCAT plug-in modules is usually the 8-digit number printed on the side of the module (see following illustration C). The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

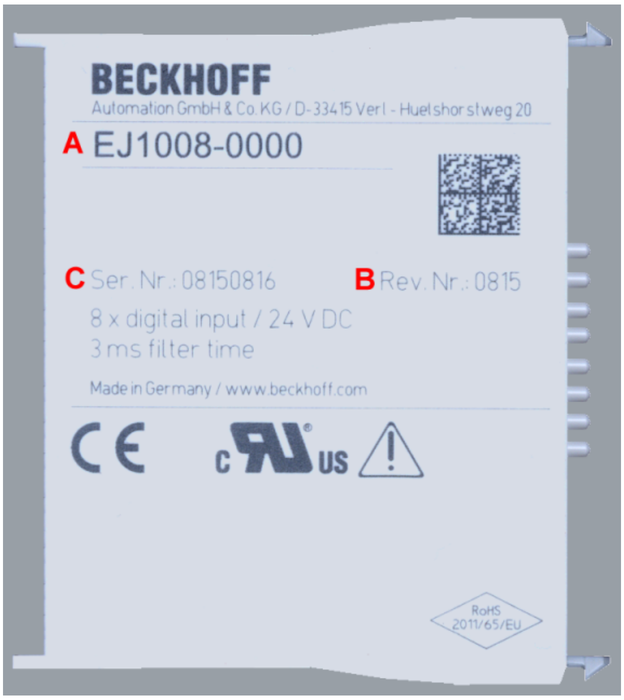


Fig. 2: Order identifier (A), revision number (B) and serial number (C) using the example of EJ1008

Serial number	Example serial number: 08 15 08 16
KK - week of production (CW, calendar week)	08 - week of production: 08
YY - year of production	15 - year of production: 2015
FF - firmware version	08 -f irmware version: 08
HH - hardware version	16 - hardware version: 16

1.8.1 Beckhoff Identification Code (BIC)

The **B**eckhoff **I**dentification **C**ode (BIC) is increasingly being applied to Beckhoff products to uniquely identify the product. The BIC is represented as a Data Matrix Code (DMC, code scheme ECC200), the content is based on the ANSI standard MH10.8.2-2016.

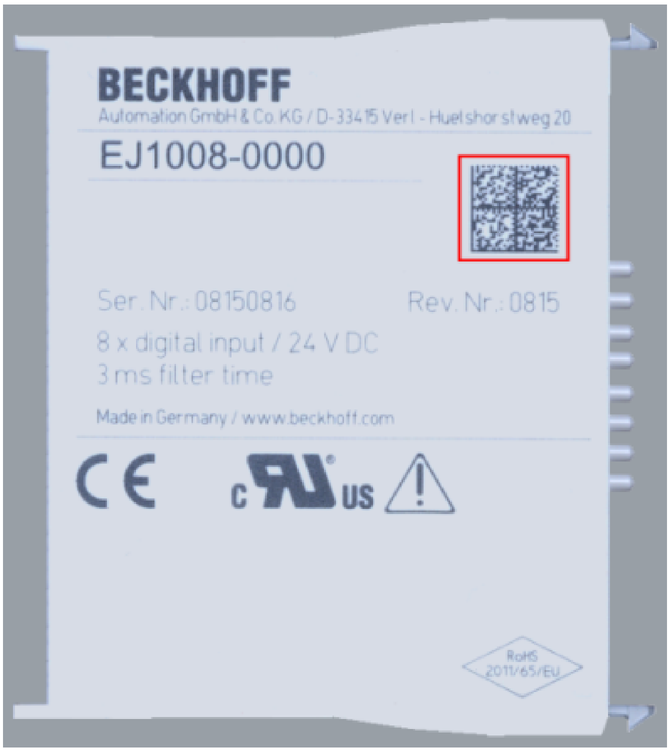


Fig. 3: BIC as data matrix code (DMC, code scheme ECC200)

The BIC will be introduced step by step across all product groups.

Depending on the product, it can be found in the following places:

- on the packaging unit
- directly on the product (if space suffices)
- on the packaging unit and the product

The BIC is machine-readable and contains information that can also be used by the customer for handling and product management.

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, it shall be replaced by spaces. The data under positions 1-4 are always available.

The following information is contained:

Item no.	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	S	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294104
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971 , 2*K183
...					

Further types of information and data identifiers are used by Beckhoff and serve internal processes.

Structure of the BIC

Example of composite information from items 1 - 4 and with the above given example value on positon 6. The data identifiers are marked in bold font for better display:

1P072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

Fig. 4: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, item no. 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTE

This information has been carefully prepared. However, the procedure described is constantly being further developed. We reserve the right to revise and change procedures and documentation at any time and without prior notice. No claims for changes can be made from the information, illustrations and descriptions in this information.

1.8.2 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

The Beckhoff Identification Code (BIC) is applied to the outside of Beckhoff products in a visible place. If possible, it should also be electronically readable.

Decisive for the electronic readout is the interface via which the product can be electronically addressed.

K-bus devices (IP20, IP67)

Currently, no electronic storage and readout is planned for these devices.

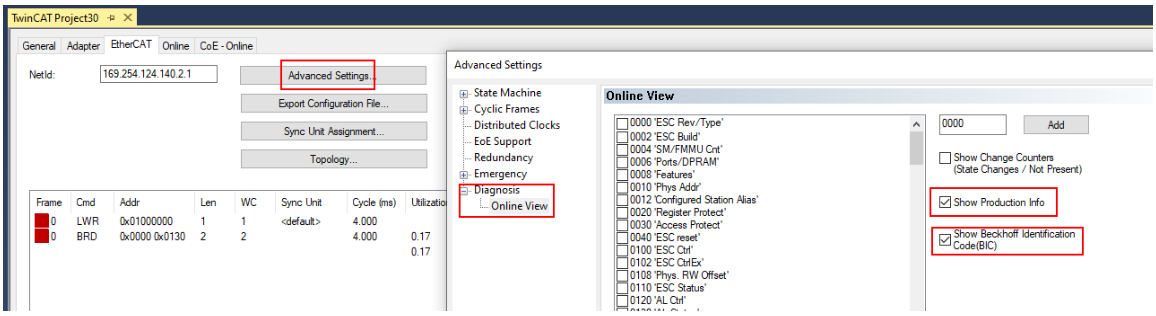
EtherCAT devices (IP20, IP67)

All Beckhoff EtherCAT devices have a so-called ESI-EEPROM, which contains the EtherCAT identity with the revision number. Stored in it is the EtherCAT slave information, also colloquially known as ESI/XML configuration file for the EtherCAT master. See the corresponding chapter in the EtherCAT system manual (Link) for the relationships.

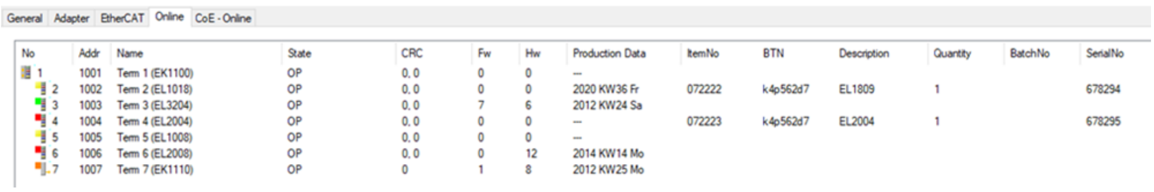
The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

The user can electronically access the eBIC (if existent) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM
  - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.
  - To do this,
    check the checkbox "Show Beckhoff Identification Code (BIC)" under
    EtherCAT → Advanced Settings → Diagnostics:



  - The BTN and its contents are then displayed:



  - Note: as can be seen in the illustration, the production data HW version, FW version and production date, which have been programmed since 2012, can also be displayed with "Show Production Info".
  - From TwinCAT 3.1. build 4024.24 the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.
- In the case of EtherCAT devices with CoE directory, the object 0x10E2:01 can additionally by used to display the device's own eBIC; the PLC can also simply access the information here:

- The device must be in PREOP/SAFEOP/OP for access:

Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
+ 1011:0	Restore default parameters	RO	> 1 <
+ 1018:0	Identity	RO	> 4 <
- 10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016
+ 10F0:0	Backup parameter handling	RO	> 1 <
+ 10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

- the object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.
- From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.

- Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
  The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored with the help of a category according to ETG.2010. ID 03 indicates to all EtherCAT masters that they must not overwrite these data in case of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see there. This results in a memory requirement of approx. 50..200 bytes in the EEPROM.
- Special cases
  - If multiple, hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC Information.
  - If multiple, non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC Information.
  - If the device consists of several sub-devices with their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

Profibus/Profinet/DeviceNet… Devices

Currently, no electronic storage and readout is planned for these devices.

1.8.3 Certificates

- The EhterCAT plug-in modules meet the requirements of the EMC and Low Voltage Directive. The CE mark is printed on the side of the modules.
- The cRUus imprint identifies devices that meet product safety requirements according to U.S. and Canadian regulations.
- The warning symbol is a request to read the corresponding documentation. The documentations for EtherCAT plug-in modules can be downloaded from the Beckhoff homepage.

Fig. 5: Marking for CE and UL using EJ1008 as an example

2 System overview

Electronically, the EJxxxx EtherCAT plug-in modules are based on the EtherCAT I/O system. The EJ system consists of the signal distribution board and EtherCAT plug-in modules. It is also possible to connect an IPC to the EJ system.
The EJ system is suitable for mass production applications, applications with small footprint and applications requiring a low total weight.
The machine complexity can be extended by means of the following:

- reserve slots,
- the use of placeholder modules,
- linking of EtherCAT Terminals and EtherCAT Boxes via an EtherCAT connection.

The following diagram illustrates an EJ system. The components shown are schematic, to illustrate the functionality.

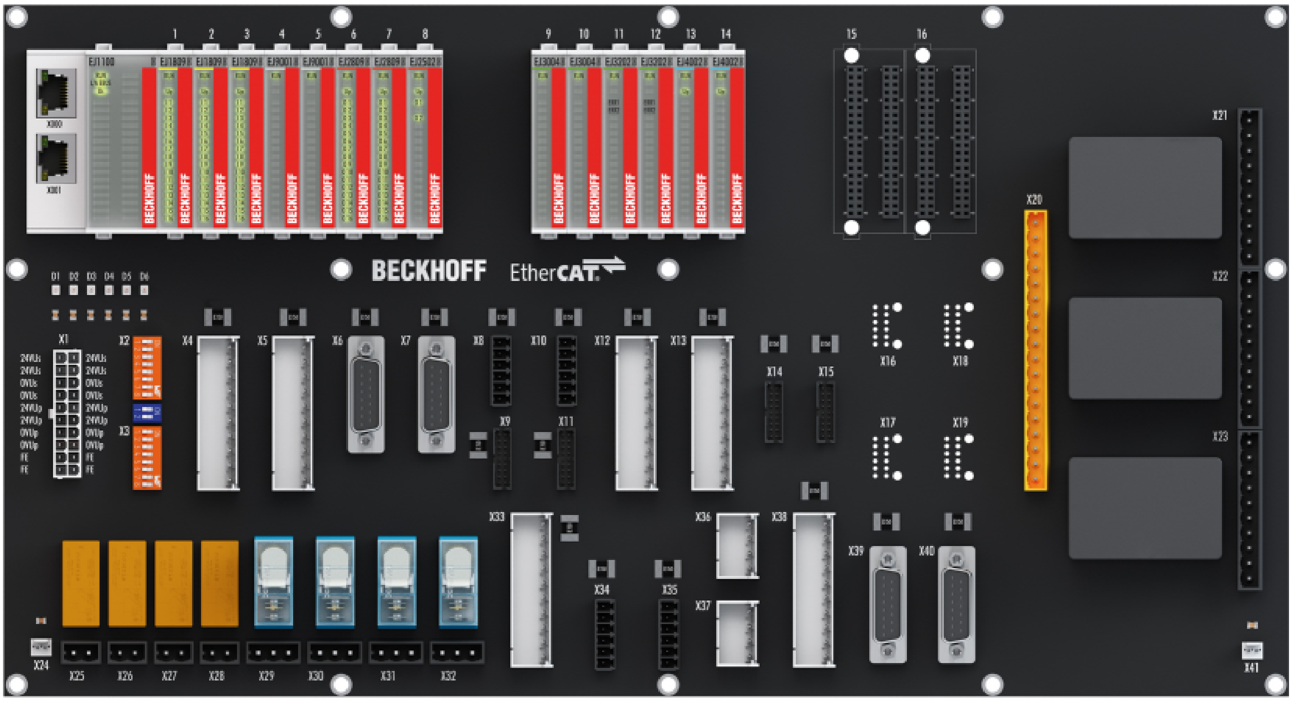


Fig. 6: EJ system sample

Signal distribution board

The signal distribution board distributes the signals and the power supply to individual application-specific plug connectors, in order to connect the controller to further machine modules. Using pre-assembled cable harnesses avoids the need for time-consuming connection of individual wires. Coded components reduce the unit costs and the risk of miswiring.
Beckhoff offers development of signal distribution boards as an engineering service. Customers have the option to develop their own signal distribution board, based on the design guide.

EtherCAT plug-in modules

Similar to the EtherCAT Terminal system, a module strand consists of a bus coupler and I/O modules. Almost all of the EtherCAT Terminals can also be manufactured in the EJ design as EtherCAT plug-in modules. The EJ modules are directly attached to the signal distribution board. The communication, signal distribution and supply take place via the contact pins at the rear of the modules and the PCB tracks of the signal distribution board. The coding pins at the rear serve as mechanical protection against incorrect connection. Color coding on the housing facilitates distinguishing of the modules.

3 EJ6224, EJ6224-0090 - Product description

3.1 Introduction

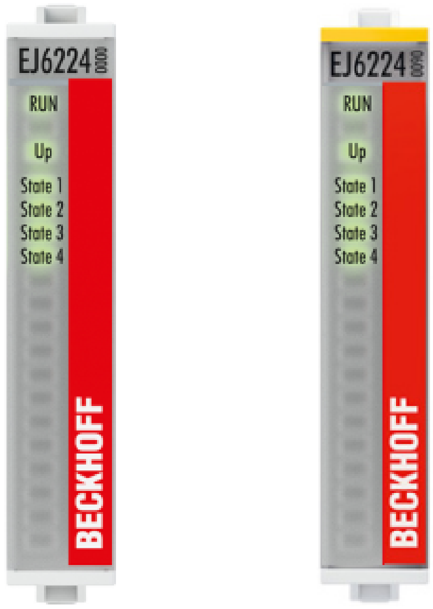


Fig. 7: EJ6224, EJ6224-0090

IO-Link master

The IO-Link masters EJ6224 and EJ6224-0090 enable the connection of up to four IO-Link devices. IO-Link devices are actuators, sensors or a combination of both. The connection between the EtherCAT plug-in module and the device is a point-to-point connection. The EJ6224-xxxx EtherCAT plug-in modules are parameterized via the EtherCAT master. IO-Link is designed as an intelligent link between the fieldbus level and the sensor, wherein parameterization information can be exchanged bidirectionally via the IO-Link connection.

The parameterization of the IO-Link devices with service data is carried out from TwinCAT via ADS or very conveniently via the integrated IO-Link commissioning tool. The tool enables automatic scanning of the IO-Link devices, convenient sensor parameter processing and an integrated online search for sensor description files.

The EJ6224-0090 EtherCAT plug-in module supports the full range of functions of the EJ6224 as well as TwinSAFE SC technology (TwinSAFE Single Channel).

The TwinSAFE SC (Single Channel) technology enables the use of standard signals for safety tasks in any networks of fieldbuses. The standard functionalities and features of the I/Os are retained. The data of the TwinSAFE SC I/Os are routed to the TwinSAFE logic and processed there in a multichannel safety-related manner. In the Safety Logic the data originating from different sources are analyzed, checked for plausibility and submitted to a ‘voting’. Certified function blocks such as Scale, Compare/Voting (1oo2, 2oo3, 3oo5), Limit etc. are used for this purpose. For safety reasons, however, at least one of the data sources must be a TwinSAFE SC component. The other data can come from other standard I/Os, drive controllers or measuring transducers.

With the aid of the TwinSAFE SC technology it is typically possible to achieve a safety level equivalent to PL d/Cat. 3 in accordance with EN ISO 13849-1 or SIL 2 in accordance with EN 62061.

3.2 Technical Data

Technical data	EJ6224	EJ6224-0090
IO-Link interfaces	4	
Specification version	IO-Link V1.1	
Field voltage	24 V_{DC} (via distribution board)	
Connection technology	3-wire	
Data transfer rates	4.8 kBaud, 38.4 kBaud and 230.4 kBaud	
Cable length between IO-Link master and device	max. 20 m	
Power supply for the electronics	via the E-bus and signal distribution board	
Current consumption via E-bus	typ. 110 mA	
Special features	**EJ6224 and EJ6224-0090:** IO-Link commissioning tool integrated in TwinCAT: • automatic device scan, convenient sensor parameter processing, • integrated online search for sensor description files **EJ6224-0090**: TwinSAFE SC [▶ 100] Technology	
MTBF (+55 °C)	-	> 1,200,000
Electrical isolation	500 V (E-bus/field voltage)	
Supply current for devices	500 mA per device	
Configuration	via TwinCAT System Manager	
Permissible ambient temperature range during operation	-25°C ... +60°C (extended temperature range)	
Permissible ambient temperature range during storage	-40°C ... +85°C	
Permissible relative air humidity	95 %, no condensation	
Operating altitude	max. 2,000 m	
Dimensions (W x H x D)	approx. 12 mm x 66 mm x 55 mm	
Weight	approx. 30 g	
Mounting	on signal distribution board	
Pollution degree	2	
Mounting position	Standard [▶ 29]	
Position of the coding pins [▶ 32]	2 and 5	
Color coding	grey	
Vibration/shock resistance	conforms to EN 60068-2-6 /EN 60068-2-27 (with corresponding signal distribution board)	
EMC immunity/emission	conforms to EN 61000-6-2 /EN 61000-6-4 (with corresponding signal distribution board)	
Protection class	EJ module: IP20 EJ system: dependent on the signal distribution board and housing	
Approvals/markings*	CE, UKCA, EAC, UL	CE, UKCA, EAC

*) Real applicable approvals/markings see type plate on the side (product marking).

CE approval

The CE Marking refers to the EtherCAT plug-in module mentioned above.
If the EtherCAT plug-in module is used in the production of a ready-to-use end product (PCB in conjunction with a housing), the manufacturer of the end product must check compliance of the overall system with relevant directives and CE certification.
To operate the EtherCAT plug-in modules, they must be installed in a housing.

3.3 Pinout

EJ6224, EJ6224-0090				
Pin#		Signal		
1	2	U_{EBUS}	U_{EBUS}	**E-Bus contacts** The power supply U_{EBUS} is provided by the coupler and supplied from the supply voltage U_S of the EtherCAT coupler.
3	4	GND	GND	
5	6	RX0+	TX1+	
7	8	RX0-	TX1-	
9	10	GND	GND	
11	12	TX0+	RX1+	
13	14	TX0-	RX1-	
15	16	GND	GND	
17	18	NC	L1+	**Signals**
19	20	NC	C/Q 1	
21	22	NC	L2+	
23	24	NC	C/Q 2	
25	26	NC	L3+	
27	28	NC	C/Q 3	
29	30	NC	L4+	
31	32	NC	C/Q 4	
33	34	0V Up	0V Up	**U_P-Contacts** The peripheral voltage U_P supplies the electronics on the field side.
35	36	0V Up	24V Up	
37	38	24V Up	24V Up	
39	40	SGND	SGND	

Signal	Description
U_{EBUS}	E-Bus power supply 3.3 V
GND	E-Bus GND signal. Don`t connect with 0V Up!
RXn+	Positive E-Bus receive signal
RXn-	Negative E-Bus receive signal
TXn+	Positive E-Bus transmit signal
TXn-	Negative E-Bus transmit signal
NC	Do not connect
L1+	24 V power supply (device 1) (internally connected to 24 V Up)
C/Q 1	Input C/Q 1
L2+	24 V power supply (device 2) (internally connected to 24 V Up)
C/Q 2	Input C/Q 2
L3+	24 V power supply (device 3) (internally connected to 24 V Up)
C/Q 3	Input C/Q 3
L4+	24 V power supply (device 4) (internally connected to 24 V Up)
C/Q 4	Input C/Q 4
0V Up	GND signal field side
24V Up	Power supply field side 24 V
SGND	Shield Ground

Fig. 8: EJ6224, EJ6224-0090- Pinout

The PCB footprint can be downloaded from the Beckhoff homepage (EJ6224, EJ6224-0090).

⚠ CAUTION

Damage to the devices possible!

The IO-Link Devices must be powered from the 24 V supply of the EL6224/EJ6224, otherwise the IO-Link port may be damaged (taking device 1 as an example: L1+ and 0 V Up)!

NOTE

Damage to devices possible!

- The pins named with “NC” must not be connected.
- Before installation and commissioning read the complete documentation!

3.4 LEDs

LED No.	EJ6224 EJ6224-0090
A	RUN
B	
C	Up
1	State 1
2	State 2
3	State 3
4	State 4
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	

Fig. 9: EJ6224, EJ6224-0090 - LEDs

LED	Color	Display	State	Description
RUN	green	off	Init	State of the EtherCAT State Machine: **INIT** = initialization of the plug-in module
		flashing	Pre-Operational	State of the EtherCAT State Machine: **PREOP =** function for mailbox communication and different default settings set
		single flash	Safe-Operational	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager channels and the distributed clocks. Outputs remain in safe state
		on	Operational	State of the EtherCAT State Machine: **OP =** normal operating state; mailbox and process data communication is possible
		flickering	Bootstrap	State of the EtherCAT State Machine: **BOOTSTRAP** = function for firmware updates of the plug-in module
Up	green	off	-	No $24V_{DC}$ power supply connected
		on	-	$24V_{DC}$ power supply connected
State 1 ... 4	green	on / off	-	State of the signal line (if configured as STD in / out)
		flashes briefly twice		establishing IO-Link communication
		permanently flashing		IO-Link communication established and in operation

4 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 EL6224/EJ6224 IO-Link master was subject to the IO-Link specification 1.1. At the time of the preparation of this documentation, the IO-Link specification is entering the IEC standardization and will be adopted in extended form as IEC 61131-9. The new designation SDCI will be 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 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 Beckhoff EL6224/EJ6224 IO-Link Master Terminal has four IO-Link ports. One IO-Link device can be connected to each of them. IO-Link is therefore not a fieldbus, but a point-to-point connection (see following figure).

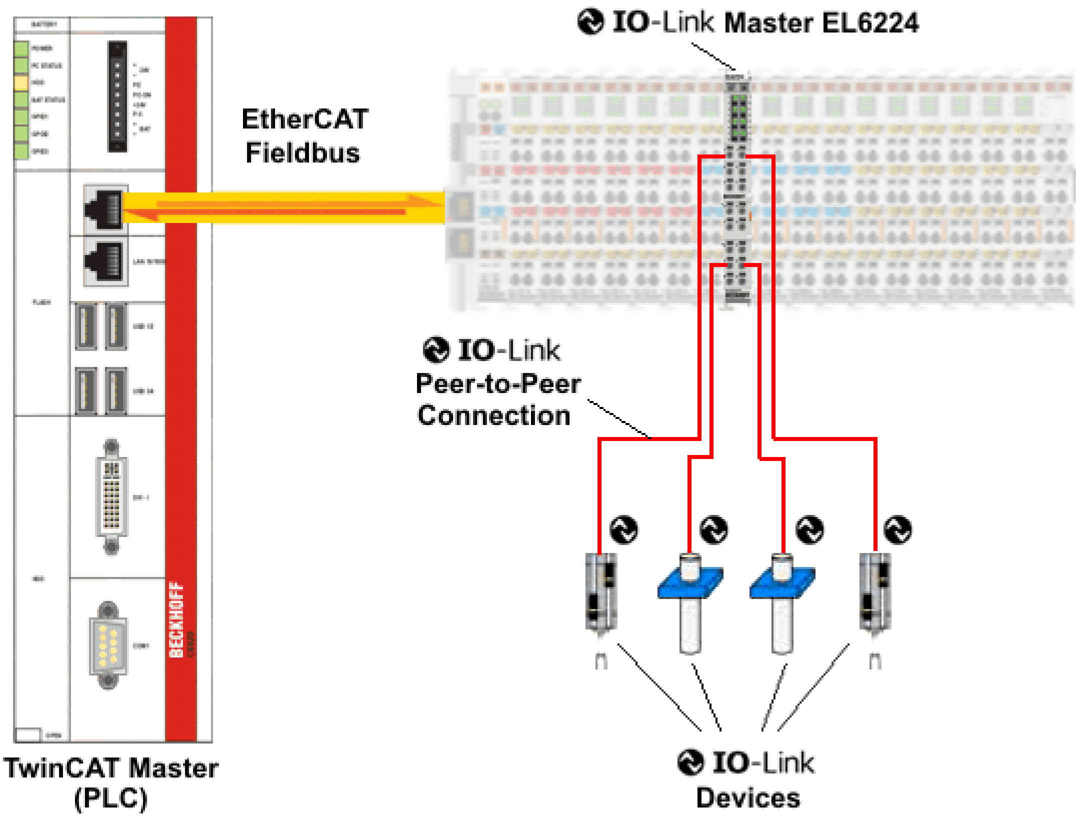


Fig. 10: IO-Link point-to-point communication using the example of the EL6224 IO-Link Master

⚠ CAUTION

Damage to the devices possible!

The IO-Link Devices must be powered from the 24 V supply of the EL6224/EJ6224, otherwise the IO-Link port may be damaged!

4.2 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 [▶ 47] the IO-Link port.

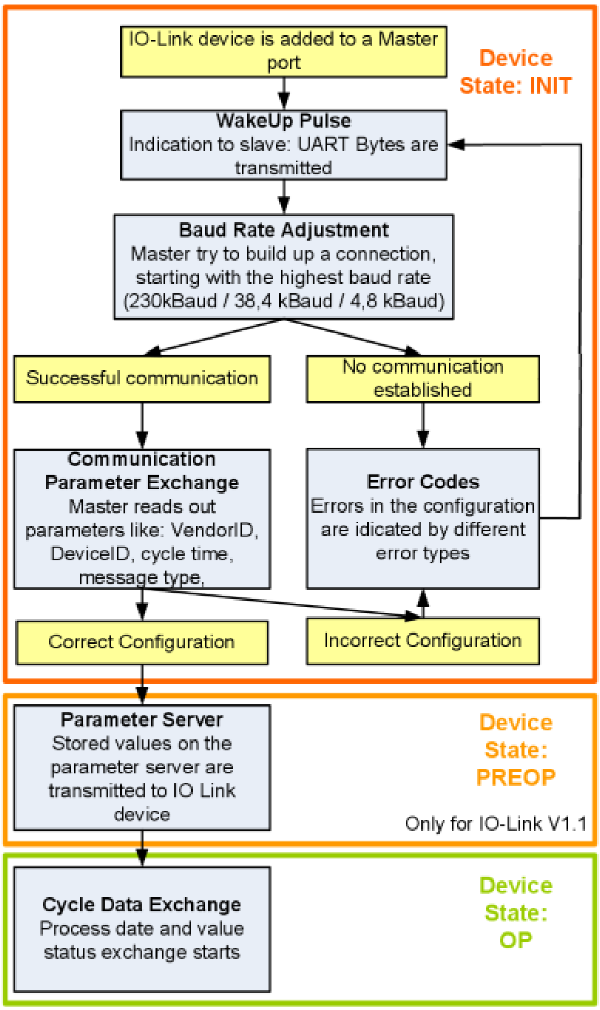


Fig. 11: Establishment of IO Link communication

- 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 [▶ 24], 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 [▶ 23] 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.

4.3 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 [▶ 47] with TwinCAT and adopted in the System Manager.

4.4 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 [▶ 54]. 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 [▶ 62] (ParamDownloadStore [▶ 63]). 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 [▶ 62] (ParamDownloadStore [▶ 63]).
  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.

4.5 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

5 Installation of EJ modules

5.1 Power supply for the EtherCAT plug-in modules

⚠ 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 (Safety Extra Low Voltage) supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor,
  a PELV (Protective Extra Low Voltage) supply also requires a safe connection to the protective conductor.

The signal distribution board should have a power supply designed for the maximum possible current load of the module string. Information on the current required from the E-bus supply can be found for each module in the respective documentation in section “Technical data”, online and in the catalog. The power requirement of the module string is displayed in the TwinCAT System Manager.

E-bus power supply with EJ1100 or EJ1101-0022 and EJ940x

The EJ1100 Bus Coupler supplies the connected EJ modules with the E-bus system voltage of 3.3 V. The Coupler can accommodate a load up to 2.2 A. If a higher current is required, a combination of the coupler EJ1101-0022 and the power supply units EJ9400 (2.5 A) or EJ9404 (12 A) should be used. The EJ940x power supply units can be used as additional supply modules in the module string.

Depending on the application, the following combinations for the E-bus supply are available:

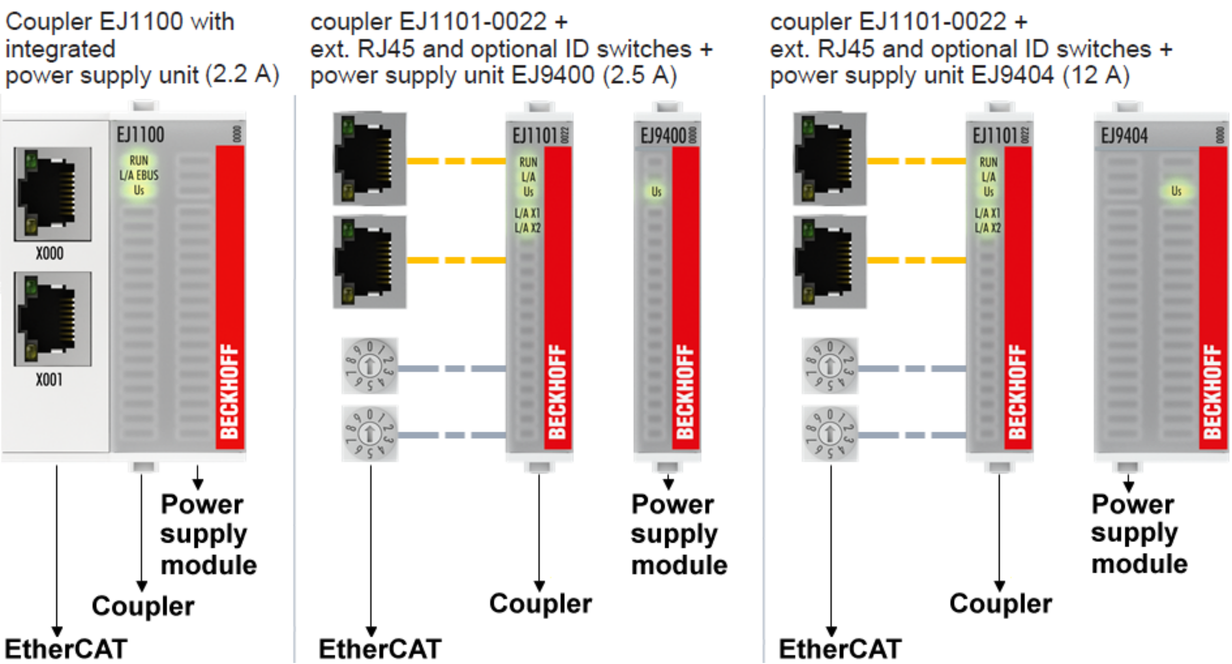


Fig. 12: E-bus power supply with EJ1100 or EJ1101-0022 + EJ940x

In the EJ1101-0022 coupler, the RJ45 connectors and optional ID switches are external and can be positioned anywhere on the signal distribution board, as required. This facilitates feeding through a housing.

The EJ940x power supply plug-in modules provide an optional reset function (see chapter Connection of the documentation for EJ9400 and EJ9404)

E-bus power supply with CXxxxx and EK1110-004x

The Embedded PC supplies the attached EtherCAT Terminals and the EtherCAT EJ coupler

- with a supply voltage Us of 24 V_{DC} (-15 %/+20 %). This voltage supplies the E-bus and the bus terminal electronics.
  The CXxxxx units supply the E-bus with up to 2,000 mA E-bus current. If a higher current is required due to the attached terminals, power feed terminals or power supply plug-in modules must be used for the E-bus supply.
- with a peripheral voltage Up of 24 V_{DC} to supply the field electronics.

The EK1110-004x EtherCAT EJ couplers relay the following parameters to the signal distribution board via the rear connector:

- the E-bus signals,
- the E-bus voltage U_{EBUS} (3.3 V) and
- the peripheral voltage U_P (24 V_{DC}).

Fig. 13: PCB with Embedded PC, EK1110-0043 and EJxxxx, rear view EK1110-0043

5.2 EJxxxx - dimensions

The EJ modules are compact and lightweight thanks to their design. Their volume is approx. 50 % smaller than the volume of the EL terminals. A distinction is made between four different module types, depending on the width and the height:

Module type	Dimensions (W x H x D)	Sample in figure below
Coupler	44 mm x 66 mm x 55 mm	EJ1100 (ej_44_2xrj45_coupler)
Single module	12 mm x 66 mm x 55 mm	EJ1809 (ej_12_16pin_code13)
Double module	24 mm x 66 mm x 55 mm	EJ7342 (ej_24_2x16pin_code18)
Single module (long)	12 mm x 152 mm x 55 mm	EJ1957 (ej_12_2x16pin_extended_code4747)

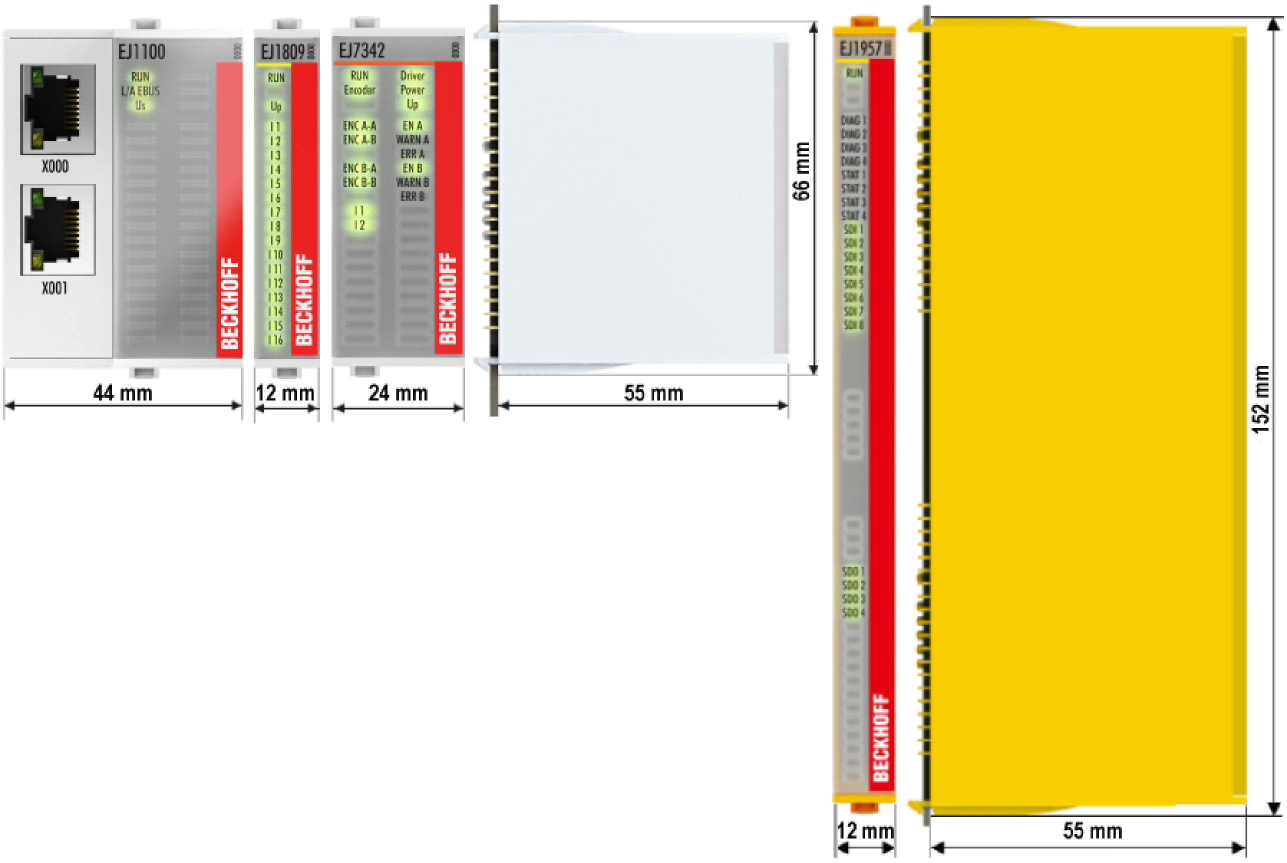


Fig. 14: EJxxxx - Dimensions

The technical drawings can be downloaded from the Beckhoff homepage. The drawings are named as described in the drawing below.

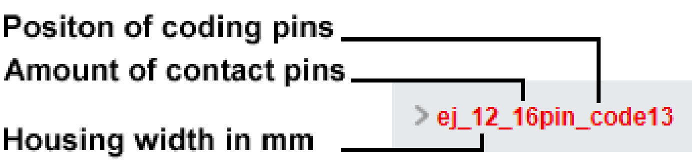


Fig. 15: Naming of the technical drawings

5.3 Installation positions and minimum distances

5.3.1 Minimum distances for ensuring installability

Note the dimensions shown in the following diagram for the design of the signal distribution board to ensure safe latching and simple assembly / disassembly of the modules.

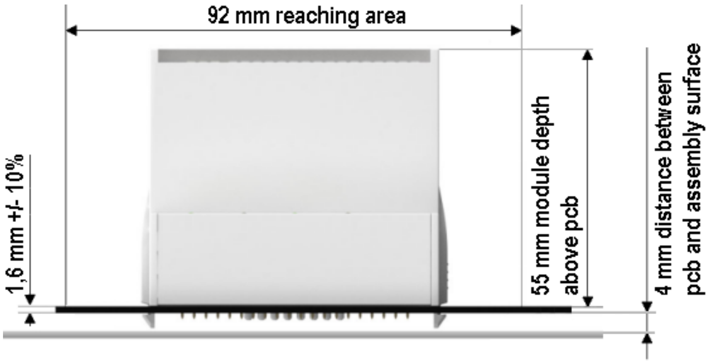


Fig. 16: Mounting distances EJ module - PCB

Observing the reaching area

A minimum reaching area of 92 mm is required for assembly / disassembly, in order to be able to reach the mounting tabs with the fingers.
Adherence to the recommended minimum distances for ventilation (see section Installation position [▶ 29]) ensures an adequate reaching area.

The signal distribution board must have a thickness of 1.6 mm and a minimum distance of 4 mm from the mounting surface, in order to ensure latching of the modules on the board.

5.3.2 Installation positions

NOTE

Constraints regarding installation position and operating temperature range

Please refer to the technical data [▶ 18] for the installed components to ascertain whether any restrictions regarding the mounting position and/or the operating temperature range have been specified. During installation of modules with increased thermal dissipation, ensure adequate distance above and below the modules to other components in order to ensure adequate ventilation of the modules during operation!

The standard installation position is recommended. If a different installation position is used, check whether additional ventilation measures are required.

Ensure that the specified conditions (see Technical data) are adhered to!

Optimum installation position (standard)

For the optimum installation position the signal distribution board is installed horizontally, and the fronts of the EJ modules face forward (see Fig. *Recommended distances for standard installation position*). The modules are ventilated from below, which enables optimum cooling of the electronics through convection. “From below” is relative to the acceleration of gravity.

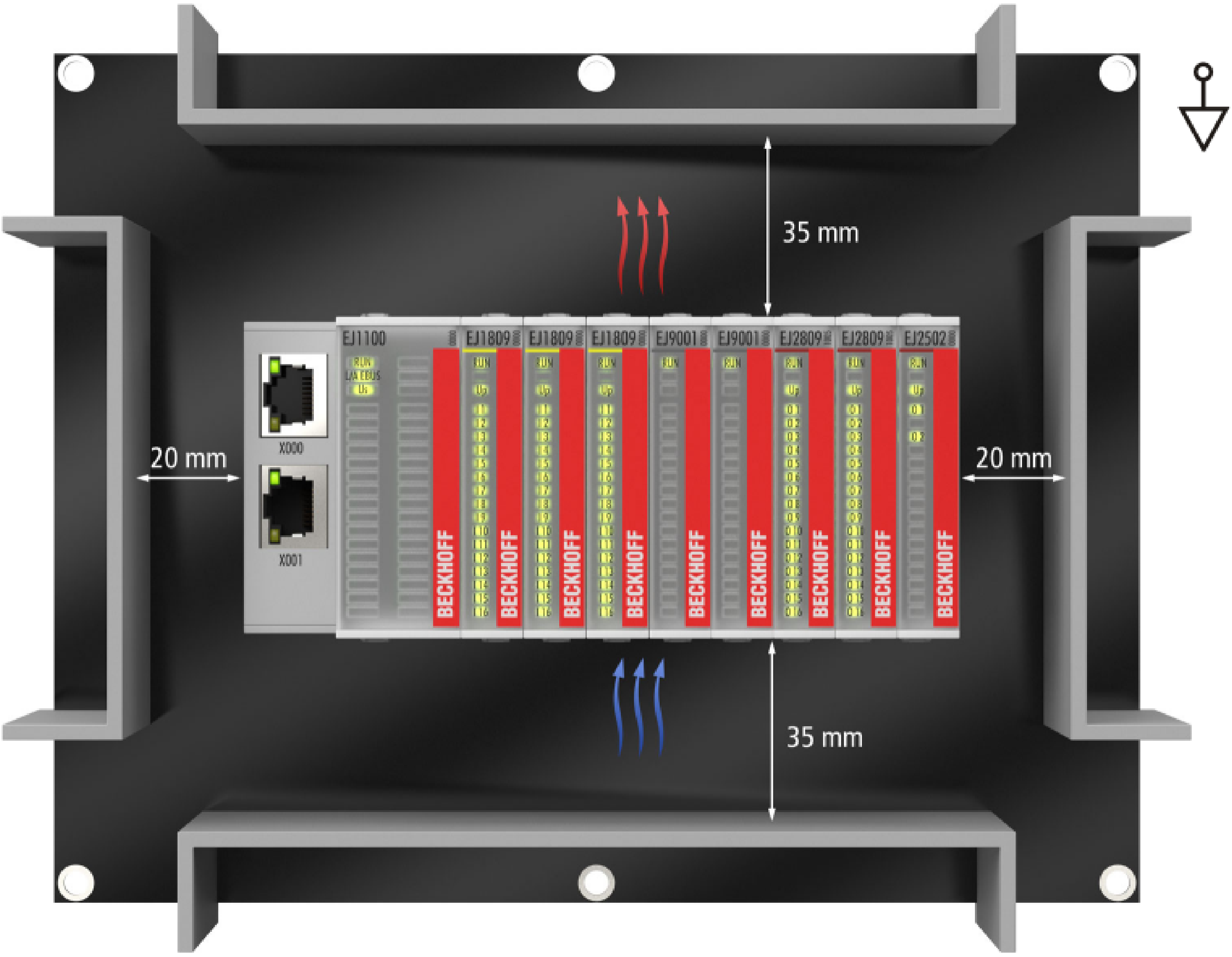


Fig. 17: Recommended distances for standard installation position

Compliance with the distances shown in Fig. *Recommended distances for standard installation position* is recommend. The recommended minimum distances should not be regarded as restricted areas for other components. The customer is responsible for verifying compliance with the environmental conditions described in the technical data. Additional cooling measures must be provided, if required.

Other installation positions

All other installation positions are characterized by a different spatial position of the signal distribution board, see Fig. *Other installation positions.*

The minimum distances to ambient specified above also apply to these installation positions.

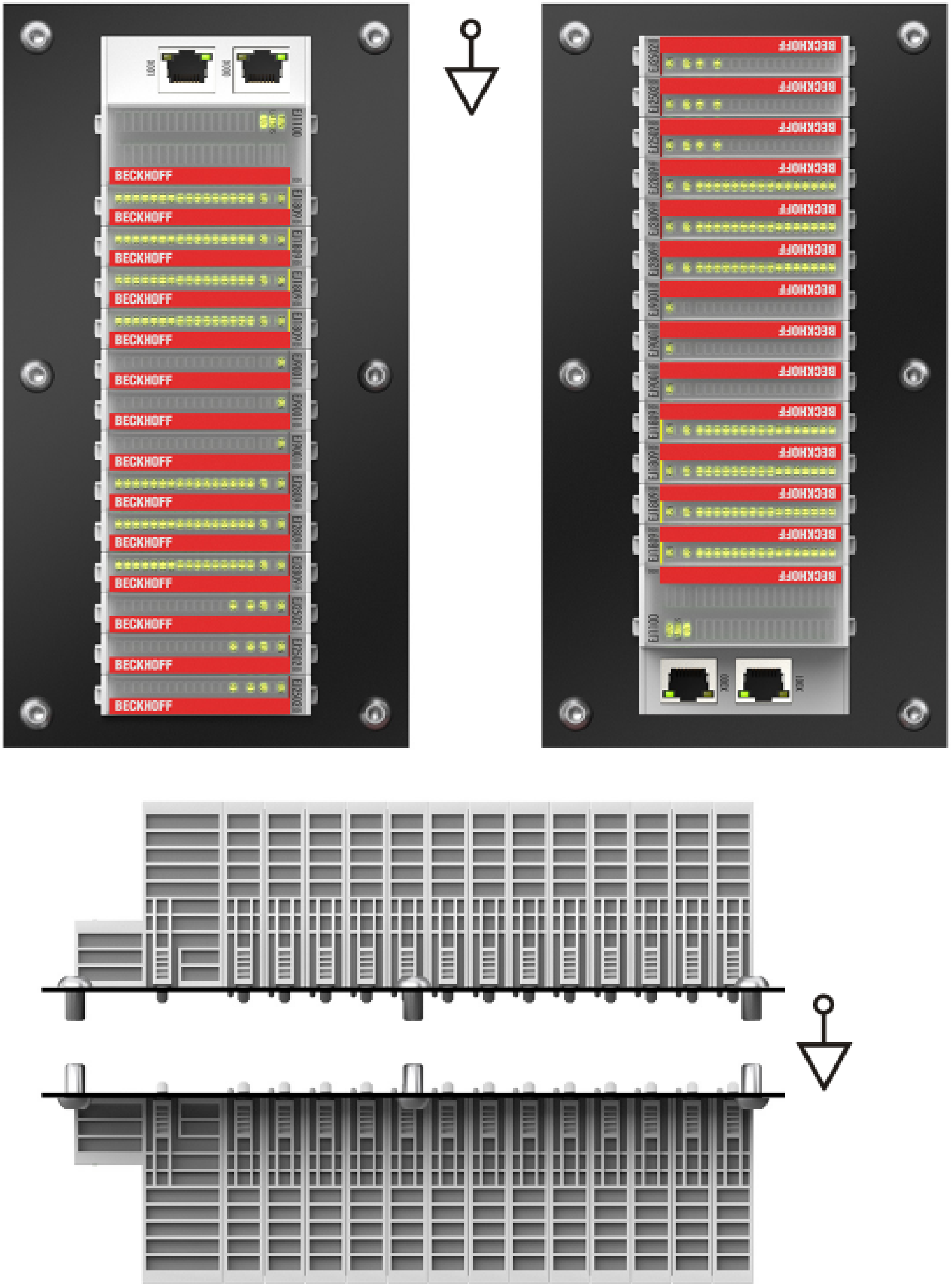


Fig. 18: Other installation positions

5.4 Codings

5.4.1 Color coding

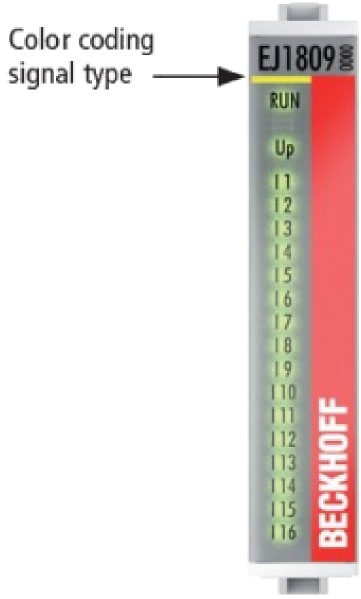


Fig. 19: EJ modules color code; sample: EJ1809

The EJ modules are color-coded for a better overview in the control cabinet (see diagram above). The color code indicates the signal type. The following table provides an overview of the signal types with corresponding color coding.

Signal type	Modules	Color
Coupler	EJ11xx	No color coding
Digital input	EJ1xxx	Yellow
Digital output	EJ2xxx	Red
Analog input	EJ3xxx	Green
Analog output	EJ4xxx	Blue
Position measurement	EJ5xxx	grey
Communication	EJ6xxx	grey
Motion	EJ7xxx	orange
System	EJ9xxx	grey

5.4.2 Mechanical position coding

The modules have two signal-specific coding pins on the underside (see Figs. B1 and B2 below). In conjunction with the coding holes in the signal distribution board (see Figs. A1 and A2 below), the coding pins provide an option for mechanical protection against incorrect connection. This significantly reduces the risk of error during installation and service.
Couplers and placeholder modules have no coding pins.

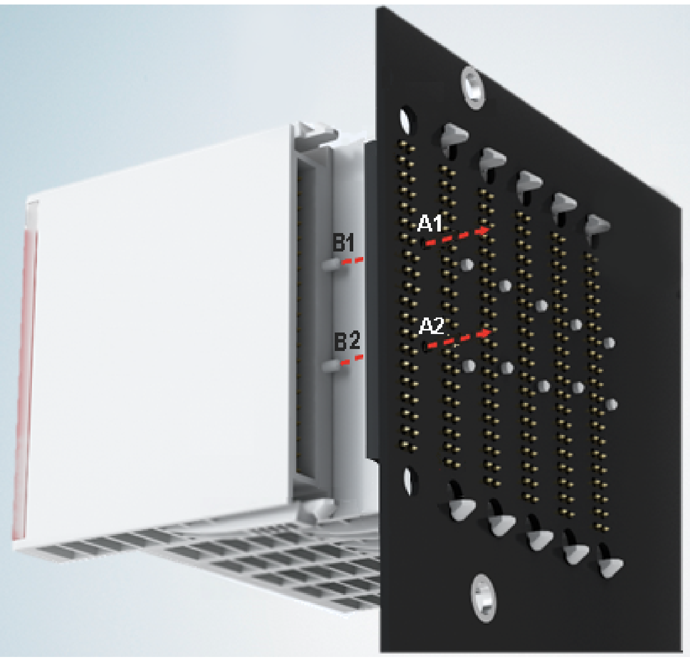


Fig. 20: Mechanical position coding with coding pins (B1 and B2) and coding holes (A1 and A2)

The following diagram shows the position of the position coding with position numbers on the left-hand side. Modules with the same signal type have the same coding. For sample, all digital input modules have the coding pins at positions one and three. There is no plug protection between modules with the same signal type. During installation the module type should therefore be verified based on the device name.

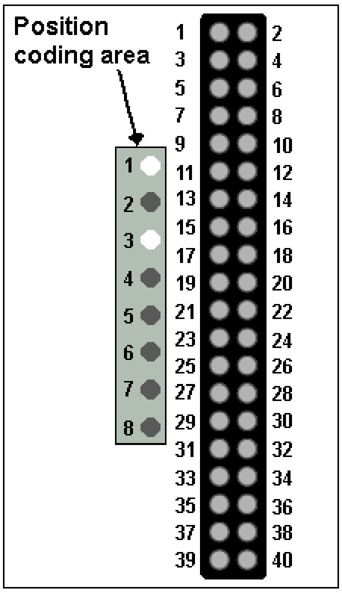


Fig. 21: Pin coding; sample: digital input modules

5.5 Installation on the signal distribution board

EJ modules are installed on the signal distribution board. The electrical connections between coupler and EJ modules are realized via the pin contacts and the signal distribution board.

The EJ components must be installed in a control cabinet or enclosure which must provide protection against fire hazards, environmental conditions and mechanical impact.

⚠ WARNING

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

Bring the module system into a safe, de-energized state before starting installation, disassembly or wiring of the modules.

NOTE

Risk of damage to components through electrostatic discharge!

Observe the regulations for ESD protection.

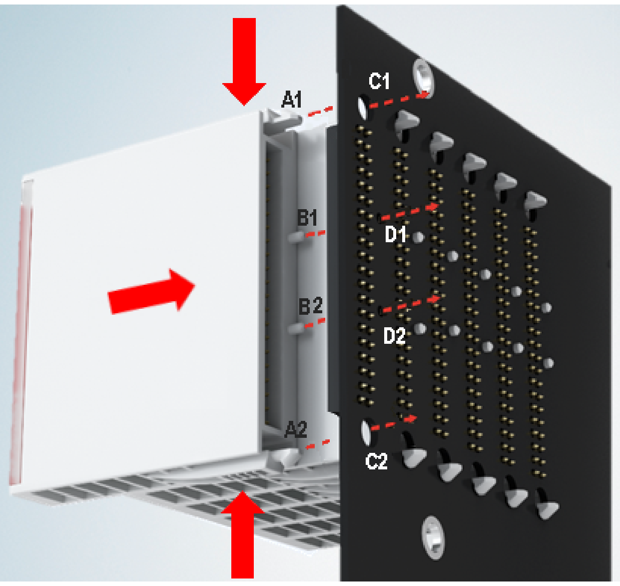


Fig. 22: Installation of EJ modules

A1 / A2	Latching lugs top / bottom	C1 / C2	Mounting holes
B1 / B2	Coding pins	D1 / D2	Coding holes

To install the modules on the signal distribution board proceed as follows:

1. Before the installation, ensure that the signal distribution board is securely connected to the mounting surface. Installation on an unsecured signal distribution board may result in damage to the board.
2. If necessary, check whether the positions of the coding pins (B) match the corresponding holes in the signal distribution board (D).
3. Compare the device name on the module with the information in the installation drawing.
4. Press the upper and the lower mounting tabs simultaneously and push the module onto the board while gently moving it up and down, until the module is latched securely.
   The required contact pressure can only be established and the maximum current carrying capacity ensured if the module is latched securely.
5. Use placeholder modules (EJ9001) to fill gaps in the module strand.

NOTE

- During installation ensure safe latching of the modules on the signal distribution board! The consequences of inadequate contact pressure include:

⇨ loss of quality of the transferred signals,

⇨ increased power dissipation of the contacts,

⇨ impairment of the service life.

5.6 Extension options

Three options are available for modifications and extensions of the EJ system.

- Replacing the placeholder modules with the function modules provided for the respective slot
- Assigning function modules specified for the respective slots for the reserve slots at the end of the module string
- Linking with EtherCAT Terminals and EtherCAT Box modules via an Ethernet/EtherCAT connection

5.6.1 Using placeholder modules for unused slots

The EJ9001 placeholder modules are used to close temporary gaps in the module strands (see Fig. A1 below). Gaps in the module strand cause interruption in EtherCAT communication and must be equipped with placeholder modules.
In contrast to the passive terminals of the EL series, the placeholder modules actively participate in the data exchange. Several placeholder modules can therefore be connected in series, without impairing the data exchange.
Unused slots at the end of the module strand can be left as reserve slots (see Fig. B1 below).

The machine complexity is extended (extended version) by allocating unused slots (see Figs. A2 below - Exchanging placeholder modules and B2 - Assigning reserve slots) according to the specifications for the signal distribution board.

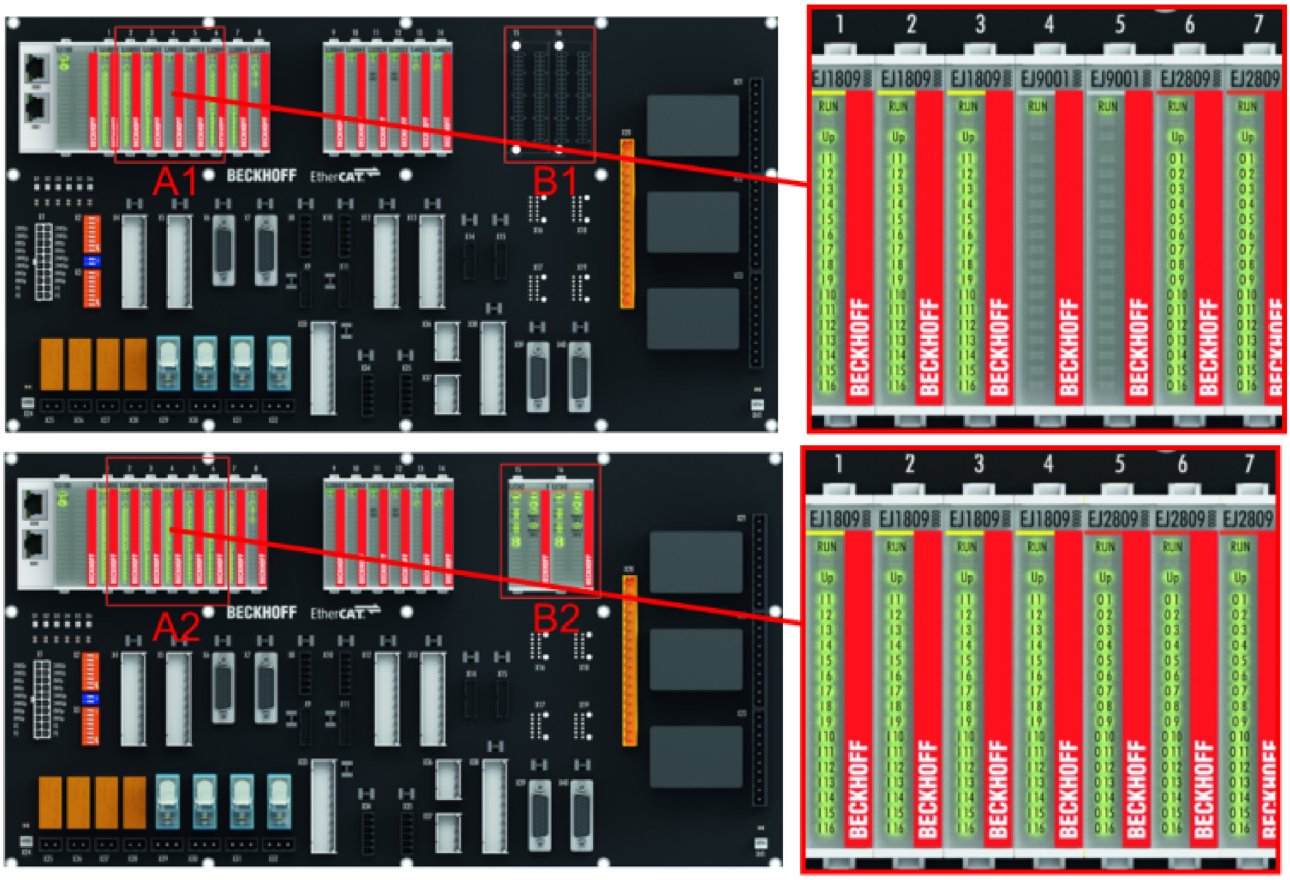


Fig. 23: Sample: Exchanging placeholder modules and assigning reserve slots

E-bus supply

Exchange the placeholder modules with other modules changes the current input from the E-Bus. Ensure that adequate power supply is provided.

5.6.2 Linking with EtherCAT Terminals and EtherCAT Box modules via an Ethernet/EtherCAT connection

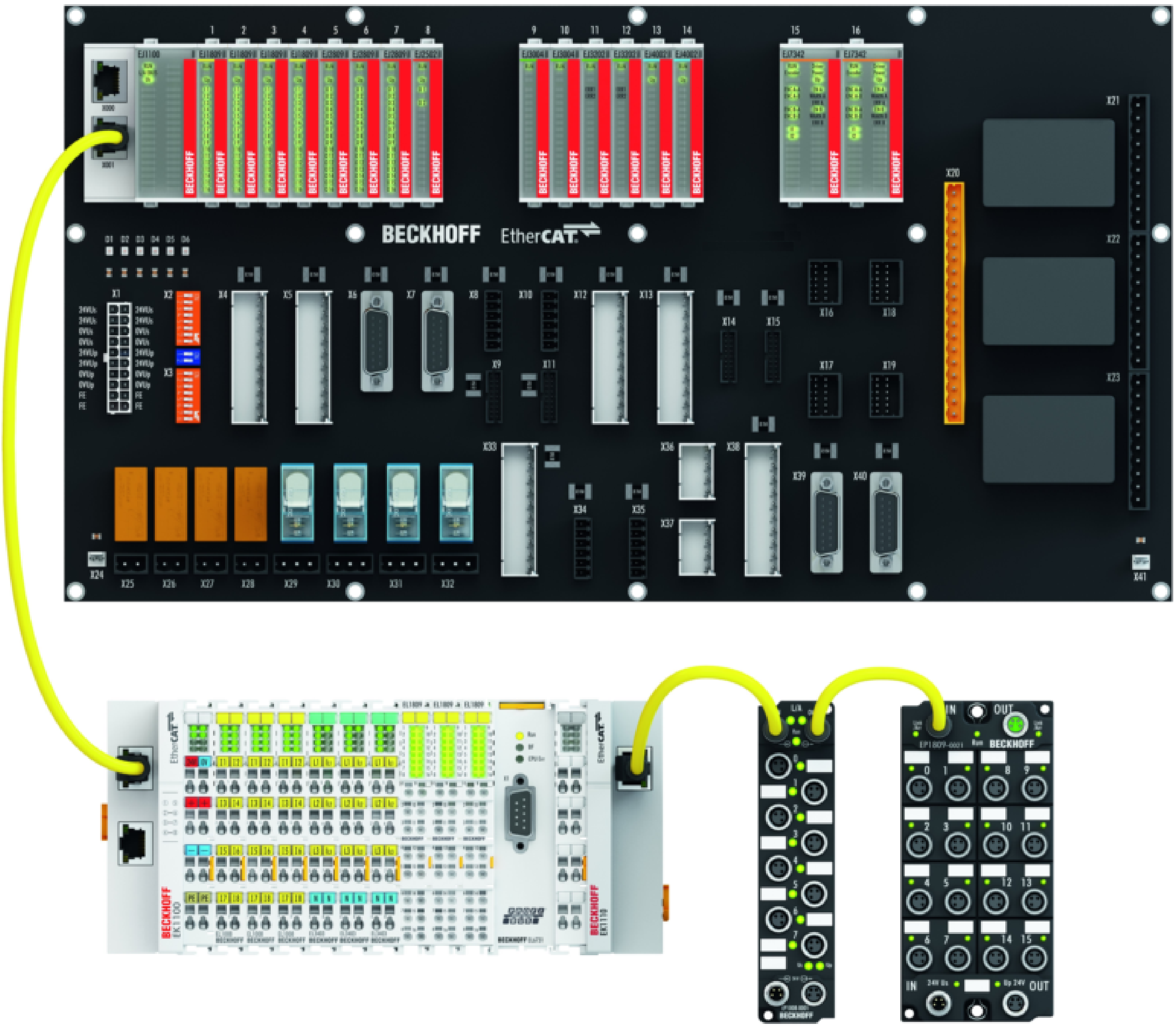


Fig. 24: Example of extension via an Ethernet/EtherCAT connection

5.7 IPC integration

Connection of CX and EL terminals via the EK1110-004x EtherCAT EJ coupler

The EK1110-0043 and EK1110-0044 EtherCAT EJ couplers connect the compact DIN-rail PCs of the CX series and attached EtherCAT Terminals (ELxxxx) with the EJ modules on the signal distribution board.

The EK1110-004x are supplied from the power supply unit of the Embedded PC.
The E-bus signals and the supply voltage of the field side U_P are routed directly to the PCB via a plug connector at the rear of the EtherCAT EJ couplers.

Due to the direct coupling of the Embedded PC and the EL terminals with the EJ modules on the PCB, no EtherCAT Extension (EK1110) or EtherCAT Coupler (EJ1100) is required.

The Embedded PC can be expanded with EtherCAT Terminals that are not yet available in the EJ system, for example.

Fig. 25: Example PCB with Embedded PC, EK1110-0043 and EJxxxx, rear view EK1110-0043

Connection of C6015 / C6017 via the EJ110x-00xx EtherCAT Coupler

Thanks to their ultra-compact design and versatile mounting options, the C6015 and C6017 IPCs are ideally suited for connection to an EJ system.

In combination with the ZS5000-0003 mounting set, it is possible to place the C6015 and C6017 IPCs compactly on the signal distribution board.
The EJ system is optimally connected to the IPC via the corresponding EtherCAT Cable (see following Fig. [A]).
The IPC can be supplied directly via the signal distribution board using the enclosed power plug (see Fig. [B] below).

NOTE	
	Positioning on the signal distribution board The dimensions and distances for placement and other details can be found in the Design Guide and the documentation for the individual components.

The figure below shows the connection of a C6015 IPC to an EJ system as an example. The components shown are schematic, to illustrate the functionality.

Fig. 26: Example for the connection of a C6015 IPC to an EJ system

5.8 Disassembly of the signal distribution board

⚠ WARNING

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

Bring the module system into a safe, de-energized state before starting installation, disassembly or wiring of the modules.

Each module is secured through latching on the distribution board, which has to be released for disassembly.

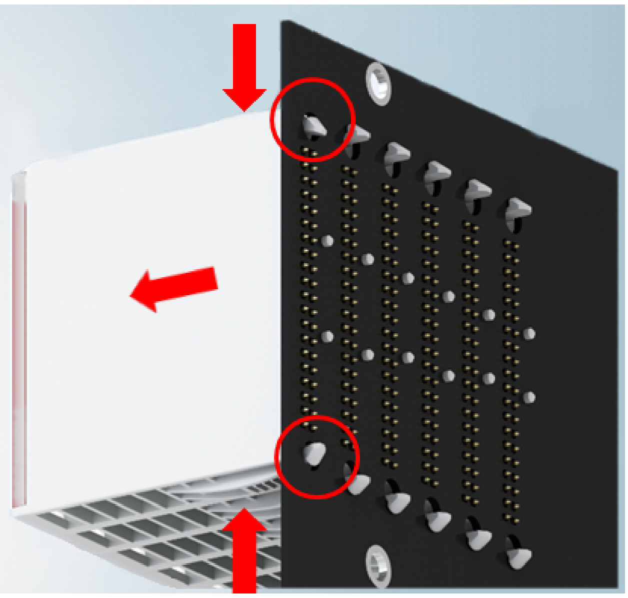


Fig. 27: Disassembly of EJ modules

To disassemble the module from the signal distribution board proceed as follows:

1. Before disassembly, ensure that the signal distribution board is securely connected to the mounting surface. Disassembly of an unsecured signal distribution board may result in damage to the board.
2. Press the upper and lower mounting tabs simultaneously and pull the module from board while gently moving it up and down.

5.9 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.

6 EtherCAT basics

Please refer to the EtherCAT System Documentation for the EtherCAT fieldbus basics.

7 IO link - Configuration and parameterizing

7.1 Configuration of the IO link master

EtherCAT XML device description and configuration files

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 (see chapter Integrating into a TwinCAT project) in the TwinCAT System Manager, an additional tab called "IO-Link" is created (fig. *IO-Link tab*). A detailed description can be found in chapter Configuration of the IO-Link devices [▸ 42]

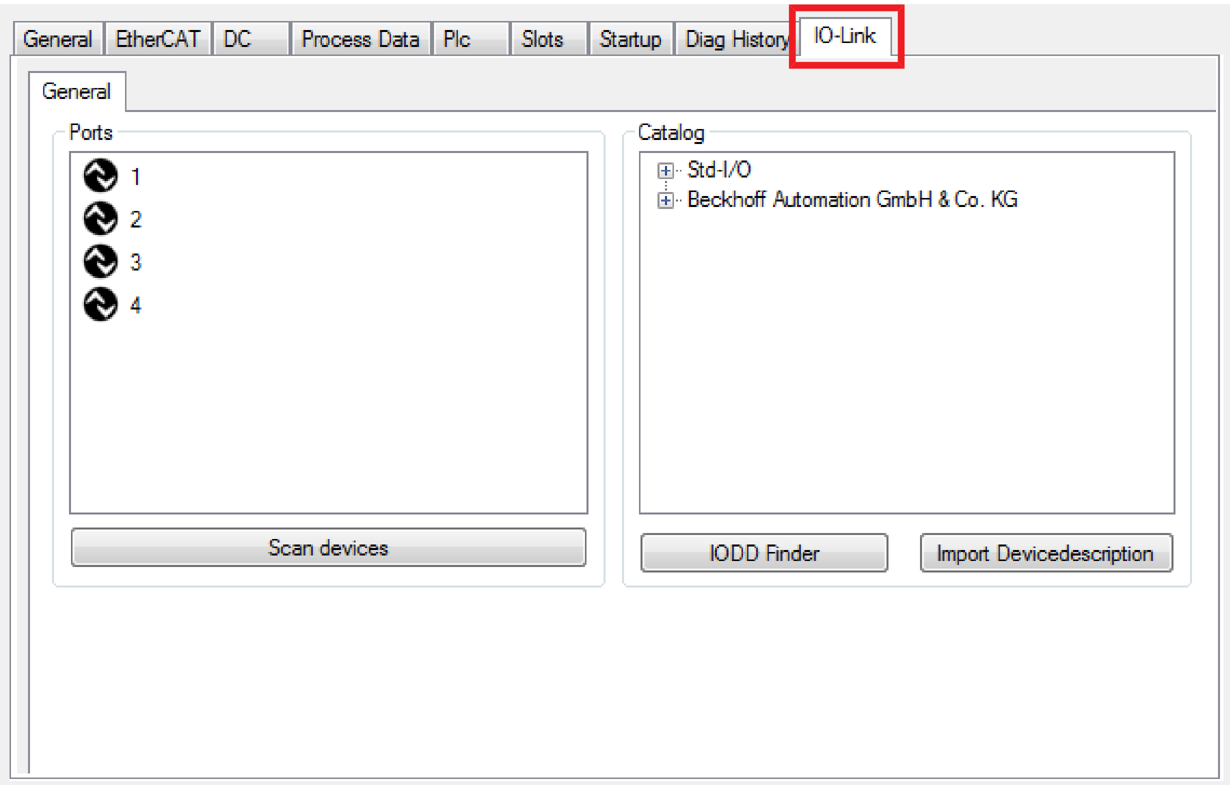


Fig. 28: "IO Link" tab

IO-Link Extension

If the tab “IO-Link” is not displayed, the associated 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 TwinCAT3 does not yet provide this support, it can be installed later if necessary. Please contact our Support [▸ 109].

7.2 Configuration of the 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.

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

1. Open the IO link configuration tool [▸ 42].
2. Import the IODD file of the IO link device [▸ 46].
3. Assign devices to ports.
   ⇨ Assign a device to a port [▸ 43].
   ⇨ Configure a port as digital in- or output [▸ 46].
4. Remove a device from a port [▸ 52].
5. Activate the IO link configuration [▸ 53], so that changes become effective.

7.2.1 Open the IO link configuration tool

✓ Requirement: 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. The configuration tool contains two fields:

- „Ports“
  The left-hand field "Ports" shows a list of the ports of the IO-Link master. If a device has been assigned to a port, the device designation is shown next to the port.
- „Catalog“
  The right-hand field "Catalog" shows the device catalog.
  The device catalog contains an alphabetically sorted list of the IO-Link devices for which a device description (IODD) exists in the local TwinCAT installation.
  The IODDs for the EPIxxxx, ERIxxxx IO-Link Box modules from Beckhoff can be downloaded via the Download finder. The downloaded zip file contains the IODD device description files for the Beckhoff EPIxxxx, ERIxxxx IO-Link Box modules.

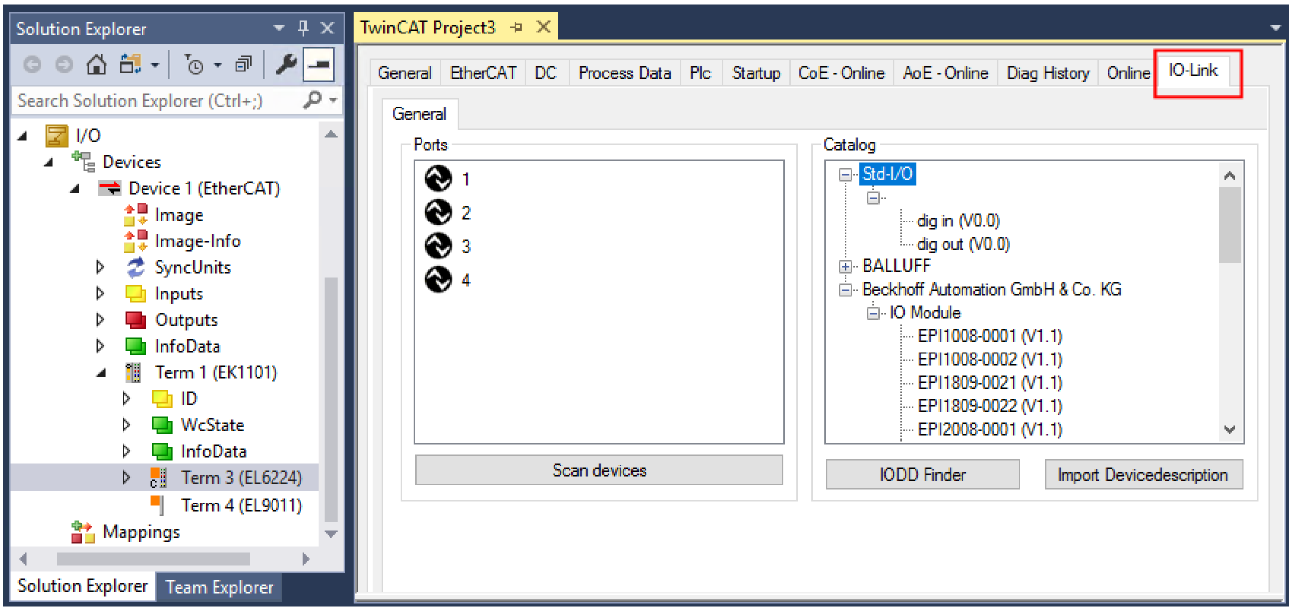


Fig. 29: IO-Link configuration tool

7.2.2 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 devices as well as the display of the parameters.

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

1. Importing the IODD file (offline and online) via
   ⇨ button Import Devicedescription [▶ 44] (A) or
   ⇨ button IODD Finder [▶ 44] (B)
2. Select the device in the "Catalog" field and assign it to a port [▶ 46].
3. Automatic scanning of the IO-Link ports (online) via
   ⇨ button Scan devices [▶ 47] (C)
4. Manual insertion (offline and online) via
   ⇨ menu Create Device [▶ 51] (D)

Application note

- If the IODD is not available,
  the IO-Link device should be integrated online by scanning.
- Manual integration of the IO-Link devices via "Create Device" should only be carried out if the IODD of the vendor and the IO-Link device are not available at the time of project creation.

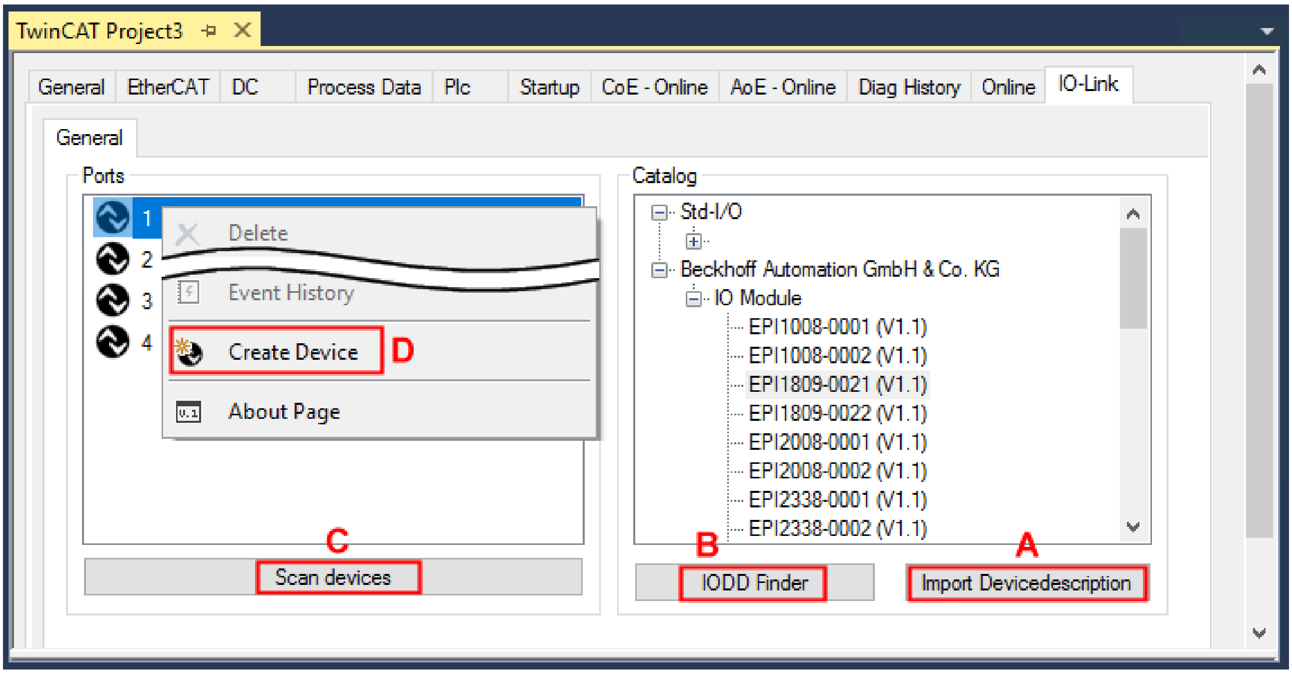


Fig. 30: Creating IO-Link devices

7.2.2.1 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:

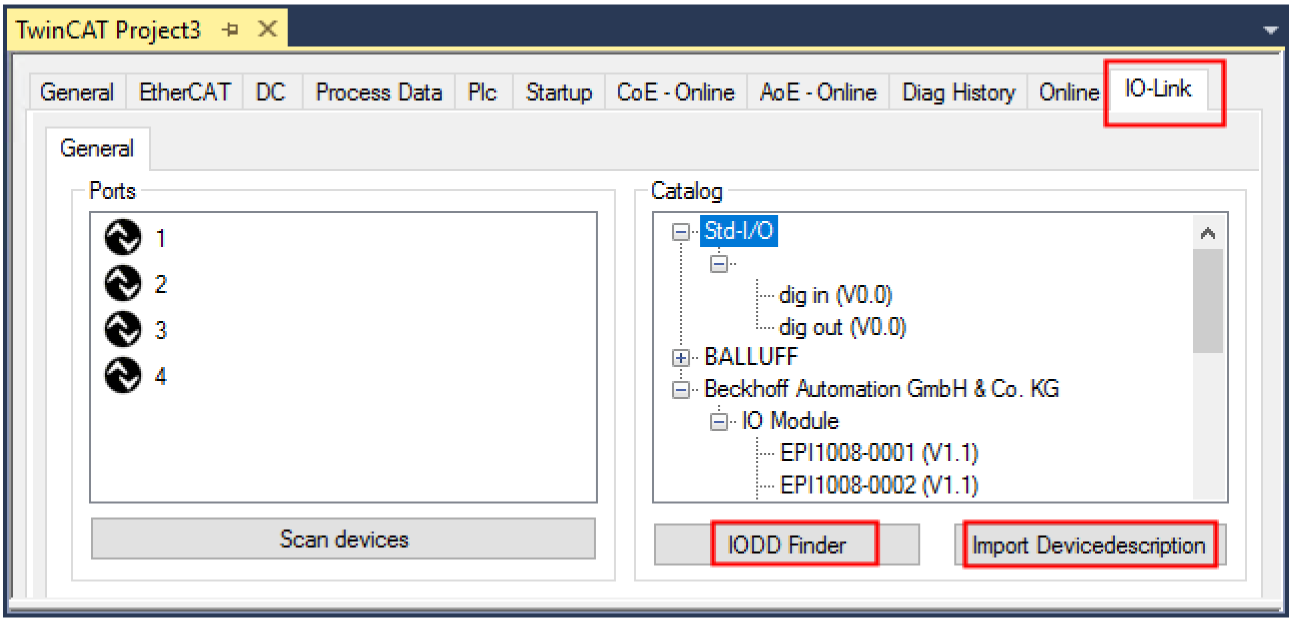


Fig. 31: Import of the IODD device description via “IODD Finder” or “Import Devicedescription”

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!

Button "IODD Finder“

1. Press the “IODD Finder” button in the “IO-Link” tab
2. Searching for the desired IO-Link sensor/device by entering them in the search mask; see the figure below (1)
3. Selecting 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 the following figure (2).

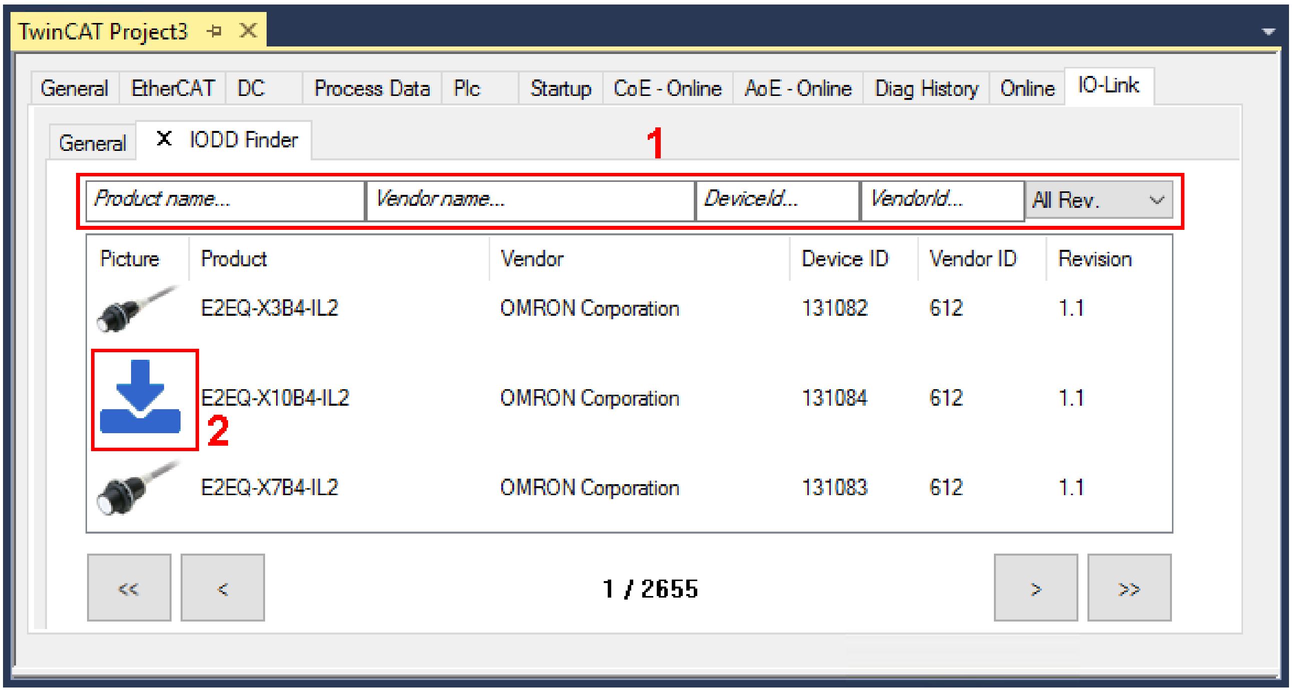


Fig. 32: IODD Finder, selection and import of the .xml-file

4. After clicking the download symbol, 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 (3)) that the .xml file already exists.

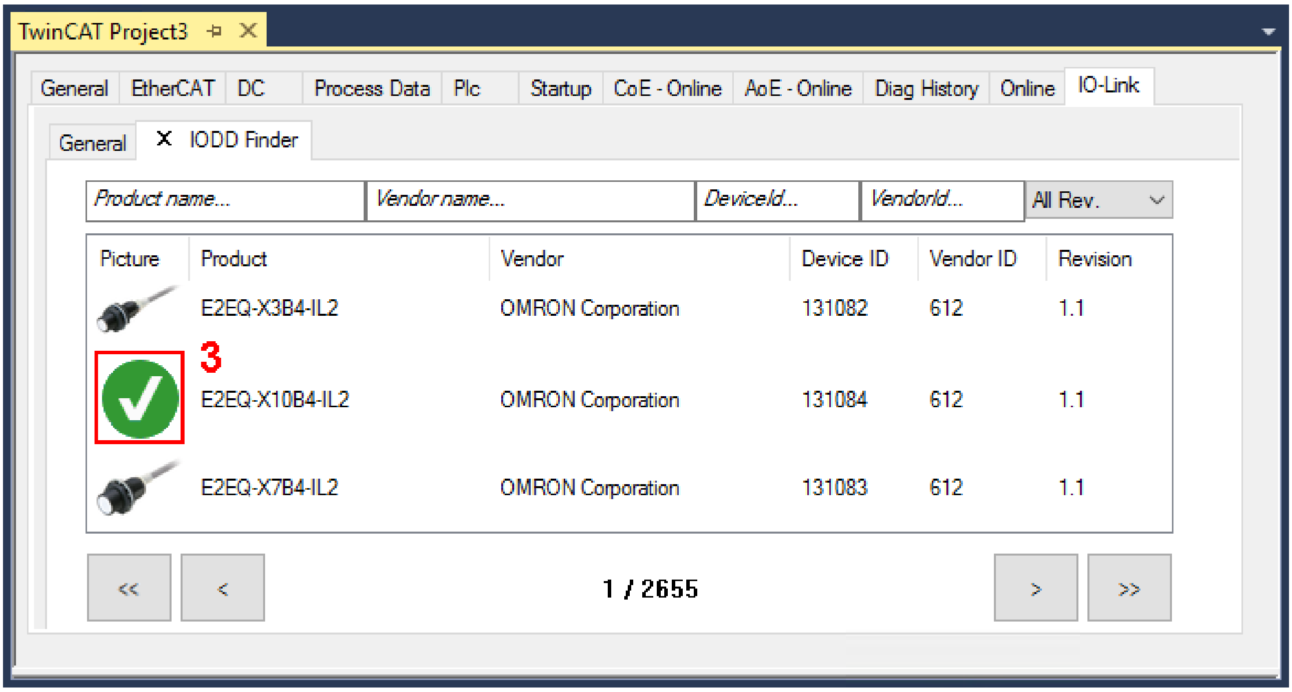


Fig. 33: IODD Finder, display of an already imported device description

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

7.2.2.2 2. Assigning IO-Link device to port n

Online configuration

✓ Requirement: The IO-Link device is connected.

1. Press the button Scan devices (see chapter Automatic scanning [▶ 47])

⇨ 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 are the same and there are only mechanical differences (e.g. other material).

Offline configuration

The *Catalog* field shows the IO-Link device catalog, which lists the already imported device descriptions in a tree structure, sorted by vendor.

1. Select the desired IO-Link device from the *Catalog* field
   - via drag and drop or
   - by right-clicking on the product with "Add to Port n".

Activating the configuration

2. Activate the IO link configuration [▶ 53], so that changes become effective.

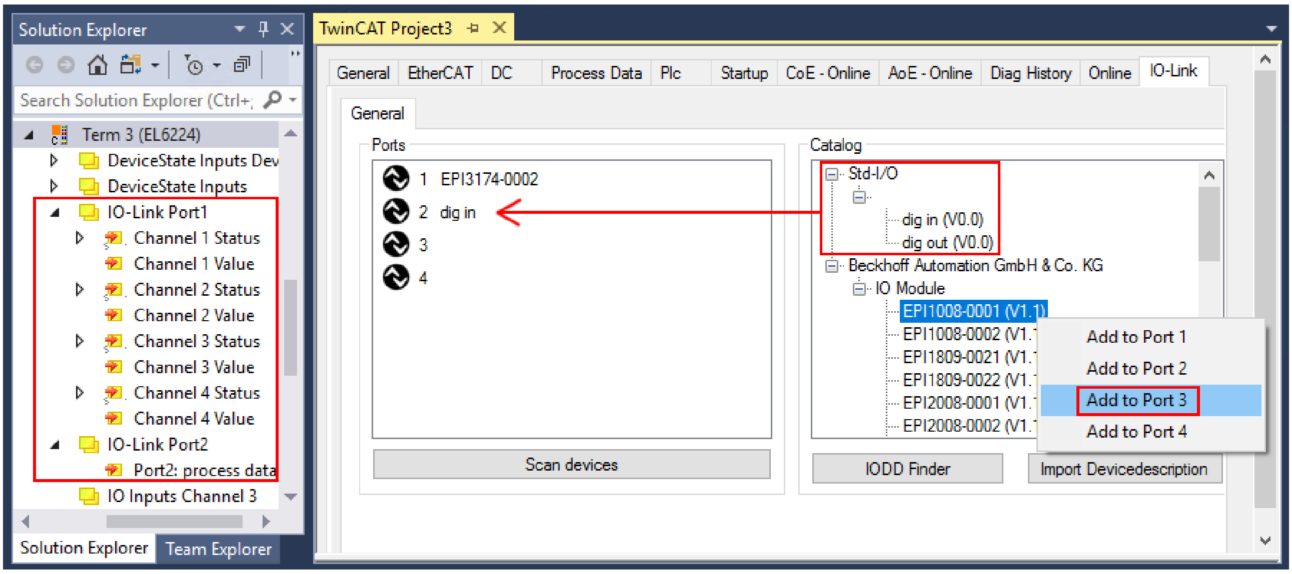
⇨ The IO-Link devices are displayed, and the process data are created. If an error is found when integrating the IO-Link device, e.g. wrong VendorID or no device connected, then this is indicated via the status of the port (object state Ch.n 0xF100:0n).

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. Expand the "Std-I/O" tree node in the "Catalog" field.
   ⇨ The operating modes "dig in" and "dig out" appear.
2. Configure the desired port. There are two ways to do this:
   - Drag-and-drop: pull "dig in" or "dig out" onto the port in the "Ports" field or
   - Right-click on "dig in" or "dig out" and click on "Add to Port n“.

Example of port assignment on the IO link master EL6224



Port1:	Port2:	Port3:
EPI3174-0002 is assigned	is configured as digital input	EPI1008-0001 will be assigned

Process data of Port1 and Port2 are displayed in the Solution Explorer.

7.2.2.3 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 [▶ 22]. 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

✓ Requirement: the master and the devices are cabled and supplied with voltage.

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

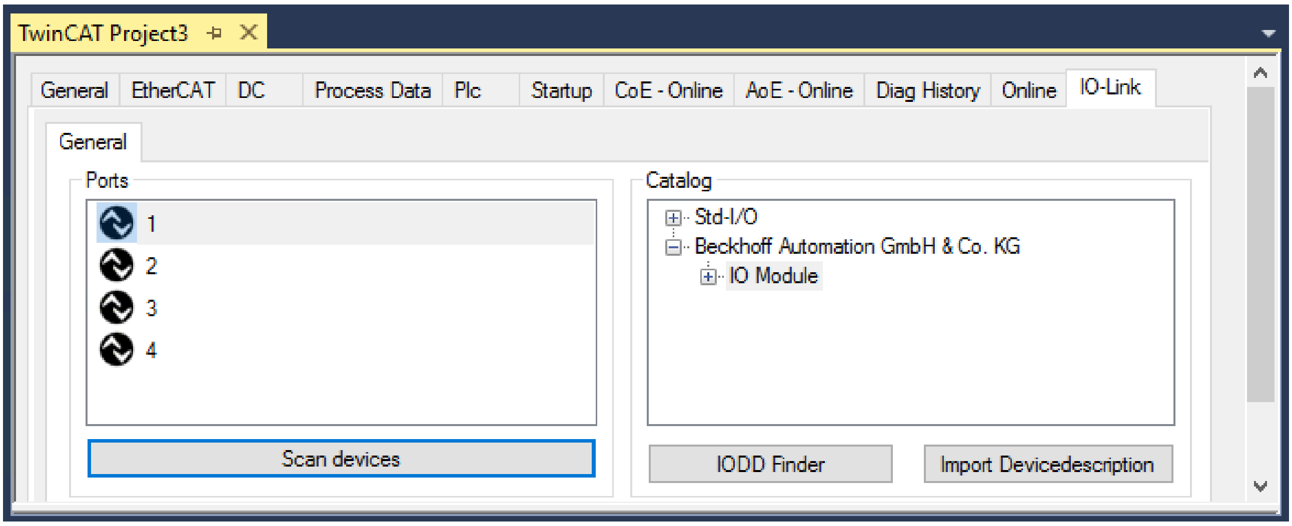


Fig. 34: Scan devices

⇨ The connected IO-Link devices can be found.

⇨ The information window lists the connected device for each of the four ports. Only port2 of the master is assigned an IO-Link device.

⇨ Confirm with the *OK* button.

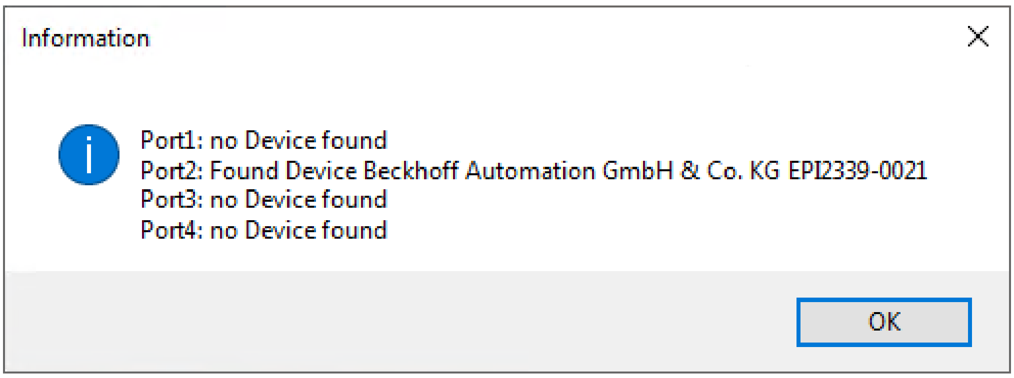


Fig. 35: Information “Scan devices”

2. 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* display. The Port2 “Details” field displays information about the connected device. Additionally the tabs Settings [▶ 49] and Parameter [▶ 50] can be opened.

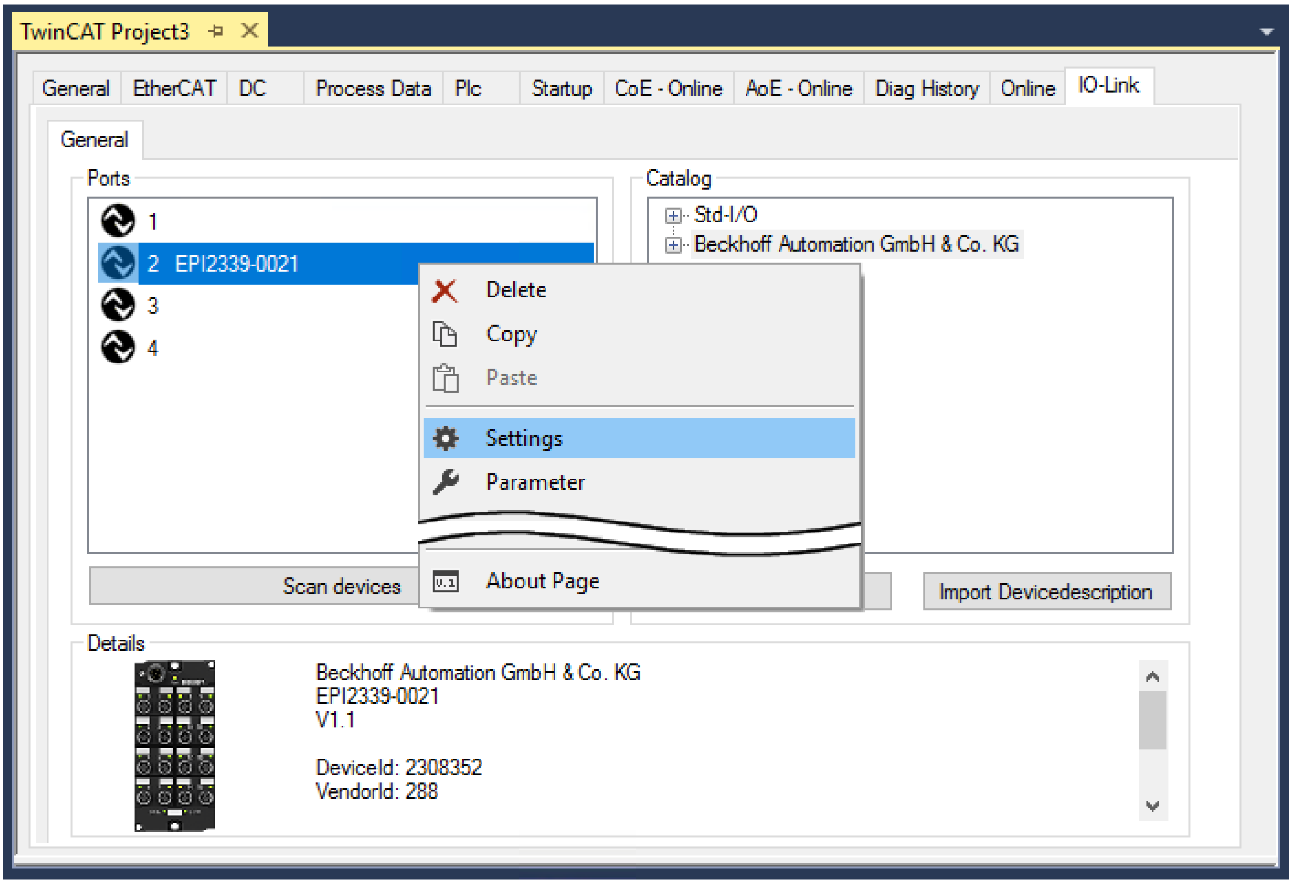


Fig. 36: Device at Port2, Display “Details”, open tabs “Settings” and “Parameter”

Show settings of the device

3. Right-click on port2, to display more details in dialog "Settings".
4. If necessary, change the settings as described in chapter Settings of the IO-Link devices [▶ 54].

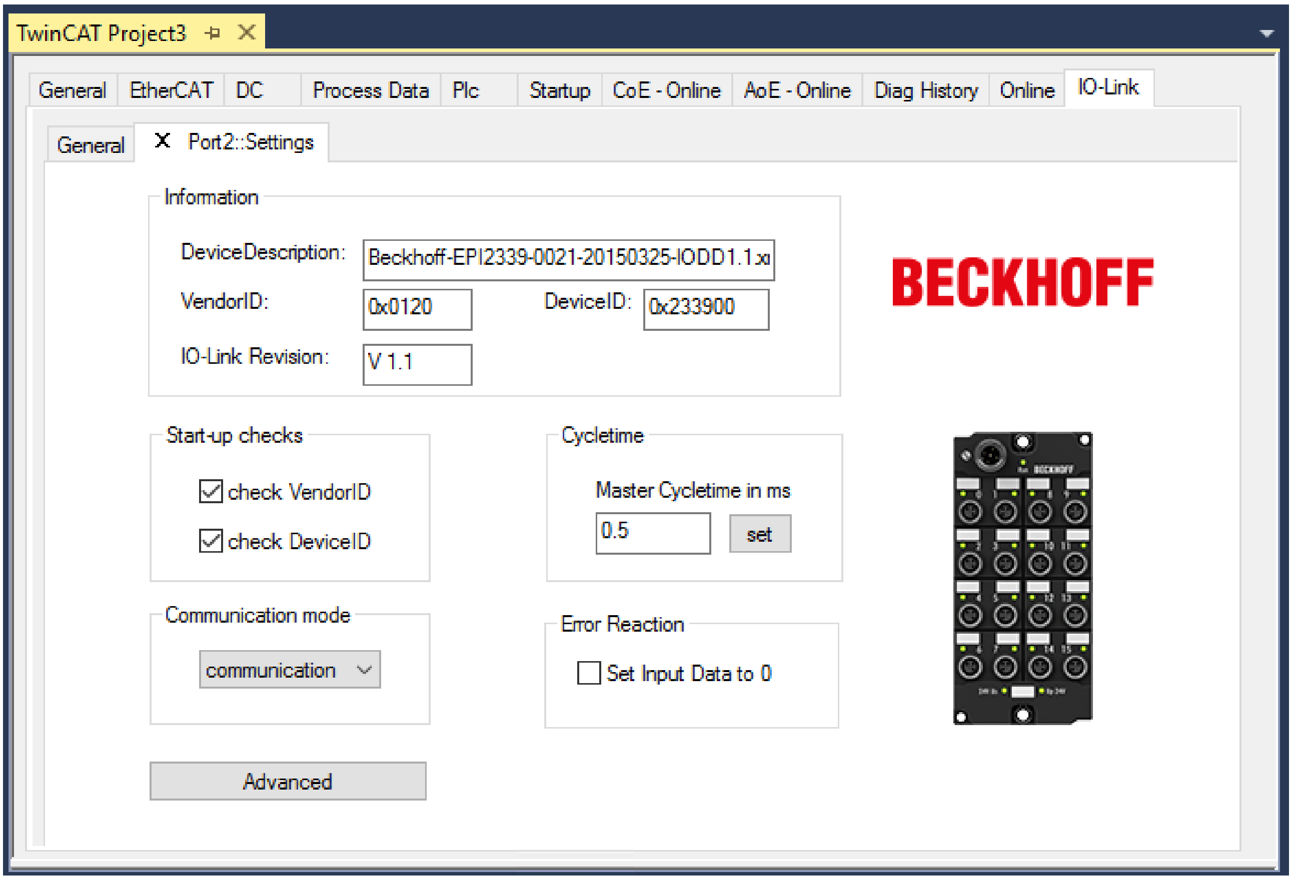


Fig. 37: Settings of the device assigned to port2

Show parameters of the device

5. Open the Parameter tab via
   - double-click on Port2 or
   - richt-click on Port2 and select “Parameter” in the menu.

   ⇨ The Parameters of of the respective IO link device are listed.

6. Parameterize the device as described in chapter EPIxxxx, ERIxxxx - Setting of the IO-Link device parameters [▶ 56].

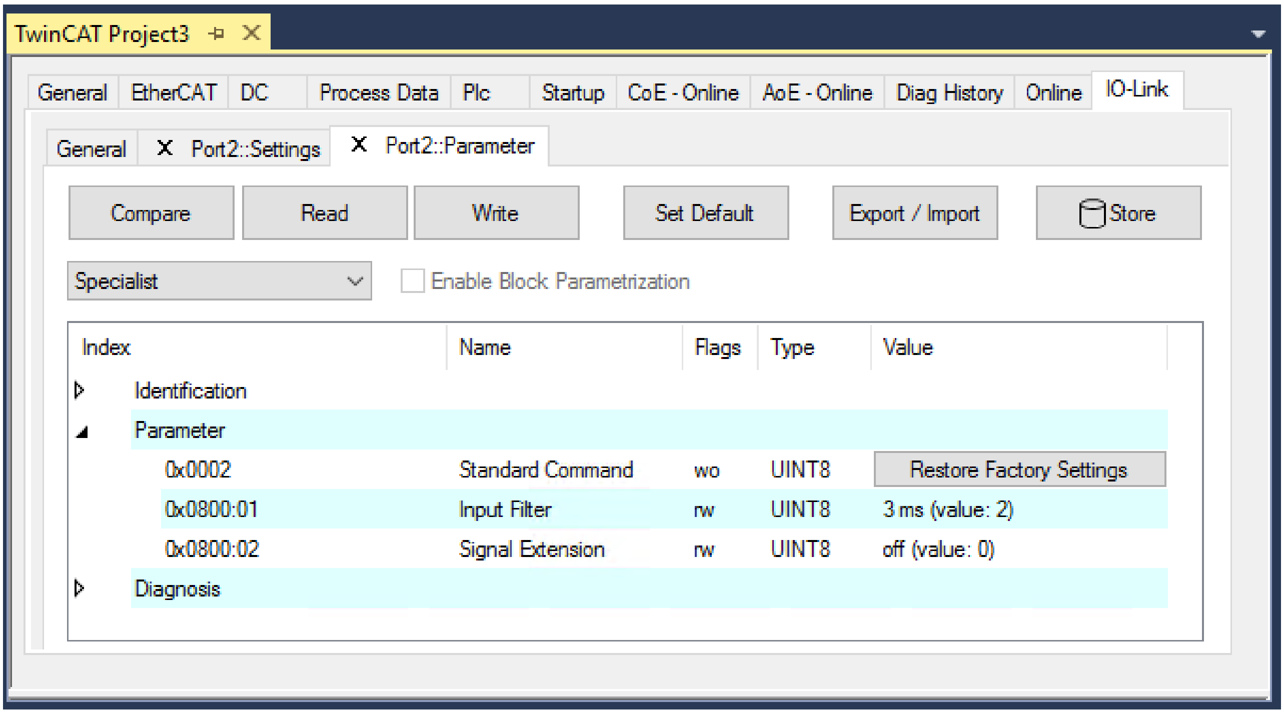


Fig. 38: Parameter of the device assigned to port2

7.2.2.4 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 that were created are **not** stored in the “Catalog” (see the figure below (A)). To insert the IO-Link devices manually via “Create Device”, proceed as follows:

1. The IODD of the IO link device is already available:
   Select the respective device from the “Catalog” field sorted by manufacturer (see following figure (A)).
2. No IODD is available:
   Add the device can be manually via “Create Device”. These data are **not** saved in the “Catalog” field and must be manually entered for each port.
3. Right-click on the port to open the context menu (see the figure below (B)) and select “Create Device”.
4. In the “Create Device” dialog an IO-Link device with the basic communication parameters can be created. The mandatory fields here are: For Vendor ID, Device ID and process data length see the figure below (C). The values VendorID and DeviceID can be entered both in hexadecimal notation (input format: 0xnnnn) and as decimal numbers (nnnn).
   The communication parameters to be entered can be found in the information provided by the device vendor.
5. If the IO-Link device version is 1.1, then the parameter server is activated by the selection of the check box “Revision V1.1” (see following figure (D)).
6. Activate the IO link configuration [▶ 53], so that changes become effective.

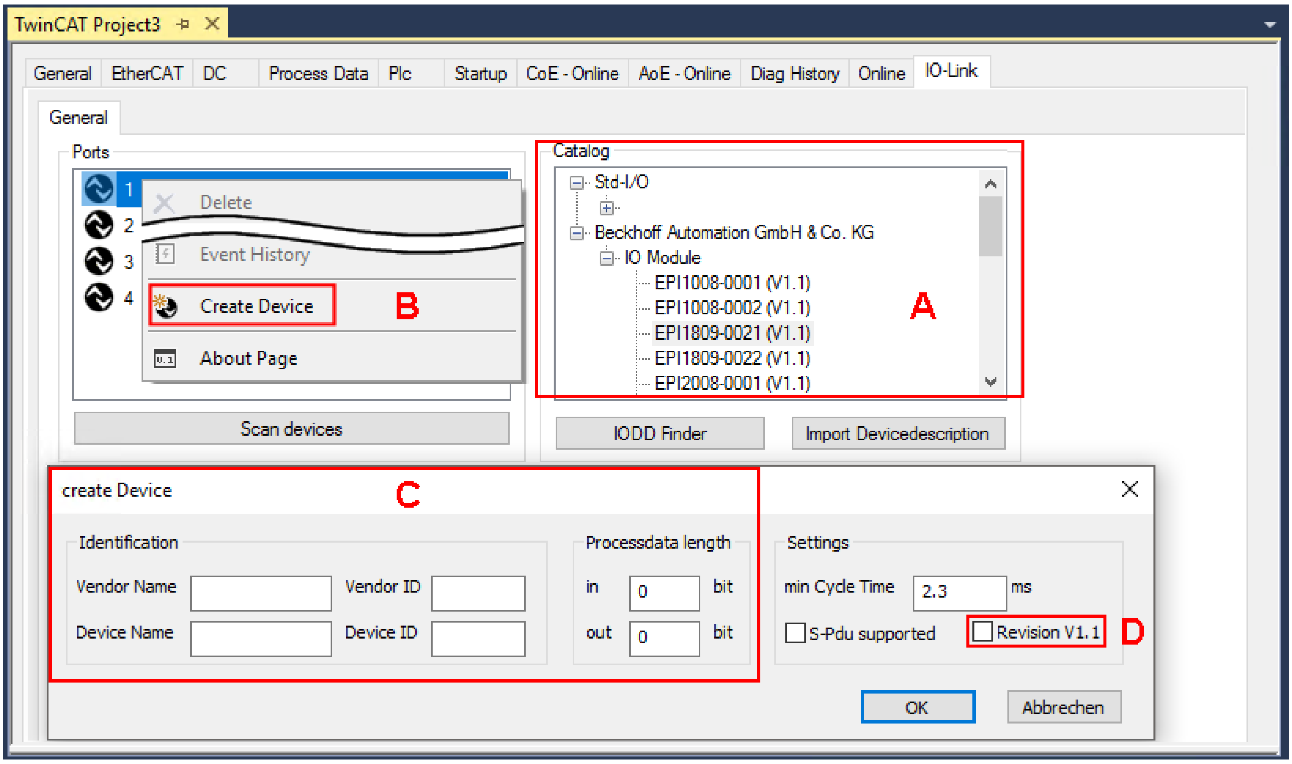


Fig. 39: Manual creation of an IO-Link device via the "Create Device" dialog (C)

Reading the IODD

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

7. In the “Settings” tab of the IO link devices further settings can be made as described in chapter Settings of the IO-Link devices [▶ 54].

7.2.3 Removal of IO-Link devices

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

1. Right-click on the port to open the context menu and select “Delete”.

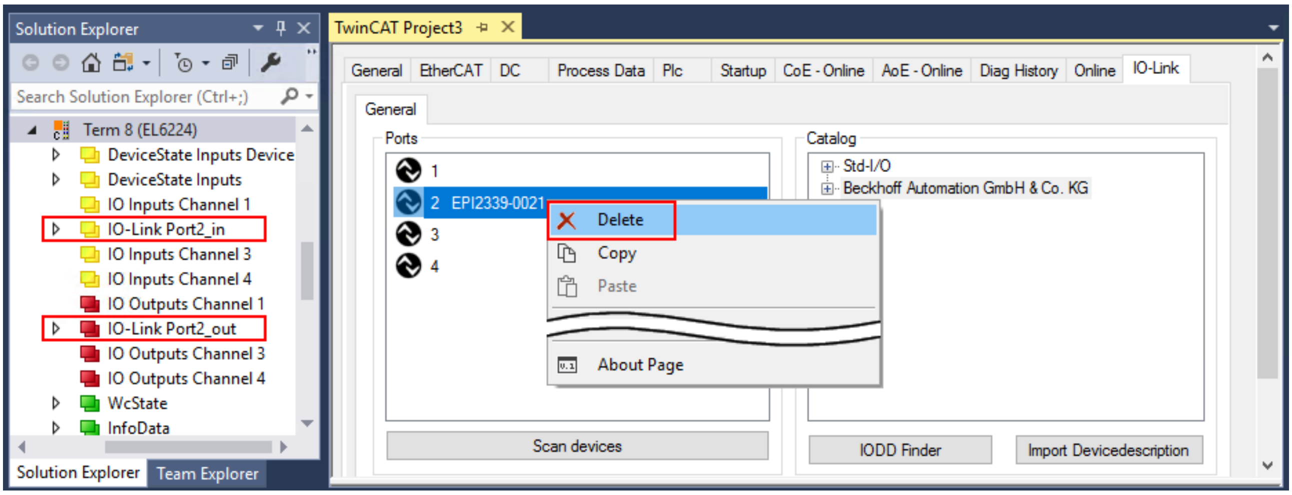


Fig. 40: Remove the device from port2

2. Activate the IO link configuration [▸ 53], so that changes become effective.

⇨ The already create process data are removed.

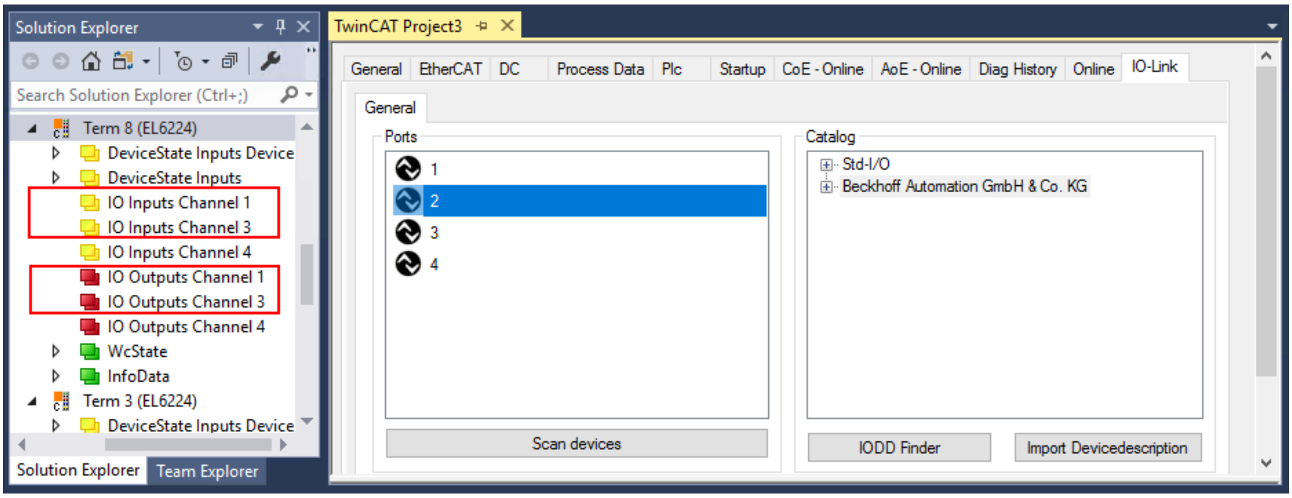


Fig. 41: The device was removed from the port2, the process data no longer displayed in the tree.

7.2.4 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

7.3 Settings of the IO-Link devices

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

1. right-click on the port to open the context menu and select “Settings”.

⇨ A new tab “Portx:: Settings” opens where the settings described below can be made.

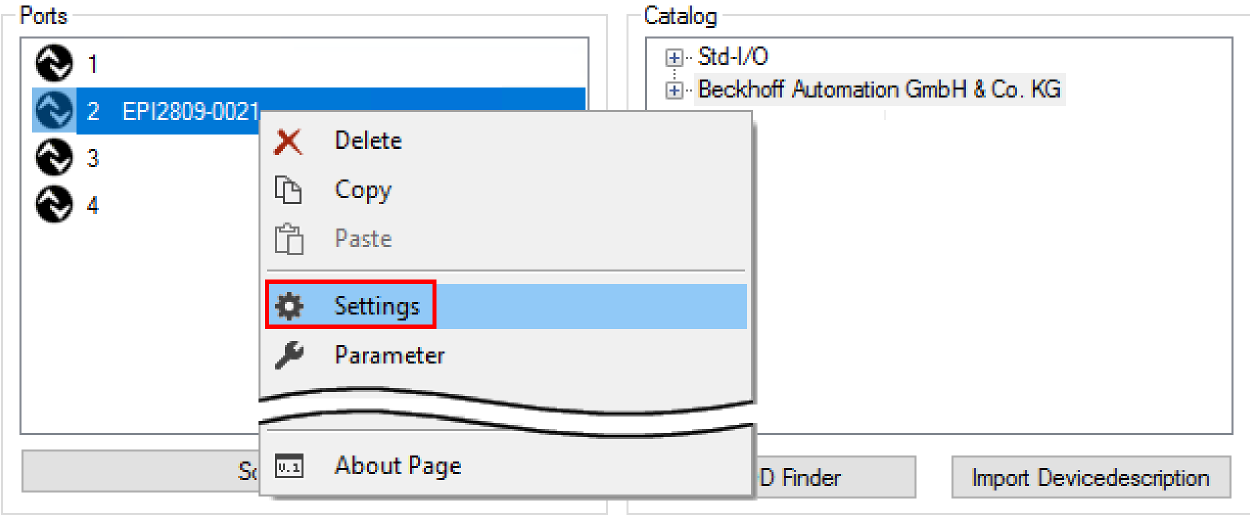


Fig. 42: Context menu - Settings

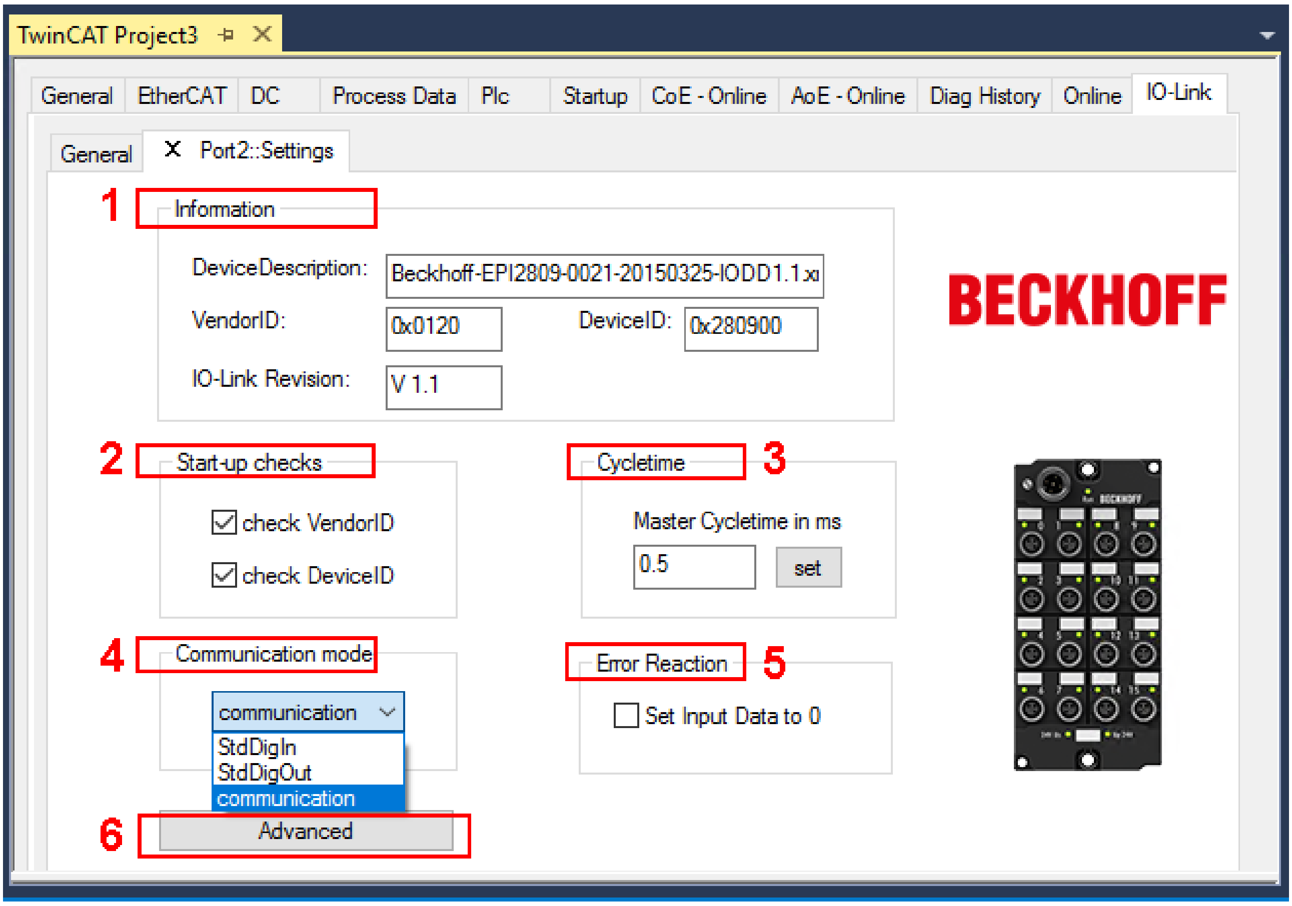
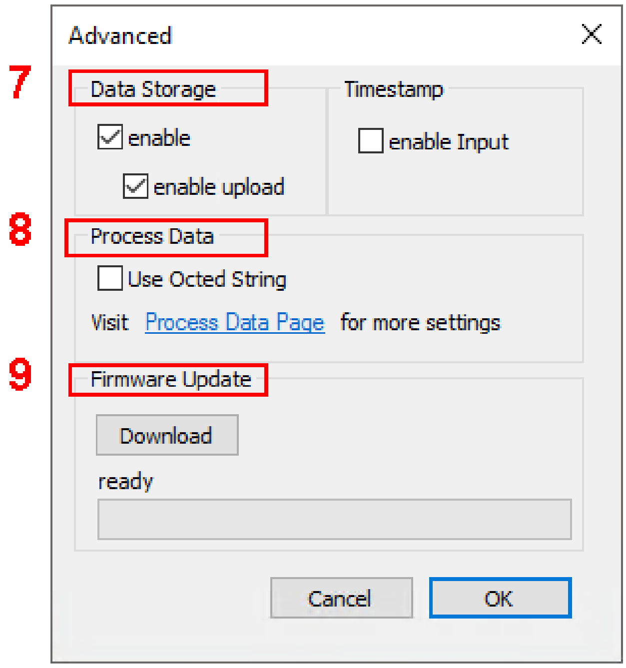


Fig. 43: Settings of the IO-Link devices

1. **Information**
   This field is for information only; the IODD that was read in is displayed under Device Description. Furthermore, the VendorID, DeviceID and the IO-Link revision (V1.0 or V1.1) of the IO-Link devices are displayed. If the device is an IO-Link device V1.1, then the parameter server [▶ 23] function is supported.

The following settings can be made in the settings for the IO-Link devices (see figure above):

2. **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 exchanging IO-Link devices.
3. **CycleTime**
   Specifies the cycle time for the IO-Link master
4. **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 [▶ 46].
5. **Error Reaction**
   If the “Set Input Data to 0” field is activated:
   ⇨ input data are set to 0 in case of error
   ⇨ Status display: “Error”
6. **Button "Advanced"**
7. **Data Storage**
   Pay attention to the sensor version:
   ⇨ V1.0 -> data storage is not supported
   ⇨ V1.1 -> data are stored in the parameter server (preset)
8. **Process Data Format**
   Adaptation of the process data format
   If Field “Use Octet String” is selected
   ⇨ complex data types (process data) are created as octet strings.
   Advantage: simple further processing in the PLC
9. **Firmware Update of the Beckhoff IO-Link devices**
   For a firmware update use the “Download” button. Observe the description in the documentation of EPIxxxx boxes in chapter Firmware Update des IO-Link Devices.



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

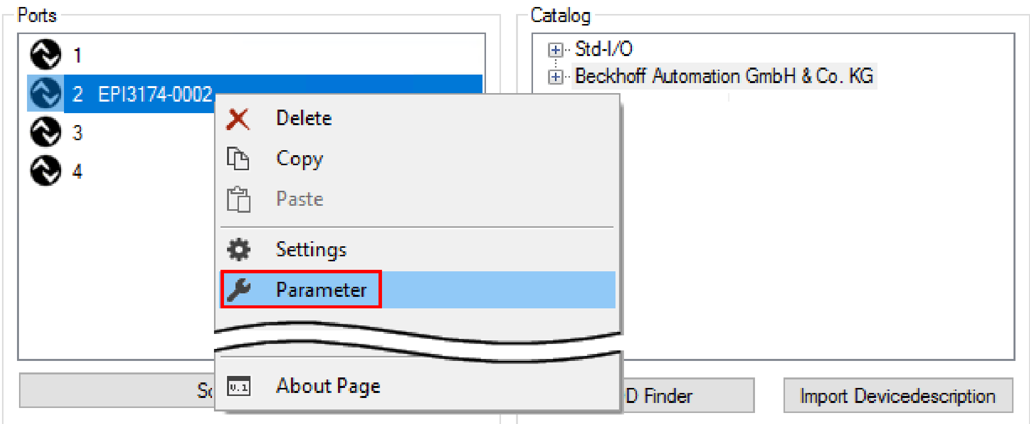
This chapter explains how to read out and set the IO-Link device parameters.

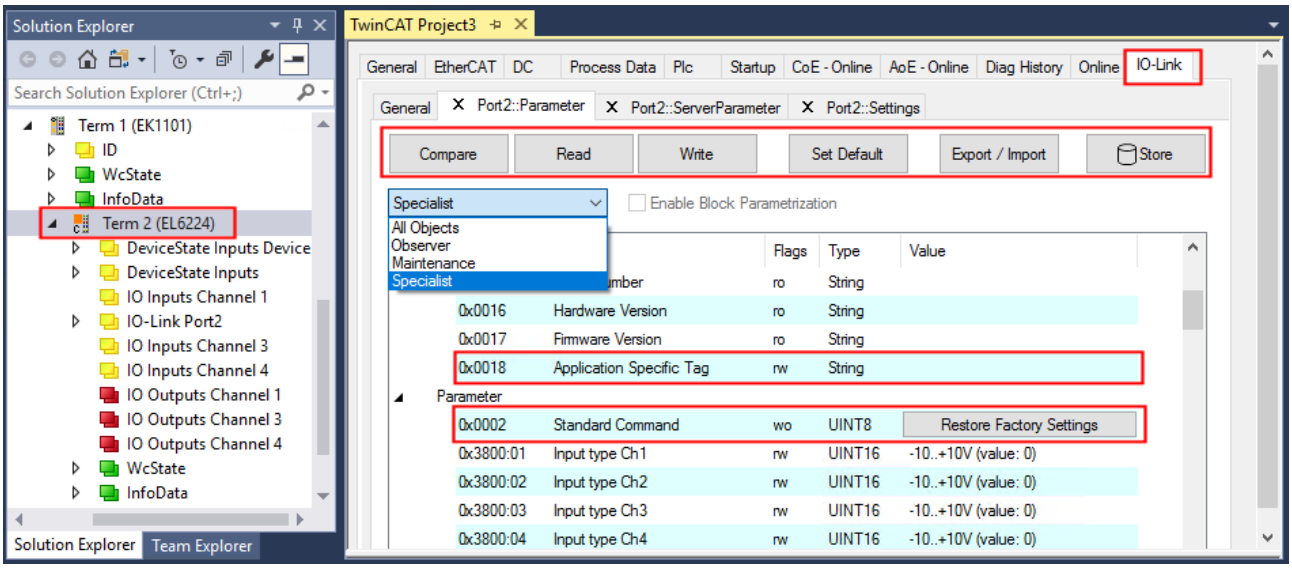
The number and type of the objects shown on the “Parameters” tab vary according to the type of sensor. The default settings as stored in the IODD can initially be seen.

To open the “Parameter” tab

1. Click the IO-Link master in the TwinCAT tree structure.
2. Click the “IO-Link” tab.
3. Select the port to which the IO-Link device is connected,
4. Double-click or by right-click to the port and select “Parameter”.

⇨ The “Parameter” tab is opened.





The device parameters are listed in the tab. The buttons Compare [▶ 57], Read, Write [▶ 59], Set Default [▶ 60], Export/Import [▶ 61] and Store [▶ 62] 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 [▶ 65].

Application specific information can be specified in parameter (0x0018) Application Specific Tag [▶ 66].

"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").

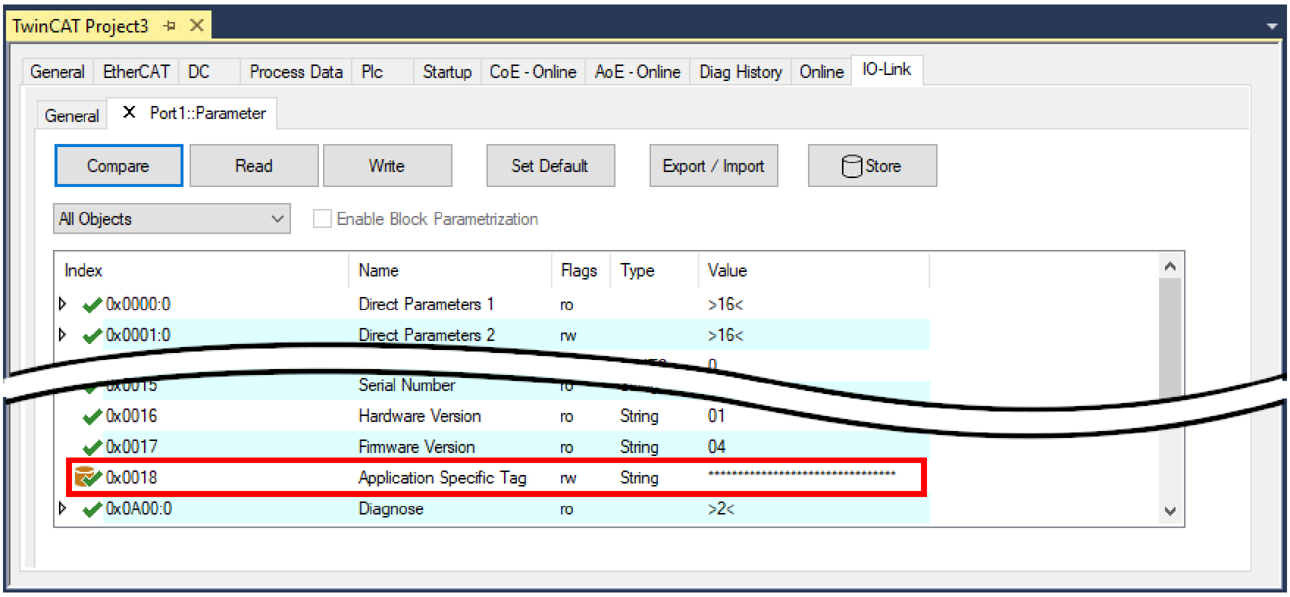


Fig. 44: Display of matching data in the "Parameter" tab

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").

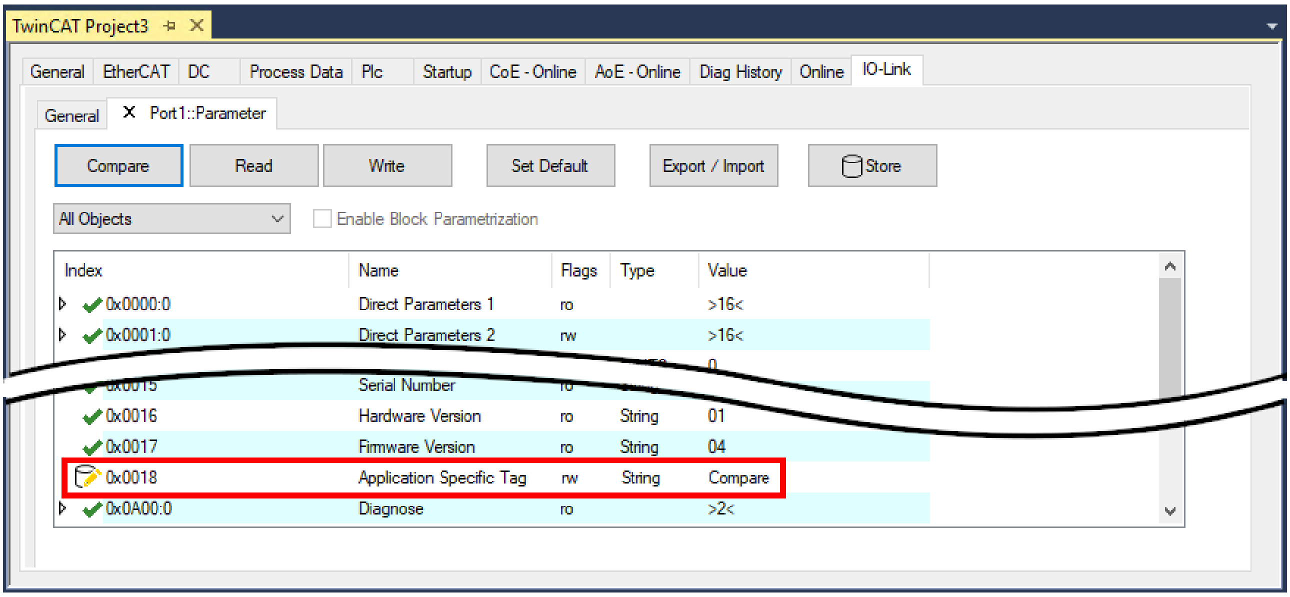


Fig. 45: Display of deviating data in the "Parameter" tab

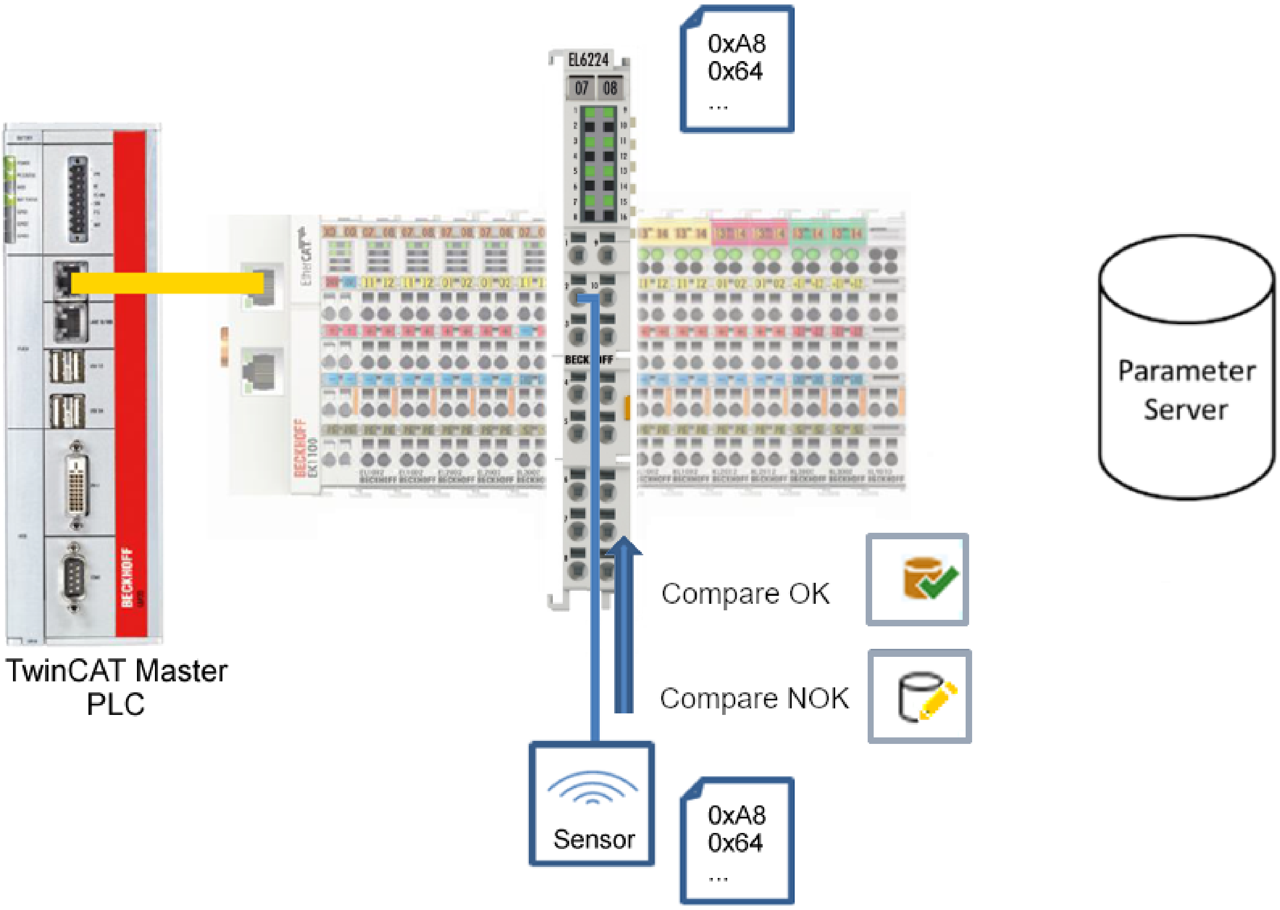


Fig. 46: Compare configuration and sensor data

“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.

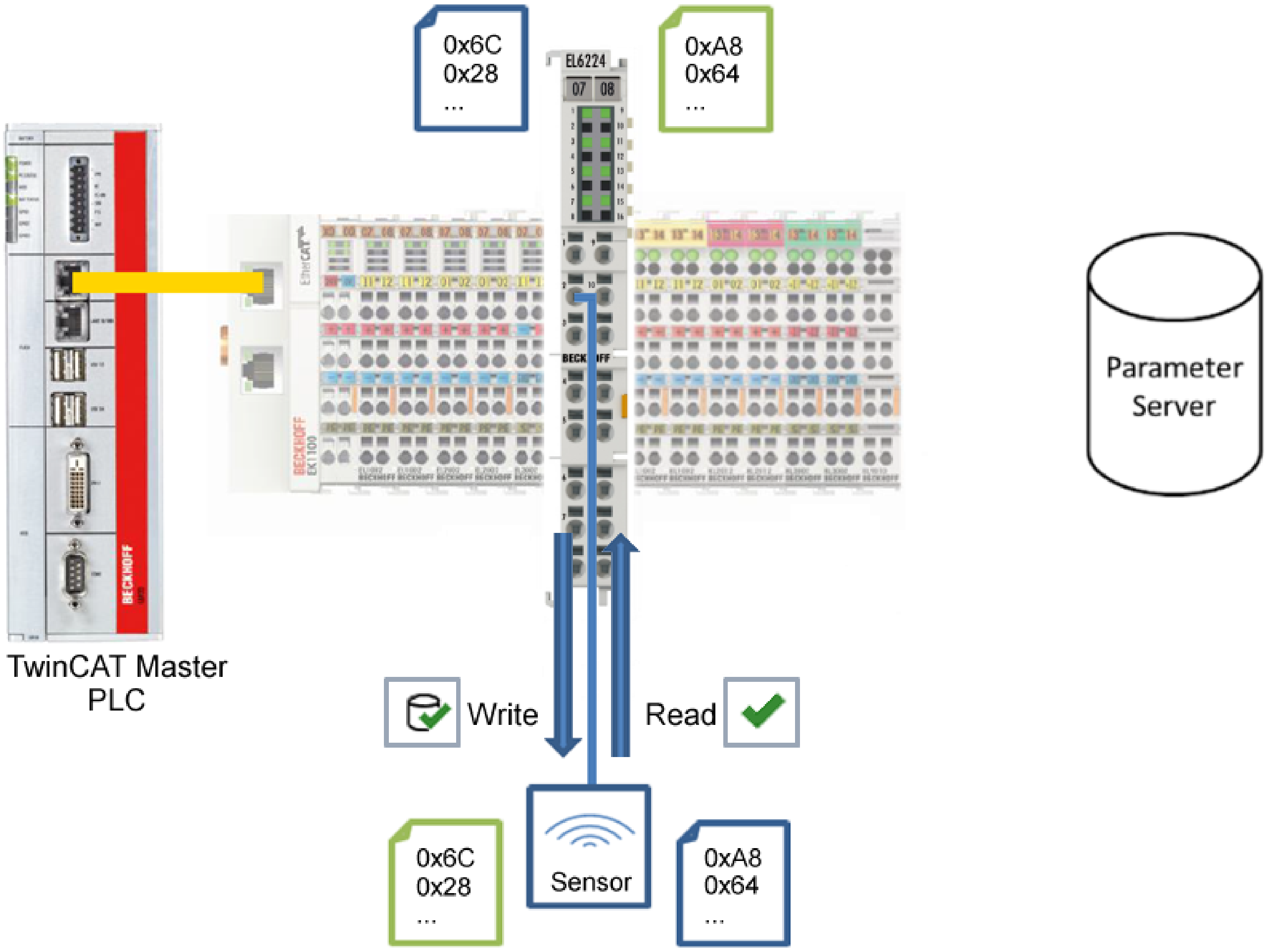


Fig. 47: Write parameter data to the sensor, read parameter data from the sensor.

“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.

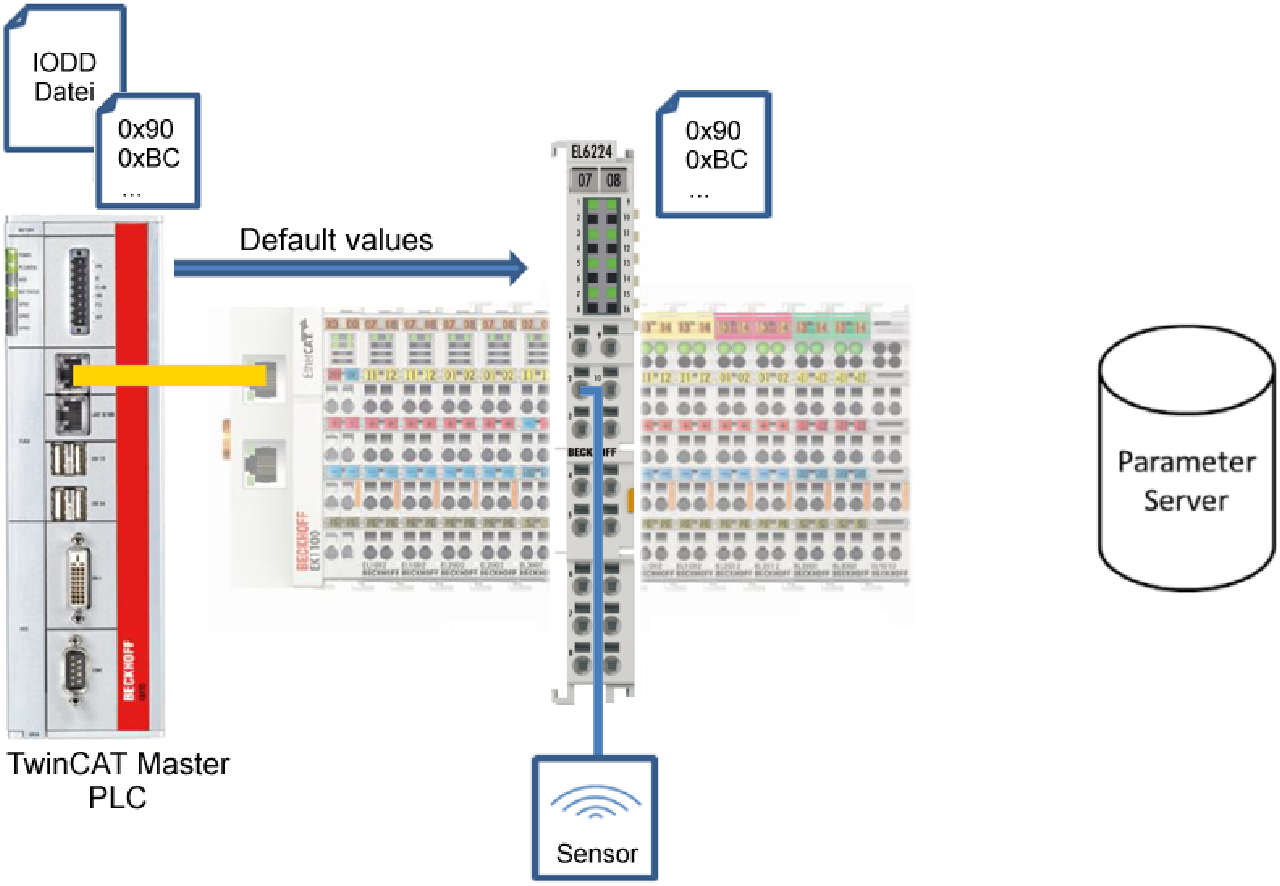


Fig. 48: Reset parameter values to default

"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.

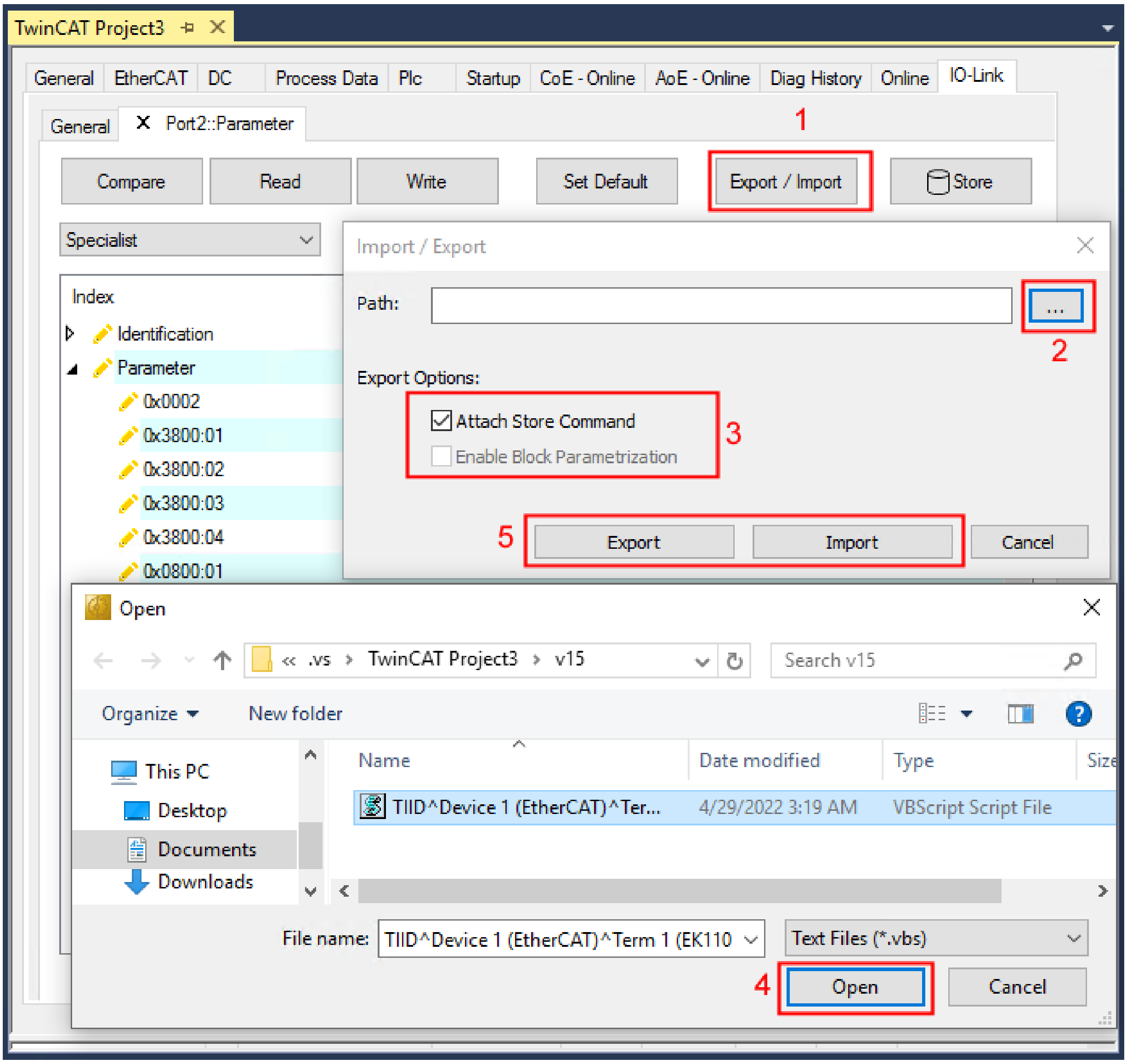


Fig. 49: Parameterization IO-Link device - Export / Import

“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

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

⇨ The stored values are displayed.

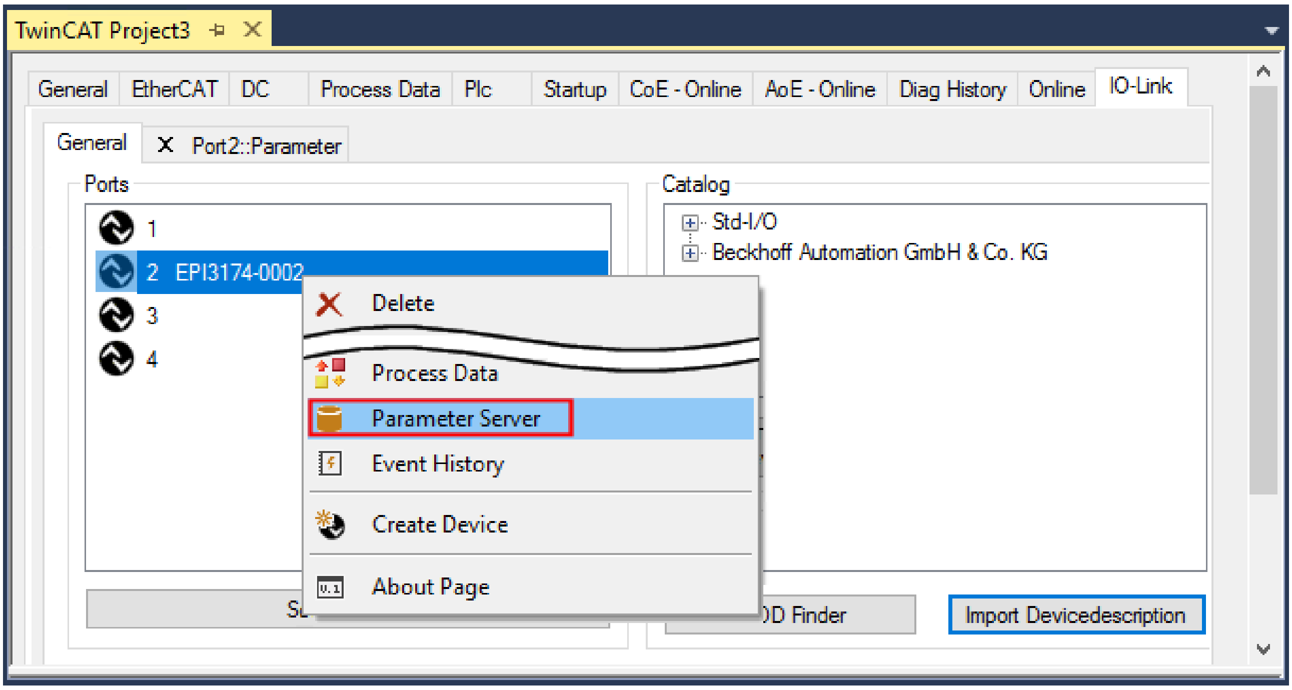


Fig. 50: Open the “ServerParameter” tab

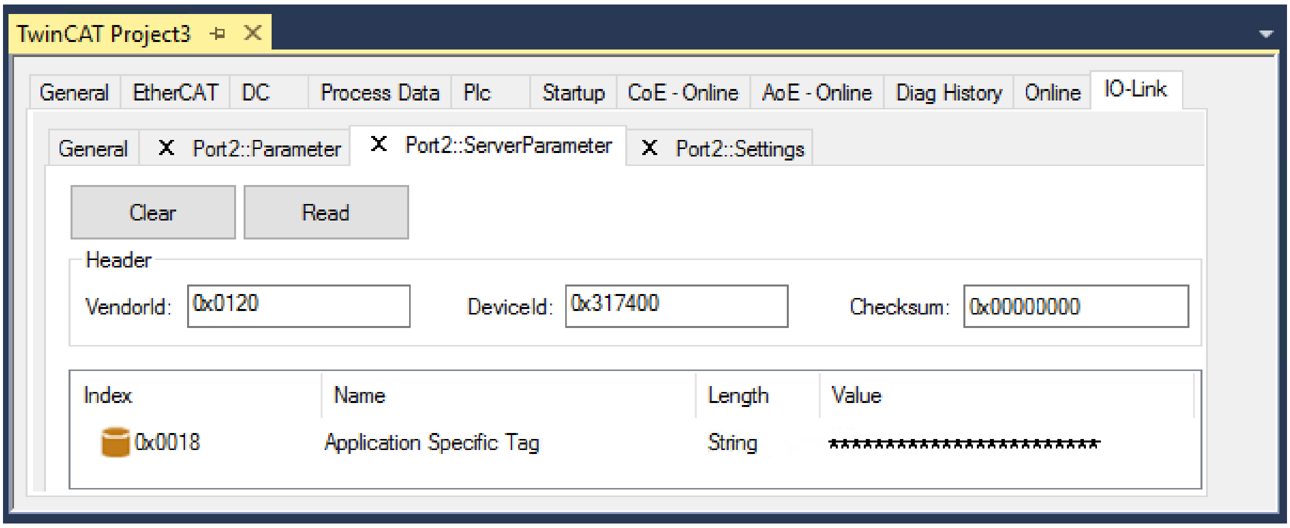


Fig. 51: „ServerParameter“ tab

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

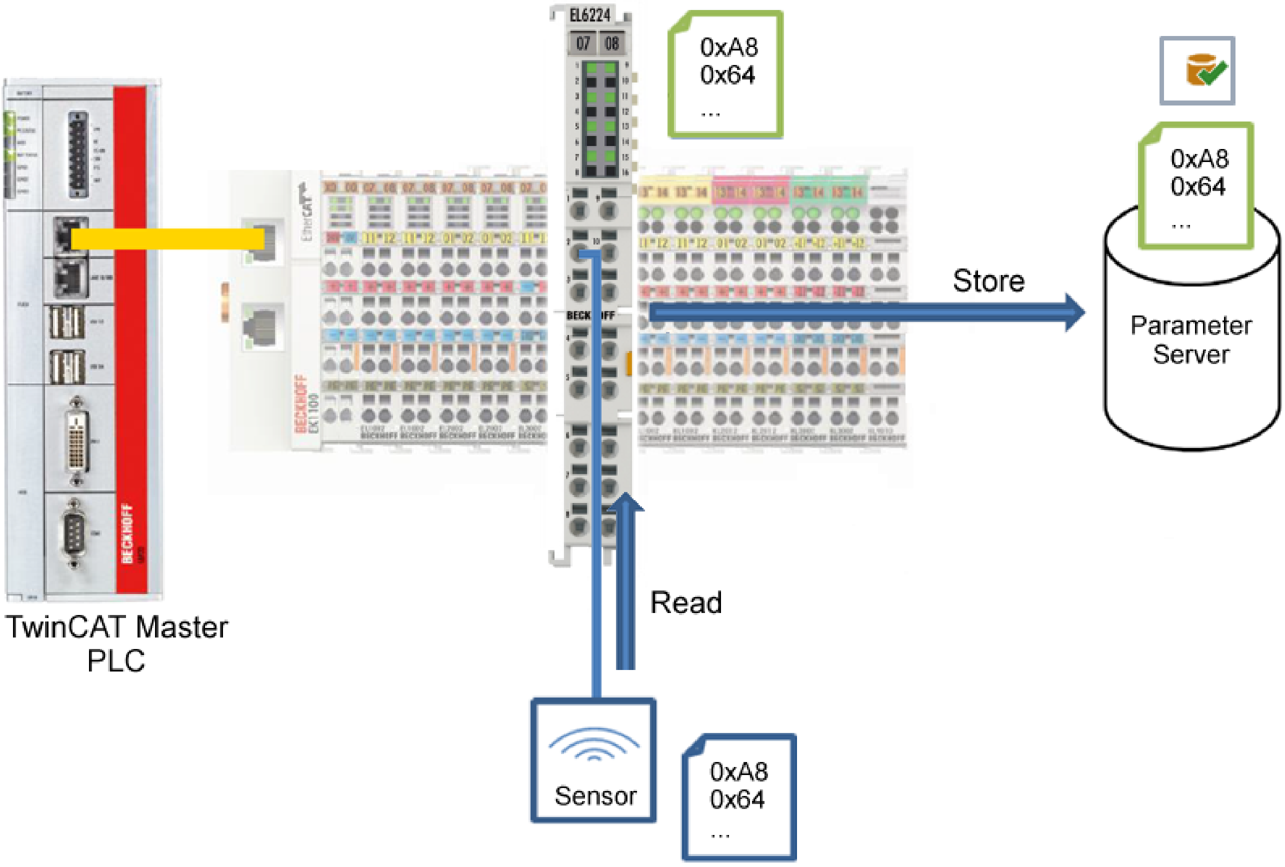


Fig. 53: Store parameters

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 [▸ 59]).

⇨ 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.

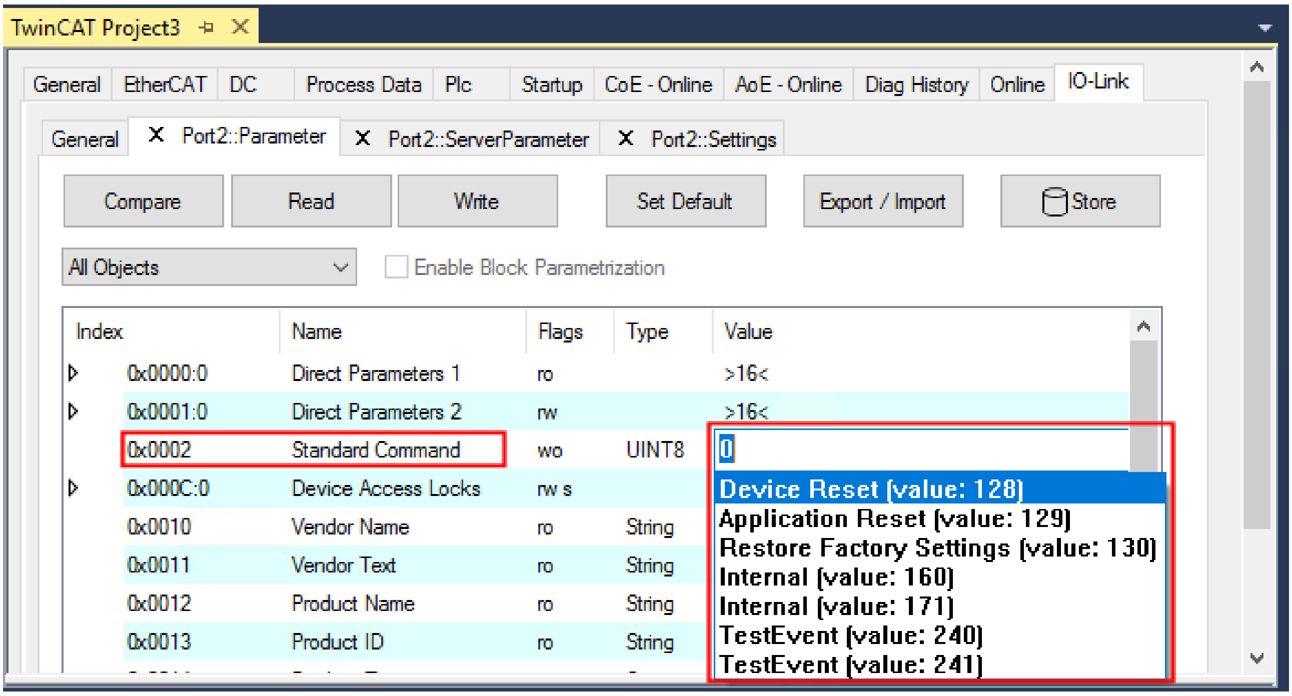


Fig. 54: Parameters IO-Link device “Standard Command”

"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 [▸ 59] button and the Store [▸ 62] button, if required (as described above).

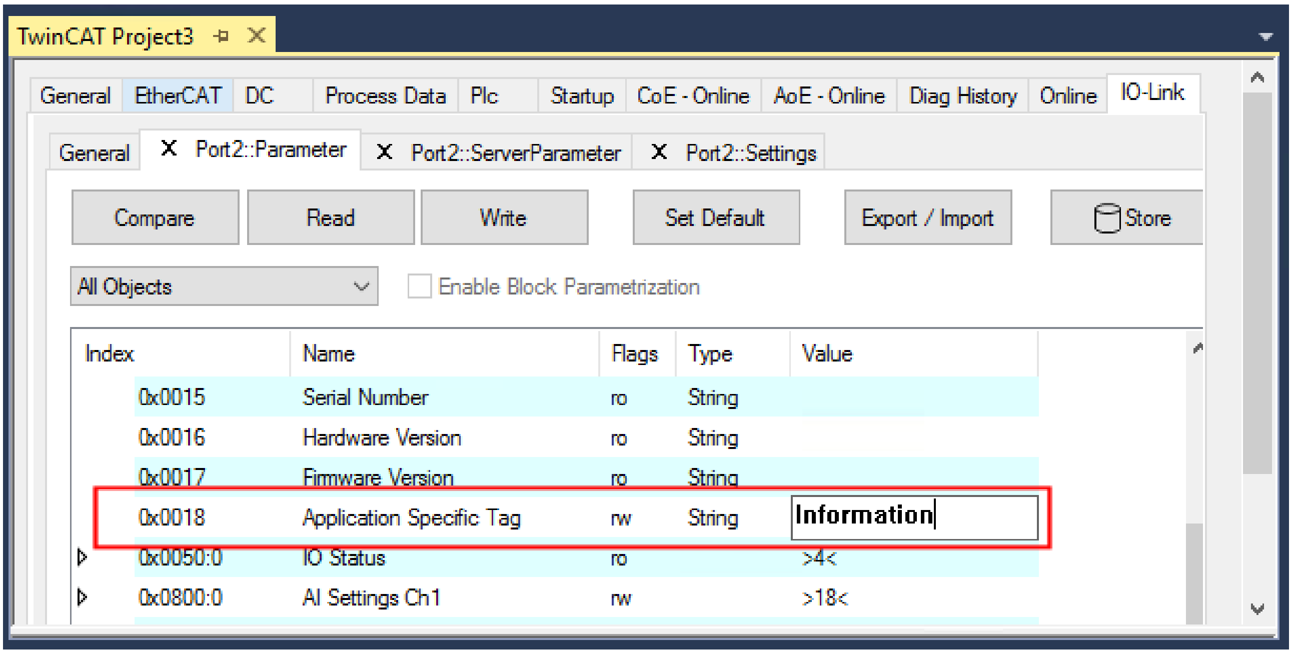


Fig. 55: Parameters IO-Link device: "Application Specific Tag"

8 Access to IO-Link data

8.1 IO-Link system communication

The Beckhoff EJ6224 IO-Link master plug-in module is divided into two services.

- On the one hand it represents an IO-Link master in relation to the connected IO-Link devices,
- while on the other it is an EtherCAT slave in relation to the PLC TwinCAT master.

The system communication is illustrated in the following figure.

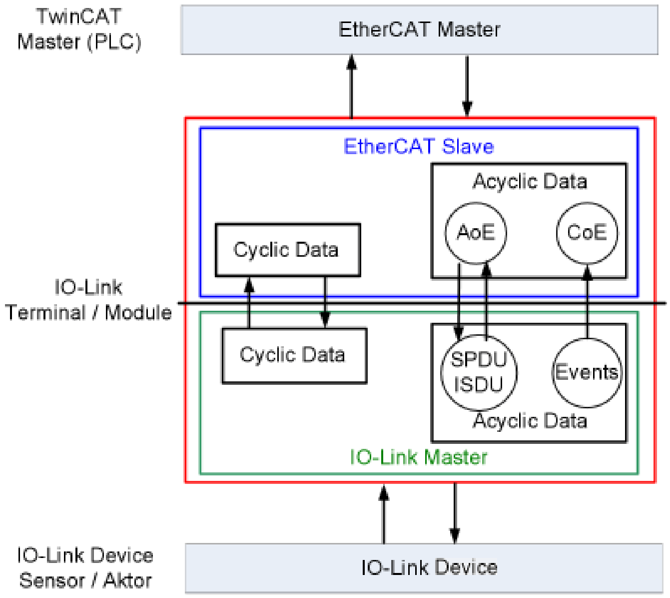


Fig. 56: Illustration of the system communication of an EtherCAT master

In principle cyclic and acyclic data are exchanged. The cyclic process data can be accessed via the PDOs, the acyclic data via AoE [▸ 69]. The events are additionally displayed under Diag History [▸ 72] in the System Manager.

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

8.2 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 [▶ 53].

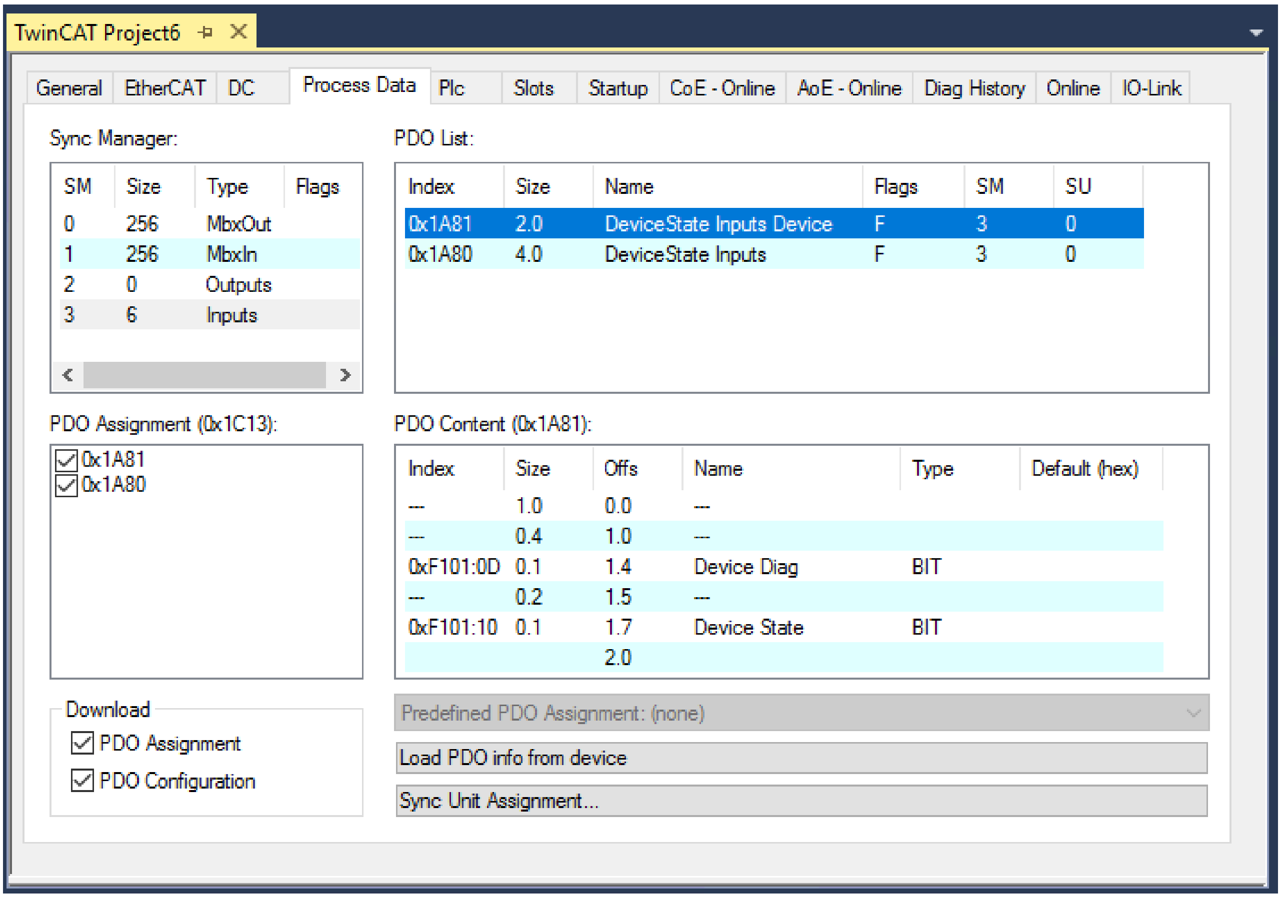


Fig. 57: Illustration of the process data allocation, showing as example SM3 inputs of EJ6224

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 octed strings.

Via index 0xF100:0n the status of the IO-Link ports [▶ 106] 1-4 is indicated.
The indices 0xF101:xx provide general diagnostic 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 devices is reported via a status bit
0xF100:01	1.0	State Ch.1	See table “Meaning 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.

8.3 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.

8.4 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 0xA0n0:01):

- 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)

8.5 ADS

Communication relating to IO link demand data is initiated via an ADS command. An ADS address always consists of a NetID and PortNo. TwinCAT relays ADS commands to EL6224/EJ6224 via AoE (**A**DS **o**ver **E**therCAT), from where the command is relayed to the IO link master section and therefore the data channel.

AoE NetID

The EL6224/EJ6224 is assigned a dedicated AoE-NetID for communication with the IO link master section. This is allocated by the configuration tool (fig. *AoE-NetID allocation*).
NetID under "EL6224/EJ6224" -> „EtherCAT"-> „Advanced Settings" -> „Mailbox" -> "AoE""

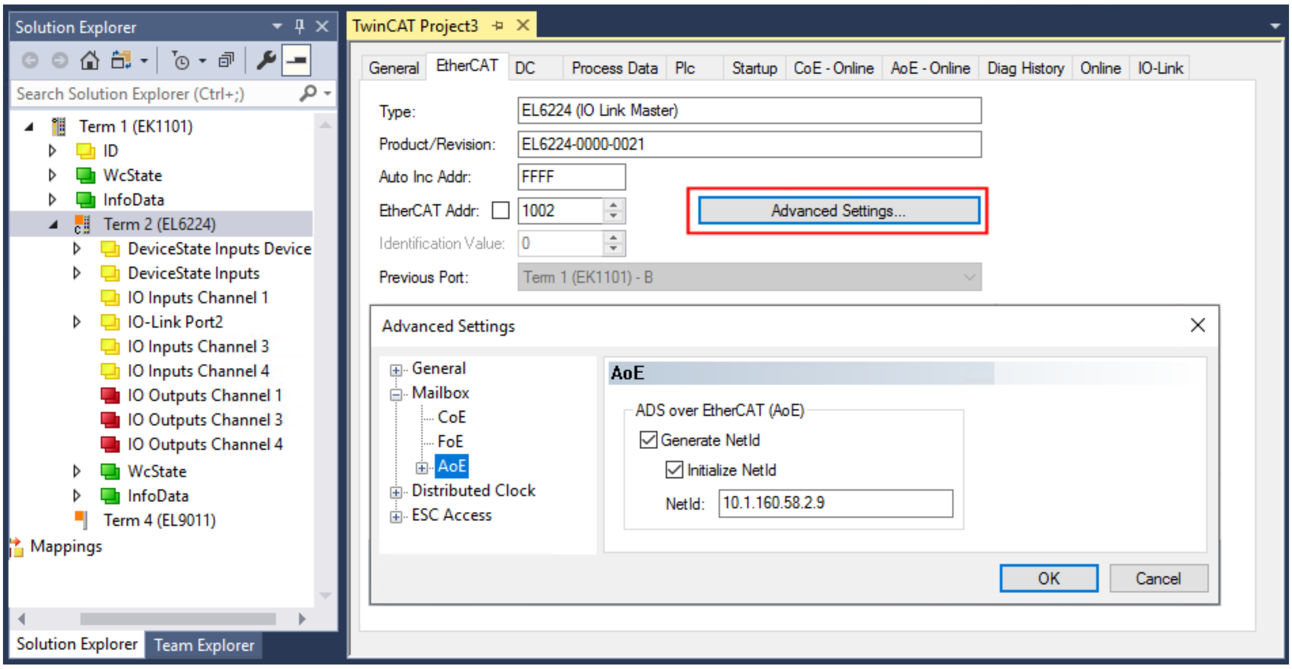


Fig. 58: AoE-NetID allocation using the example of EL6224

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. i.e. IO-Link Port1 corresponds to PortNo 0x1000 and IO-Link Portn corresponds to PortNo 0x1000 + n-1.
The following specification applies for the EL6224 (4-port IO link master):

- IO-Link Port1 corresponds to PortNo 0x1000
- IO-Link Port2 corresponds to PortNo 0x1001
- IO-Link Port3 corresponds to PortNo 0x1002
- IO-Link Port4 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

8.6 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 bit. Further information on the events can be read via the CoE directory or the DiagHistory tab.

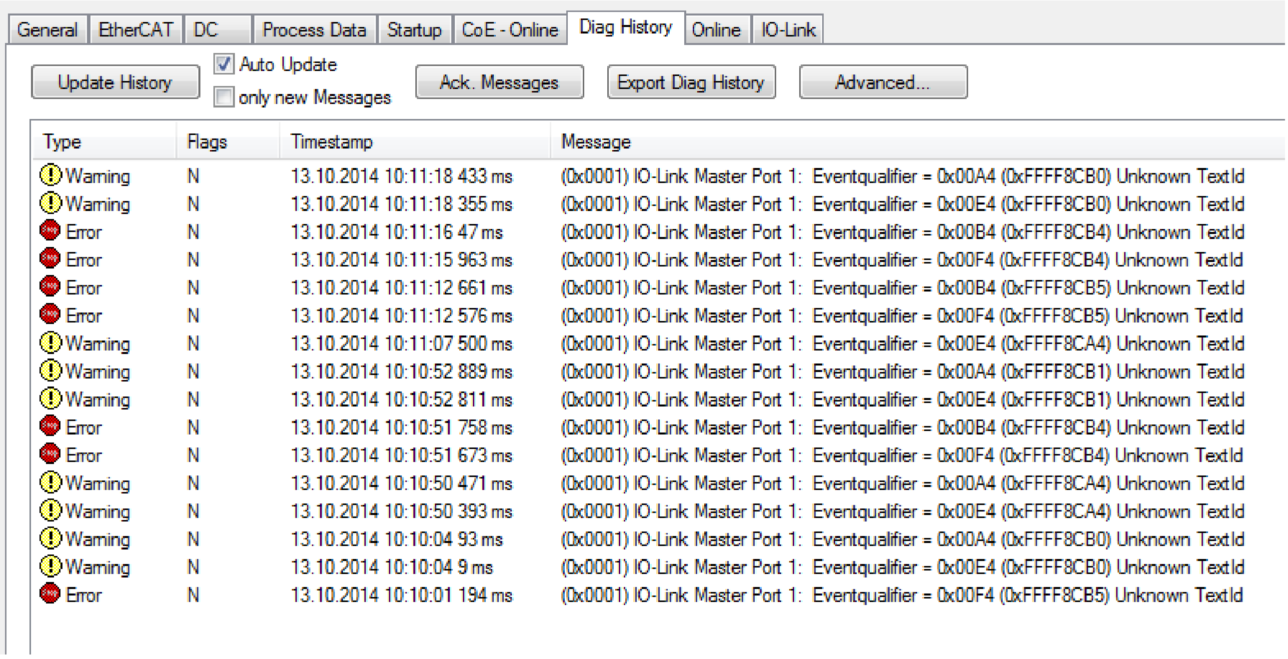


Fig. 59: DiagHistory tab

The events are arranged according to type (information, warning, error), flag, occurrence of the event (time stamp) and message (port number & 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. 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.

8.7 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

9 EJ6224 - Object description and parameterization

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.

NOTE

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:
  - Store parameters [▸ 73] Index 0x1010
  - Restore object [▸ 74] Index 0x1011
  - Configuration data [▸ 74] Index 0x80n0
- Profile-specific objects:
  - Information and diagnostic data [▸ 76] Index 0x10F3, 0x10F8, 0x90n0 .. 0xF900
- Standard objects [▸ 80]

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

9.1 Store parameters

Index 1010 Store parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1010:0	Store parameters	Max. Subindex	UINT8	RO	0x01 (1_{dec})
1010:01	SubIndex 001	Changes in the configuration can be saved permanently by entering **0x65766173**	UINT32	RW	0x00000000 (0_{dec})

9.2 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})

9.3 Configuration data

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

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	IO Settings Ch.(n+1)	Max. Subindex (hex)	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	Vendor ID	The vendor ID is used for validating the manufacturer 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 transferred in the IO-Link format for Min Cycle Time. Bit 6 und 7: Time Base Bit 0 to 5: Multiplier (see Table 1)	UINT8	RW	0x00 (0_{dec})

Table 1

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 transferred 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 IO mode [bit set] Bit 0 to 4: LENGTH length of the process data	UINT8	RW	0x00 (0_{dec})
80n0:25	Process data out length	These parameters are transferred 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 IO mode [bit set] Bit 0 to 4: LENGTH length of the process data	UINT16	RW	0x0000 (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 operating 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 (none cyclic communication, data are exchanged on demand). Bits 4...15 2: Data storage active 4: Data storage upload inactive	UINT16	RW	0x0000 (0_{dec})

9.4 Information and diagnostic data

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Max. Subindex	UINT8	RO	0x15 (21$_{dec}$)
10F3:01	Maximum Messages	Maximum number of stored messages. A maximum of 16 messages can be stored	UINT32	RO	0x00000000 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	RO	0x00 (0$_{dec}$)
10F3:04	New Message available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0$_{dec}$)
10F3:05	Flags	not used	UINT16	RO	0x0000 (0$_{dec}$)
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[20]	RO	{0}
…	…	…	…	…	…
10F3:015	Diagnosis Message 016	Message 16	OCTET-STRING[20]	RO	{0}

Index 10F8 Actual Time Stamp

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Actual Time Stamp	Timestamp	UINT64	RO	

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

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:0	IO Info data Ch. (n+1)	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	Vendor ID	The vendor ID is used for validating the manufacturer 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 refers to the communication between the IO-Link master and the IO-Link device. This value is transferred in the IO-Link format for Min Cycle Time. Bit 6 und 7: Time Base Bit 0 to 5: Multiplier (see Table 2)	UINT8	RO	0x00 (0$_{dec}$)

Table 2

Time Base	Time base meaning	Calculation	Min. Cycle Time
00$_{bin}$	0.100 ms	Multiplier x Time Base	0.000 - 6.300 ms
01$_{bin}$	0.400 ms	6.4 ms + Multiplier x Time Base	6.400 - 31.600 ms
10$_{bin}$	1.600 ms	32.0 ms + Multiplier x Time Base	32.000 - 132.800 ms
11$_{bin}$	6.400 ms	134.4 ms + Multiplier x Time Base	134.400 - 537.600 ms

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:23	Offset time	reserved	UINT8	RO	0x00 (0_{dec})
90n0:24	Process data in length	These parameters are transferred 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 IO 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 transferred 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 IO 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})

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

Index (hex)	Name	Meaning	Data type	Flags	Default
A0n0:0	IO Diag data Ch. (n+1)	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_INACTIVE 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	0x0008 (8_{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	Length of this object	UINT8	RW	0x04 (4_{dec})
F010:01	SubIndex 001	Revision	UINT32	RW	0x0000184C (6220_{dec})
F010:02	SubIndex 002	-	UINT32	RW	0x0000184C (6220_{dec})
F010:03	SubIndex 003	-	UINT32	RW	0x0000184C (6220_{dec})
F010:04	SubIndex 004	-	UINT32	RW	0x0000184C (6220_{dec})

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 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	RW	0x10 (16_{dec})
F101:0D	Device Diag	TRUE: A new diagnostic message is in the DiagHistory	UINT8	RW	0x00 (0_{dec})
F101:10	Device State	TRUE Collective message if at least 1 slave is faulty	UINT8	RW	0x00 (0_{dec})

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	RW	0x01 (1_{dec})
F900:01	IO-Link version	-	UINT8	RW	0x00 (0_{dec})

9.5 Standard objects (0x1000-0x1FFF)

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

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x184C1389 (4072638921_{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	EJ6224

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	00

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	01

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	0x18502852 (407906386_{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates 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; 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 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 160n IO RxPDO-Map Outputs Ch.1 (n = 0) - Ch.4 (n = 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
16n0:0	IO RxPDO-Map Outputs Ch.(n+)1	PDO Mapping RxPDO 1	UINT8	RO	0x00 (0_{dec})

Index 1A0n IO TxPDO-Map Inputs Ch.1 (n = 0) - Ch.4 (n = 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
1An0:0	IO TxPDO-Map Inputs Ch.(n+1)	PDO Mapping TxPDO 1	UINT8	RO	0x00 (0_{dec})

Index 1A80 DeviceState TxPDO-Map Inputs

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

Index 1A81 DeviceState TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A81:0	DeviceState TxPDO-Map Inputs	PDO Mapping TxPDO 3	UINT8	RO	0x04 (4_{dec})
1A81:01	SubIndex 001	1. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1A81:02	SubIndex 002	2. PDO Mapping entry (object 0xF101 (Device State Status data), entry 0x0D (Device Diag))	UINT32	RO	0xF101:0D, 1
1A81:03	SubIndex 003	3. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A81:04	SubIndex 004	4. PDO Mapping entry (object 0xF101 (DeviceState Status data), entry 0x10 (Device State))	UINT32	RO	0xF101:10, 1

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	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:20	SubIndex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1601 (5633_{dec})
1C12:03	SubIndex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1602 (5634_{dec})
1C12:04	SubIndex 004	4. allocated 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	RO	0x06 (6_{dec})
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A01 (6657_{dez})
1C13:03	SubIndex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A02 (6658_{dec})
1C13:04	SubIndex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A03 (6659_{dec})
1C13:05	SubIndex 005	5. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A80 (6784_{dec})
1C13:06	SubIndex 006	6. allocated 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 inputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0007A120 (500000_{dec})
1C32:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron 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 0x1C33:08)	UINT16	RO	0x0003 (3_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0007A120 (500000_{dec})
1C32:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	With this entry the real required process data provision time can be measured. • 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	RW	0x0000 (0_{dec})
1C32:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	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 occasions 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: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • 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, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron 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 0x1C33:08)	UINT16	RO	0x0003 (3_{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 availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:08	Command	With this entry the real required process data provision time can be measured. • 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	RW	0x0000 (0_{dec})
1C33:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	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 occasions 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})

10 EJ6224-0090 - Object description and parameterization

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.

NOTE

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:
  - Store parameters [▸ 85] Index 0x1010
  - Restore object [▸ 86] Index 0x1011
  - Configuration data [▸ 86] index 0x80n0
- Profile-specific objects:
  - Input data [▸ 87] Index 0x6001 .. 0x6130
  - Information and diagnostic data [▸ 89] Index 0x10F3, 0x10F8, 0x90n0 .. 0xF900
- Standard objects [▸ 93]
- TwinSAFE Single Channel objects [▸ 98] Index 0x1690, 0x1A90, 0x600F, 0x700F, 0x800F

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

10.1 Store parameters

Index 1010 Store parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1010:0	Store parameters	Max. Subindex	UINT8	RO	0x01 (1_{dec})
1010:01	SubIndex 001	Changes in the configuration can be saved permanently by entering **0x65766173**	UINT32	RW	0x00000000 (0_{dec})

10.2 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})

10.3 Configuration data

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

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	IO Settings Ch.(n+1)	Max. Subindex (hex)	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	Vendor ID	The vendor ID is used for validating the manufacturer 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 transferred in the IO-Link format for Min Cycle Time. Bit 6 und 7: Time Base Bit 0 to 5: Multiplier (see Table 1)	UINT8	RW	0x00 (0_{dec})

Table 1

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 transferred 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 IO mode [bit set] Bit 0 to 4: LENGTH length of the process data	UINT8	RW	0x00 (0_{dec})
80n0:25	Process data out length	These parameters are transferred 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 IO mode [bit set] Bit 0 to 4: LENGTH length of the process data	UINT16	RW	0x0000 (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 operating 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 (none cyclic communication, data are exchanged on demand). Bits 4...15 2: Data storage active 4: Data storage upload inactive	UINT16	RW	0x0000 (0_{dec})

10.4 Input data (0x6001-0x6130)

Index 60n1 DWord Inputs Ch.1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n1:0	DWord Inputs Ch. (n+1)	PDO Mapping TxPDO	UINT8	RO	0x1D (29_{dec})
60n1:01	Offset Byte 0	Offset Byte 0	()	RO	()
60n1:02	Offset Byte 1	Offset Byte 1	()	RO	()
…	…	…	…	…	…
60n1:1D	Offset Byte 28	Offset Byte 28	()	RO	()

Index 60n2 unsigned DWord Inputs Ch.1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n2:0	unsigned DWord Inputs Ch. (n+1)	PDO Mapping TxPDO	UINT8	RO	0x1D (29_{dec})
60n2:01	Offset Byte 0	Offset Byte 0	()	RO	()
60n2:02	Offset Byte 1	Offset Byte 1	()	RO	()
…	…	…	…	…	…
60n2:1D	Offset Byte 28	Offset Byte 28	()	RO	()

Index 60n3 Word Inputs Ch.1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n3:0	Word Inputs Ch. (n+1)	PDO Mapping TxPDO	UINT8	RO	0x1F (31_{dec})
60n3:01	Offset Byte 0	Offset Byte 0	()	RO	()
60n3:02	Offset Byte 1	Offset Byte 1	()	RO	()
…	…	…	…	…	…
60n3:1F	Offset Byte 30	Offset Byte 30	()	RO	()

Index 60n4 unsigned Word Inputs Ch.1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n4:0	Unsigned Word Inputs Ch. (n+1)	PDO Mapping TxPDO	UINT8	RO	0x1F (31_{dec})
60n4:01	Offset Byte 0	Offset Byte 0	()	RO	()
60n4:02	Offset Byte 1	Offset Byte 1	()	RO	()
…	…	…	…	…	…
60n4:1F	Offset Byte 30	Offset Byte 30	()	RO	()

Index 61n0 Bool Inputs Ch.1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
61n0:0	Bool Inputs Ch. (n+1)	PDO Mapping TxPDO	UINT8	RO	0x40 (64_{dec})
61n0:01	Offset Byte 0 Bit 0	Offset Byte 0 Bit 0	()	RO	()
61n0:02	Offset Byte 0 Bit 1	Offset Byte 0 Bit 1	()	RO	()
…	…	…	…	…	…
61n0:08	Offset Byte 0 Bit 7	Offset Byte 0 Bit 7	()	RO	()
61n0:09	Offset Byte 1 Bit 0	Offset Byte 1 Bit 0	()	RO	()
61n0:0A	Offset Byte 1 Bit 1	Offset Byte 1 Bit 1	()	RO	()
…	…	…	…	…	…
61n0:10	Offset Byte 1 Bit 7	Offset Byte 1 Bit 7	()	RO	()
61n0:11	Offset Byte 2 Bit 0	Offset Byte 2 Bit 0	()	RO	()
61n0:02	Offset Byte 2 Bit 1	Offset Byte 2 Bit 1	()	RO	()
…	…	…	…	…	…
61n0:18	Offset Byte 2 Bit 7	Offset Byte 2 Bit 7	()	RO	()
61n0:19	Offset Byte 3 Bit 0	Offset Byte 3 Bit 0	()	RO	()
61n0:02	Offset Byte 3 Bit 1	Offset Byte 3 Bit 1	()	RO	()
…	…	…	…	…	…
61n0:20	Offset Byte 3 Bit 7	Offset Byte 3 Bit 7	()	RO	()
61n0:21	Offset Byte 4 Bit 0	Offset Byte 4 Bit 0	()	RO	()
61n0:22	Offset Byte 4 Bit 1	Offset Byte 4 Bit 1	()	RO	()
…	…	…	…	…	…
61n0:28	Offset Byte 4 Bit 7	Offset Byte 4 Bit 7	()	RO	()
61n0:29	Offset Byte 5 Bit 0	Offset Byte 5 Bit 0	()	RO	()
61n0:2A	Offset Byte 5 Bit 1	Offset Byte 5 Bit 1	()	RO	()
…	…	…	…	…	…
61n0:30	Offset Byte 5 Bit 7	Offset Byte 5 Bit 7	()	RO	()
61n0:31	Offset Byte 6 Bit 0	Offset Byte 6 Bit 0	()	RO	()
61n0:32	Offset Byte 6 Bit 1	Offset Byte 6 Bit 1	()	RO	()
…	…	…	…	…	…
61n0:38	Offset Byte 6 Bit 7	Offset Byte 6 Bit 7	()	RO	()
61n0:39	Offset Byte 7 Bit 0	Offset Byte 7 Bit 0	()	RO	()
61n0:3A	Offset Byte 7 Bit 1	Offset Byte 7 Bit 1	()	RO	()
…	…	…	…	…	…
61n0:40	Offset Byte 7 Bit 7	Offset Byte 7 Bit 7	()	RO	()

10.5 Information and diagnostic data

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Max. Subindex	UINT8	RO	0x15 (21_{dec})
10F3:01	Maximum Messages	Maximum number of stored messages. A maximum of 16 messages can be stored	UINT32	RO	0x00000000 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	RO	0x00 (0_{dec})
10F3:04	New Message available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0_{dec})
10F3:05	Flags	not used	UINT16	RO	0x0000 (0_{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[20]	RO	{0}
…	…	…	…	…	…
10F3:015	Diagnosis Message 016	Message 16	OCTET-STRING[20]	RO	{0}

Index 10F8 Actual Time Stamp

Index (hex)	Name	Meaning	Data type	Flags	Default
10F8:0	Actual Time Stamp	Timestamp	UINT64	RO	

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

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:0	IO Info data Ch. (n+1)	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	Vendor ID	The vendor ID is used for validating the manufacturer 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 refers to the communication between the IO-Link master and the IO-Link device. This value is transferred in the IO-Link format for Min Cycle Time. Bit 6 und 7: Time Base Bit 0 to 5: Multiplier (see Table 2)	UINT8	RO	0x00 (0_{dec})

Table 2

Time Base	Time base meaning	Calculation	Min. Cycle Time
00_{bin}	0.100 ms	Multiplier x Time Base	0.000 - 6.300 ms
01_{bin}	0.400 ms	6.4 ms + Multiplier x Time Base	6.400 - 31.600 ms
10_{bin}	1.600 ms	32.0 ms + Multiplier x Time Base	32.000 - 132.800 ms
11_{bin}	6.400 ms	134.4 ms + Multiplier x Time Base	134.400 - 537.600 ms

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:23	Offset time	reserved	UINT8	RO	0x00 (0_{dec})
90n0:24	Process data in length	These parameters are transferred 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 IO 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 transferred 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 IO 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})

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

Index (hex)	Name	Meaning	Data type	Flags	Default
A0n0:0	IODiag data Ch. (n+1)	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_INACTIVE 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	0x0008 (8_{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	Length of this object	UINT8	RW	0x04 (4_{dec})
F010:01	SubIndex 001	Revision	UINT32	RW	0x0000184C (6220_{dec})
F010:02	SubIndex 002	-	UINT32	RW	0x0000184C (6220_{dec})
F010:03	SubIndex 003	-	UINT32	RW	0x0000184C (6220_{dec})
F010:04	SubIndex 004	-	UINT32	RW	0x0000184C (6220_{dec})

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 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	RW	0x10 (16_{dec})
F101:0C	Device Diag	TRUE: A new diagnostic message is in the DiagHistory	UINT8	RW	0x00 (0_{dec})
F100:10	Device State	TRUE Collective message if at least 1 slave is faulty	UINT8	RW	0x00 (0_{dec})

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	RW	0x01 (1_{dec})
F900:01	IO-Link version	-	UINT8	RW	0x10 (16_{dec})

10.6 Standard objects (0x1000-0x1FFF)

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

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x184C1389 (4072638921$_{dec}$)

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EJ6224-0090

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	00

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	01

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	0x18502852 (407906386$_{dec}$)
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates 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; 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 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 160n IO RxPDO-Map Outputs Ch.1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
16n0:0	IO RxPDO-Map Outputs Ch.(n+)1	PDO Mapping RxPDO 1	UINT8	RO	0x00 (0_{dec})

Index 1A0n IO TxPDO-Map Inputs Ch.1 - 4 (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
1An0:0	IO TxPDO-Map Inputs Ch.(n+1)	PDO Mapping TxPDO 1	UINT8	RO	0x00 (0_{dec})

Index 1A80 DeviceState TxPDO-Map Inputs

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

Index 1A81 DeviceState TxPDO-Map Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
1A81:0	DeviceState TxPDO-Map Inputs	PDO Mapping TxPDO 3	UINT8	RO	0x04 (4_{dec})
1A81:01	SubIndex 001	1. PDO Mapping entry (12 bits align)	UINT32	RO	0x0000:00, 12
1A81:02	SubIndex 002	2. PDO Mapping entry (object 0xF101 (Device State Status data), entry 0x0D (Device Diag))	UINT32	RO	0xF101:0D, 1
1A81:03	SubIndex 003	3. PDO Mapping entry (2 bits align)	UINT32	RO	0x0000:00, 2
1A81:04	SubIndex 004	4. PDO Mapping entry (object 0xF101 (DeviceState Status data), entry 0x10 (Device State))	UINT32	RO	0xF101:10, 1

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	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:20	SubIndex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1601 (5633_{dec})
1C12:03	SubIndex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1602 (5634_{dec})
1C12:04	SubIndex 004	4. allocated 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	RO	0x06 (6_{dec})
1C13:01	SubIndex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656_{dec})
1C13:02	SubIndex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A01 (6657_{dez})
1C13:03	SubIndex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A02 (6658_{dec})
1C13:04	SubIndex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A03 (6659_{dec})
1C13:05	SubIndex 005	5. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A80 (6784_{dec})
1C13:06	SubIndex 006	6. allocated 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 inputs	UINT8	RO	0x20 (32_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0001 (1_{dec})
1C32:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x0007A120 (500000_{dec})
1C32:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron 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 0x1C33:08)	UINT16	RO	0x0003 (3_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x0007A120 (500000_{dec})
1C32:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C32:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	With this entry the real required process data provision time can be measured. • 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	RW	0x0000 (0_{dec})
1C32:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	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 occasions 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: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0022 (34_{dec})
1C33:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchron with SM 2 Event: Master cycle time • 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, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron 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 0x1C33:08)	UINT16	RO	0x0003 (3_{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 availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0_{dec})
1C33:07	Minimum delay time	Min. time between SYNC1 event and the reading of the inputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C33:08	Command	With this entry the real required process data provision time can be measured. • 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	RW	0x0000 (0_{dec})
1C33:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	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 occasions 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})

10.7 TwinSAFE Single Channel objects (EJ6224-0090)

Index 1690 TSC RxPDO-Map Master Message

Index (hex)	Name	Meaning	Data type	Flags	Default
1690:0	TSC RxPDO-Map Master Message	PDO Mapping RxPDO	UINT8	RO	0x04 (4_{dec})
1690:01	SubIndex 001	1. PDO Mapping entry (object 0x700F (TSC Master Frame Elements), entry 0x01 (Master Cmd))	UINT32	RO	0x700F:01, 8
1690:02	SubIndex 002	2. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1690:03	SubIndex 003	3. PDO Mapping entry (object 0x700F (TSC Master Frame Elements), entry 0x03 (Master CRC_0))	UINT32	RO	0x700F:03, 16
1690:04	SubIndex 004	4. PDO Mapping entry (object 0x700F (TSC Master Frame Elements), entry 0x02 (Master ConnID))	UINT32	RO	0x700F:02, 16

Index 1A90 TSC TxPDO-Map Slave Message

Index (hex)	Name	Meaning	Data type	Flags	Default
1A90:0	TSC TxPDO-Map Slave Message	PDO Mapping TxPDO	UINT8	RO	0x12 (18_{dec})
1A90:01	SubIndex 001	1. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x01 (Slave Cmd))	UINT32	RO	0x600F:01, 8
1A90:02	SubIndex 002	2. PDO Mapping entry (object 0x6004 (unsigned Word Inputs Ch.1), entry 0x01 (Offset Byte 0))	UINT32	RO	0x6004:01, 16
1A90:03	SubIndex 003	3. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x03 (Slave CRC_0))	UINT32	RO	0x600F:03, 16
1A90:04	SubIndex 004	4. PDO Mapping entry (16 bits align)	UINT32	RO	0x0000:00, 16
1A90:05	SubIndex 005	5. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x04 (Slave CRC_1))	UINT32	RO	0x600F:04, 16
1A90:06	SubIndex 006	6. PDO Mapping entry (16 bits align)	UINT32	RO	0x0000:00, 16
1A90:07	SubIndex 007	7. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x05 (Slave CRC_2))	UINT32	RO	0x600F:05, 16
1A90:08	SubIndex 008	8. PDO Mapping entry (16 bits align)	UINT32	RO	0x0000:00, 16
1A90:09	SubIndex 009	9. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x06 (Slave CRC_3))	UINT32	RO	0x600F:06, 16
1A90:0A	SubIndex 010	10. PDO Mapping entry (16 bits align)	UINT32	RO	0x0000:00, 16
1A90:0B	SubIndex 011	11. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x07 (Slave CRC_4))	UINT32	RO	0x600F:07, 16
1A90:0C	SubIndex 012	12. PDO Mapping entry (16 bits align)	UINT32	RO	0x0000:00, 16
1A90:0D	SubIndex 013	13. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x08 (Slave CRC_5))	UINT32	RO	0x600F:08, 16
1A90:0E	SubIndex 014	14. PDO Mapping entry (16 bits align)	UINT32	RO	0x0000:00, 16
1A90:0F	SubIndex 015	15. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x09 (Slave CRC_6))	UINT32	RO	0x600F:09, 16
1A90:10	SubIndex 016	16. PDO Mapping entry (16 bits align)	UINT32	RO	0x0000:00, 16
1A90:11	SubIndex 017	17. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x0A (Slave CRC_7))	UINT32	RO	0x600F:0A, 16
1A90:12	SubIndex 018	18. PDO Mapping entry (object 0x600F (TSC Slave Frame Elements), entry 0x02 (Slave ConnID))	UINT32	RO	0x600F:02, 16

Index 600F TSC Slave Frame Elements

Index (hex)	Name	Meaning	Data type	Flags	Default
6000F:0	TSC Slave Frame Elements	PDO Mapping TxPDO	UINT8	RO	0x0A (10_{dec})
6000F:01	Slave Cmd	reserved	UINT8	RO	0x00 (0_{dec})
6000F:02	Slave ConnID	reserved	UINT16	RO	0x0000 (0_{dec})
6000F:03	Slave CRC_0	reserved	UINT16	RO	0x0000 (0_{dec})
6000F:04	Slave CRC_1	reserved	UINT16	RO	0x0000 (0_{dec})
6000F:05	Slave CRC_2	reserved	UINT16	RO	0x0000 (0_{dec})
6000F:06	Slave CRC_3	reserved	UINT16	RO	0x0000 (0_{dec})
6000F:07	Slave CRC_4	reserved	UINT16	RO	0x0000 (0_{dec})
6000F:08	Slave CRC_5	reserved	UINT16	RO	0x0000 (0_{dec})
6000F:09	Slave CRC_6	reserved	UINT16	RO	0x0000 (0_{dec})
6000F:0A	Slave CRC_7	reserved	UINT16	RO	0x0000 (0_{dec})

Index 700F TSC Master Frame Elements

Index (hex)	Name	Meaning	Data type	Flags	Default
7000F:0	TSC Master Frame Elements	PDO Mapping TxPDO	UINT8	RO	0x03 (3_{dec})
7000F:01	Master Cmd	reserved	UINT8	RO	0x00 (0_{dec})
7000F:02	Master ConnID	reserved	UINT16	RO	0x0000 (0_{dec})
7000F:03	Master CRC_0	reserved	UINT16	RO	0x0000 (0_{dec})

Index 800F TSC Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
8000F:0	TSC Settings	PDO Mapping TxPDO	UINT8	RO	0x02 (2_{dec})
8000F:01	Address	TwinSAFE SC address	UINT16	RO	0x0000 (0_{dec})
8000F:02	Connection Mode	Selection of TwinSAFE SC CRC	UINT32	RO	0x00000000 (0_{dec})

11 EJ6224-0090 – TwinSAFE SC

11.1 TwinSAFE SC

11.1.1 TwinSAFE SC - operating principle

The TwinSAFE SC (Single Channel) technology enables the use of standard signals for safety tasks in any networks of fieldbuses. To do this, EtherCAT Terminals from the areas of analog input, angle/displacement measurement or communication (4…20 mA, incremental encoder, IO-Link, etc.) are extended by the TwinSAFE SC function. The typical signal characteristics and standard functionalities of the I/O components are retained. TwinSAFE SC I/Os have a yellow strip at the front of the housing to distinguish them from standard I/Os.

The TwinSAFE SC technology enables communication via a TwinSAFE protocol. These connections can be distinguished from the usual safe communication via Safety over EtherCAT.

The data of the TwinSAFE SC components are transferred via a TwinSAFE protocol to the TwinSAFE logic, where they can be used in the context of safety-relevant applications. Detailed examples for the correct application of the TwinSAFE SC components and the respective normative classification, which were confirmed/calculated by TÜV SÜD, can be found in the TwinSAFE application manual.

11.1.2 TwinSAFE SC - configuration

The TwinSAFE SC technology enables communication with standard EtherCAT terminals via the Safety over EtherCAT protocol. These connections use another checksum, in order to be able to distinguish between TwinSAFE SC and TwinSAFE. Eight fixed CRCs can be selected, or a free CRC can be entered by the user.

By default the TwinSAFE SC communication channel of the respective TwinSAFE SC component is not enabled. In order to be able to use the data transfer, the corresponding TwinSAFE SC module must first be added under the Slots tab. Only then is it possible to link to a corresponding alias device.

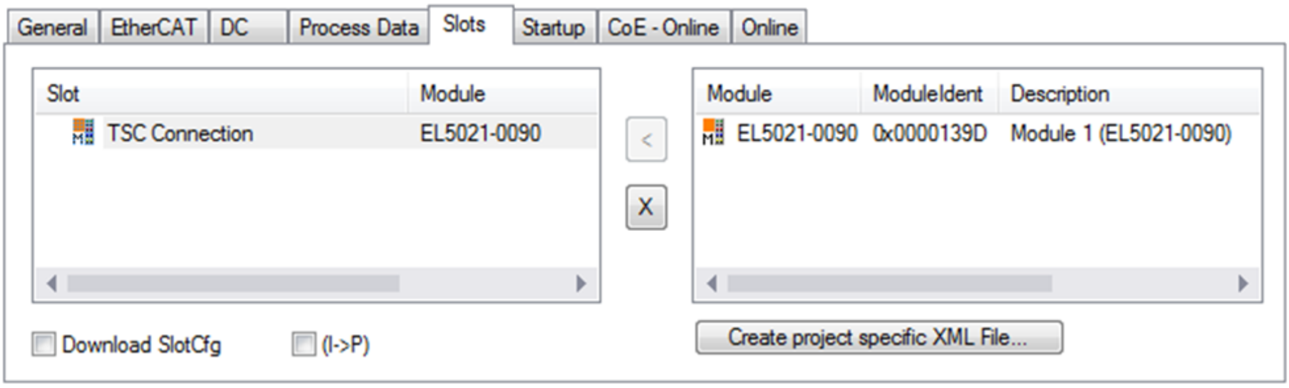


Fig. 60: Adding the TwinSAFE SC process data under the component, e.g. EL5021-0090

Additional process data with the ID TSC Inputs, TSC Outputs are generated (TSC - TwinSAFE Single Channel).

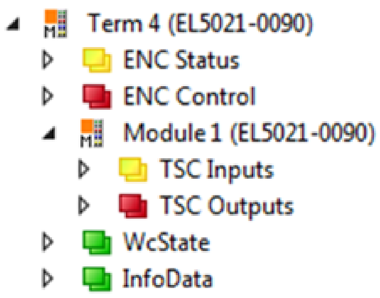


Fig. 61: TwinSAFE SC component process data, example EL5021-0090

A TwinSAFE SC connection is added by adding an alias devices in the safety project and selecting TSC (*TwinSAFE Single Channel*)

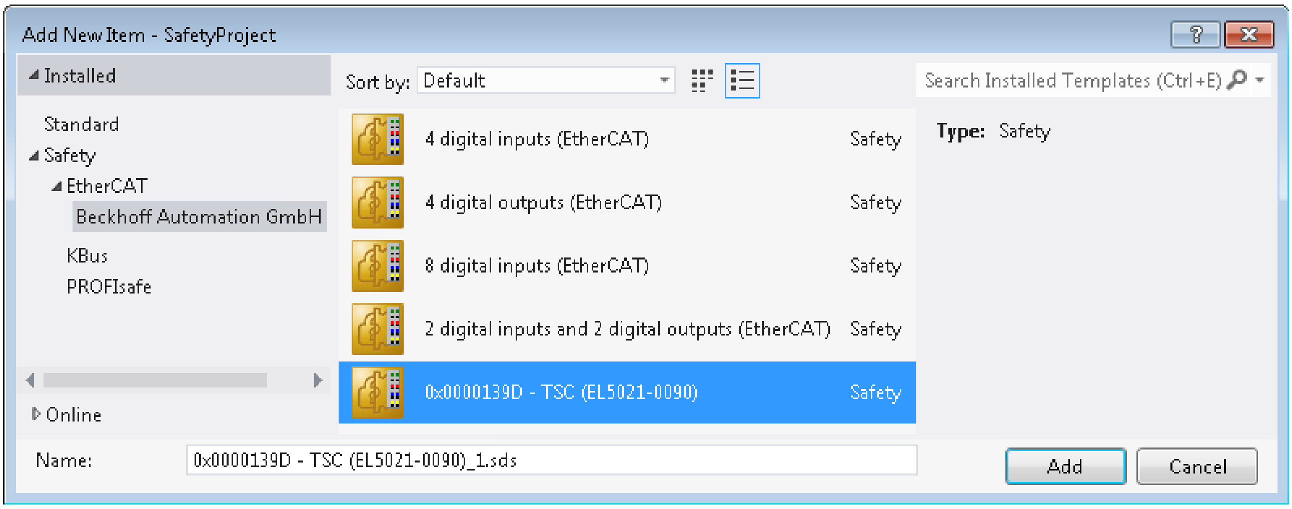


Fig. 62: Adding a TwinSAFE SC connection

After opening the alias device by double-clicking, select the Link button  next to *Physical Device,* in order to create the link to a TwinSAFE SC terminal. Only suitable TwinSAFE SC terminals are offered in the selection dialog.

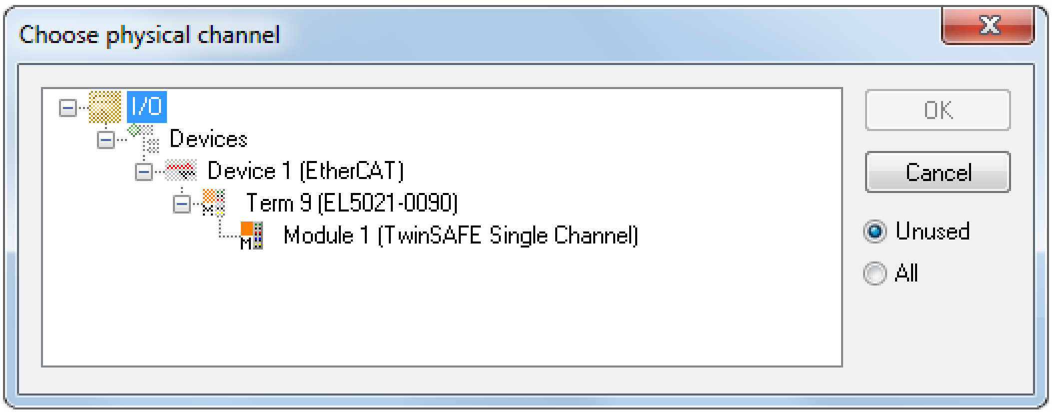


Fig. 63: Creating a link to TwinSAFE SC terminal

The CRC to be used can be selected or a free CRC can be entered under the Connection tab of the alias device.

Entry Mode	Used CRCs
TwinSAFE SC CRC 1 master	0x17B0F
TwinSAFE SC CRC 2 master	0x1571F
TwinSAFE SC CRC 3 master	0x11F95
TwinSAFE SC CRC 4 master	0x153F1
TwinSAFE SC CRC 5 master	0x1F1D5
TwinSAFE SC CRC 6 master	0x1663B
TwinSAFE SC CRC 7 master	0x1B8CD
TwinSAFE SC CRC 8 master	0x1E1BD

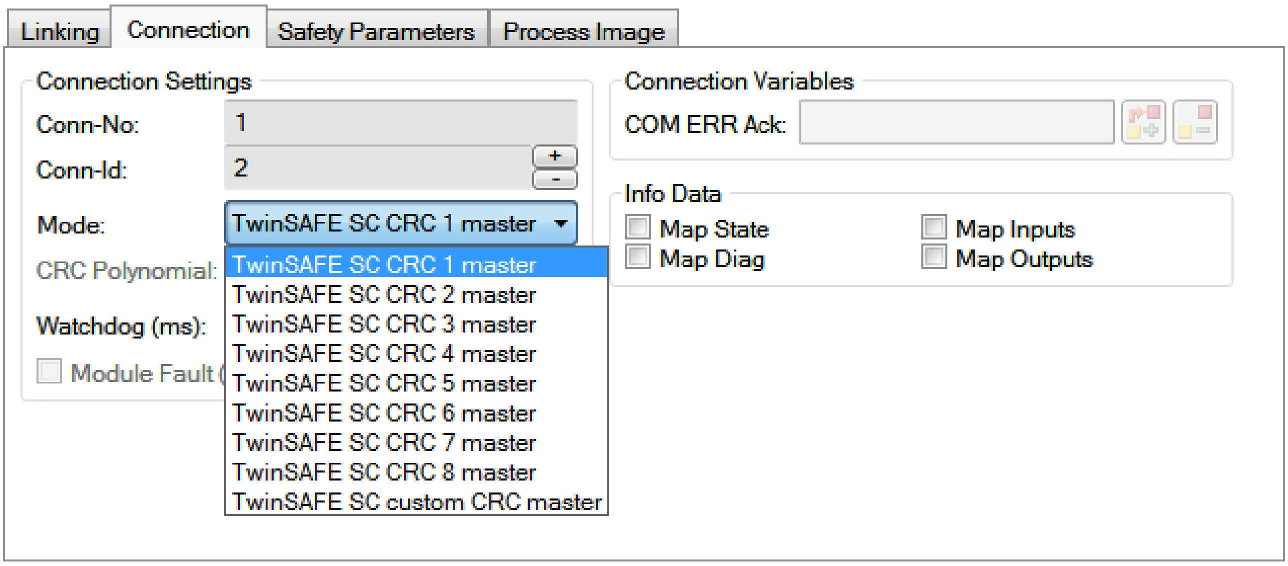


Fig. 64: Selecting a free CRC

These settings must match the settings in the CoE objects of the TwinSAFE SC component.
The TwinSAFE SC component initially makes all available process data available. The *Safety Parameters* tab typically contains no parameters. The process data size and the process data themselves can be selected under the *Process Image* tab.

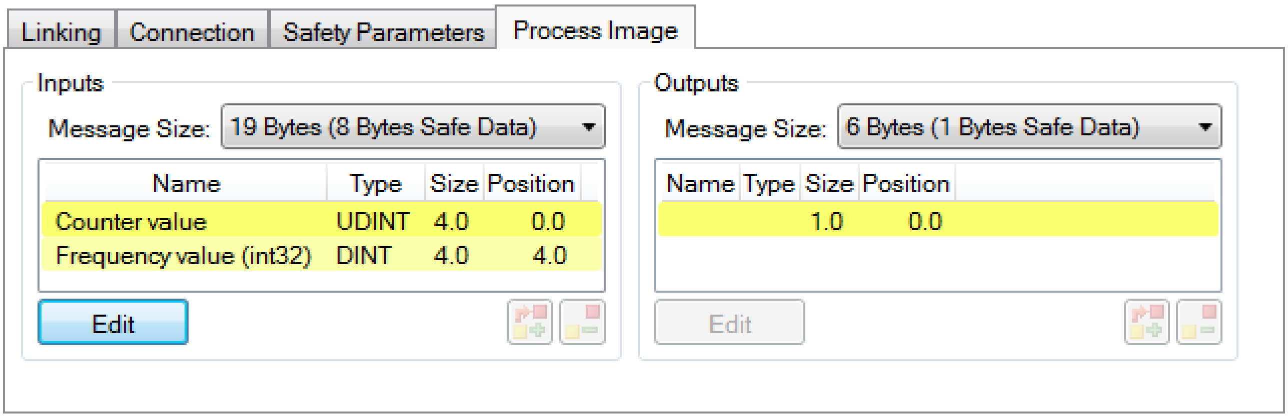


Fig. 65: Selecting the process data size and the process data

The process data (defined in the ESI file) can be adjusted to user requirements by selecting the *Edit* button in the dialog *Configure I/O element(s)*.

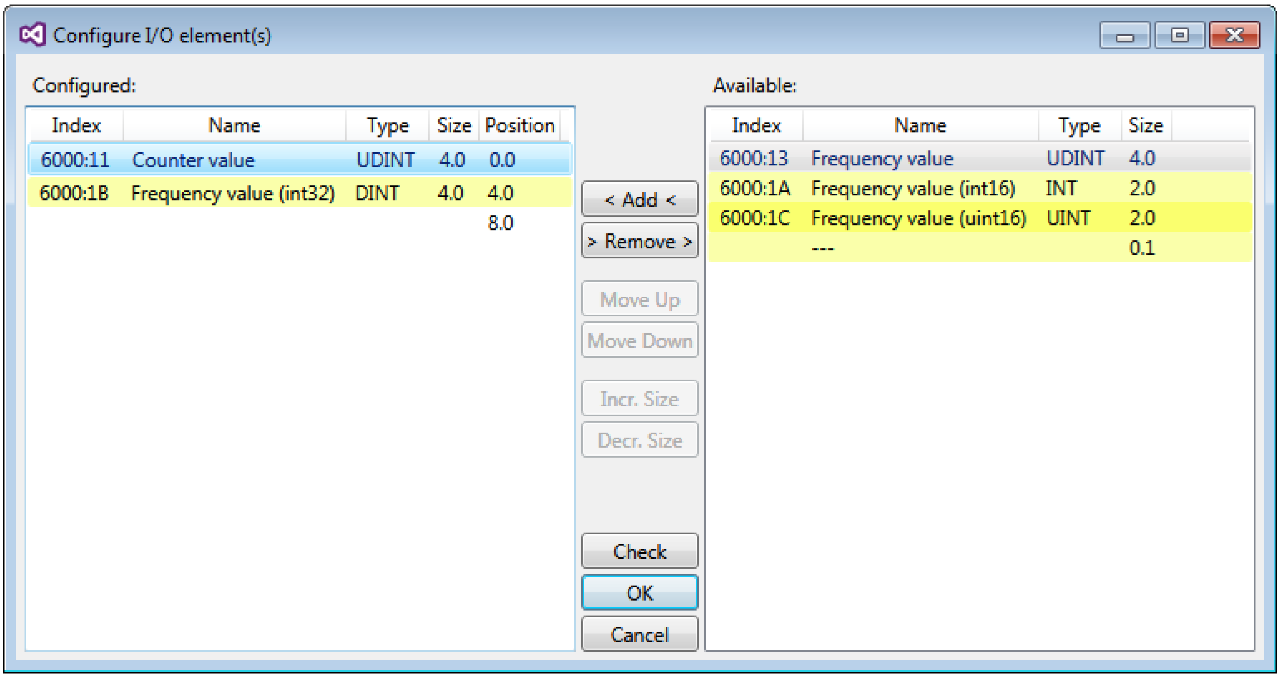


Fig. 66: Selection of the process data

The safety address together with the CRC must be entered on the TwinSAFE SC slave side. This is done via the CoE objects under *TSC settings* of the corresponding TwinSAFE SC component (here, for example, EL5021-0090, 0x8010: 01 and 0x8010: 02). The address set here must also be set in the *alias device* as *FSoE* address under the *Linking* tab.

Under the object 0x80n0:02 Connection Mode the CRC to be used is selected or a free CRC is entered. A total of 8 CRCs are available. A free CRC must start with 0x00ff in the high word.

Fig. 67: CoE objects 0x8010:01 and 0x8010:02

Object *TSC Settings*

Depending on the terminal, the index designation of the configuration object *TSC Settings* can vary. Example:
- EL3214-0090 and EL3314-0090, TSC Settings, Index 8040
- EL5021-0090, TSC Settings, Index 8010
- EL6224-0090, TSC Settings, Index 800F

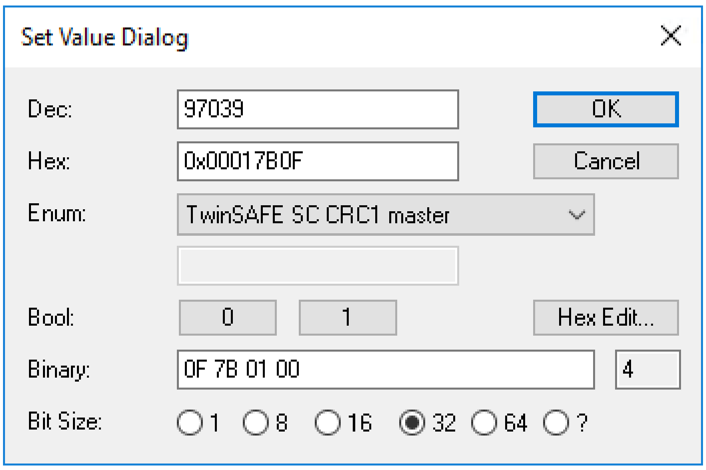


Fig. 68: Entering the safety address and the CRC

TwinSAFE SC connections

If several TwinSAFE SC connections are used within a configuration, a different CRC must be selected for each TwinSAFE SC connection.

11.2 TwinSAFE SC process data EJ6224-0090

The process data size and the process data themselves can be selected under the *Process Image* tab in TwinCAT. The process data depend on the connected IO-Link slave and have to be adjusted accordingly. The corresponding information can be found in the sensor-specific data sheet. By default, a 16-bit unsigned integer is mapped with byte offset 0 from IO-Link channel 1.

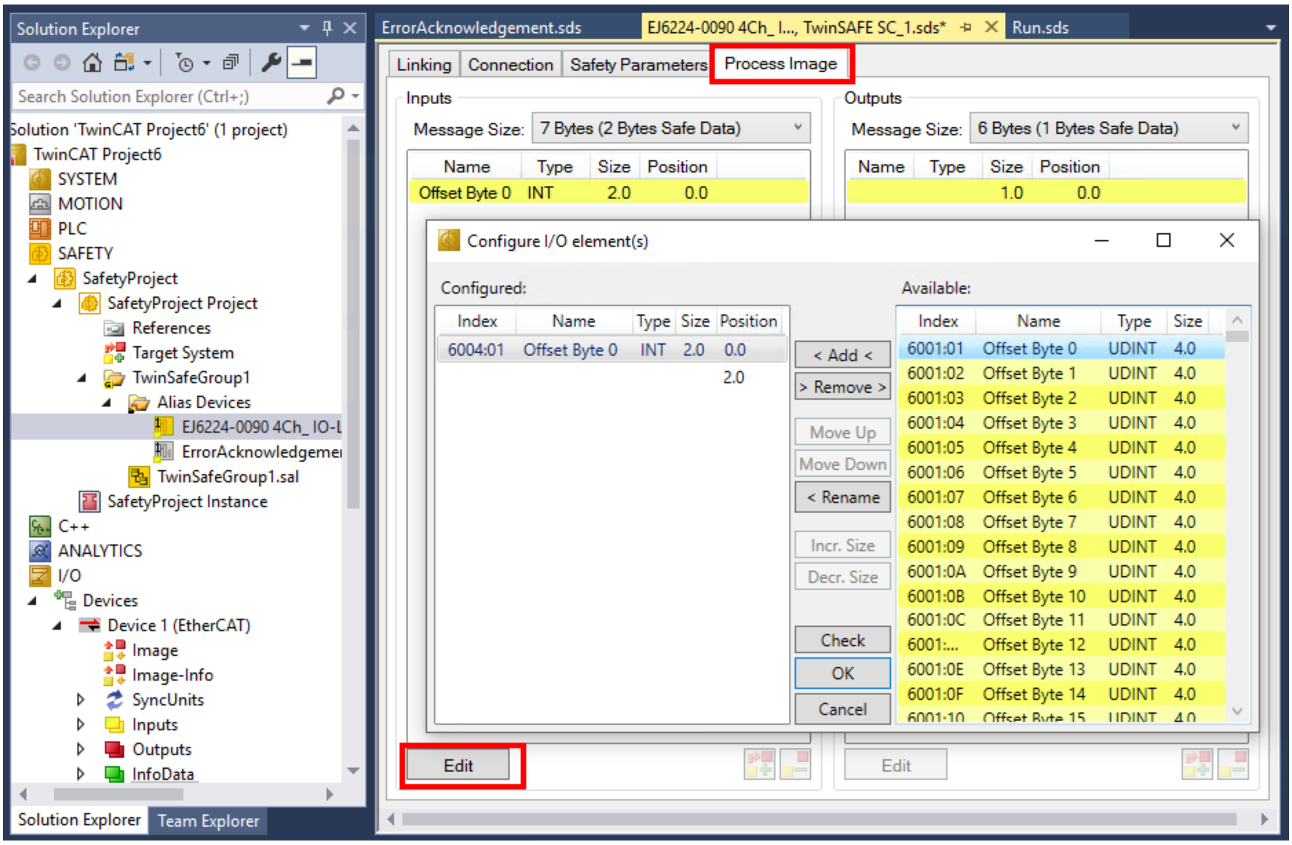


Fig. 69: EJ6224-0090, process data image (default)

The process data can be adjusted to user requirements by selecting the *Edit* button in the dialog *Configure I/O element(s)*.

The following rule applies for process data mapping:

0x60ab:0c		
a = IO-Link port	b = data type integer	c = byte offset
0 = IO-Link port 1	0 = data type bit	01 = offset byte 0
1 = IO-Link port 2	1 = UDINT	02 = offset byte 1
2 = IO-Link port 3	2 = DINT	03 = offset byte 2
3 = IO-Link port 4	3 = UINT	…..
	4 = INT	0p = offset byte p-1

Examples for index mapping:

0x6004:01 => IO-Link port 1, data type INT, offset byte 0

0x6011:01 => IO-Link port 2, data type UDINT, offset byte 0

TwinSAFE SC objects

The TwinSAFE SC objects are listed in the chapter Objects TwinSAFE Single Channel (EJ6224-0090) [▶ 98].

12 Diagnostics

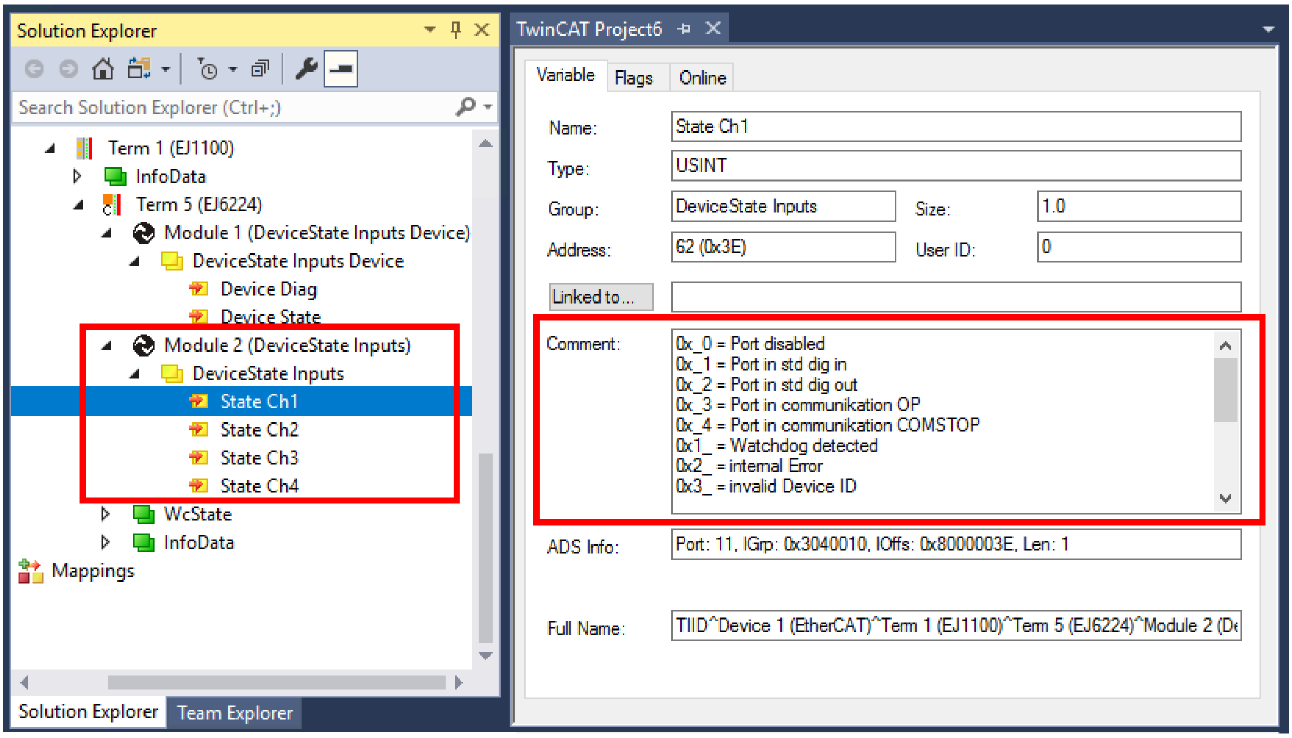
12.1 State of the IO-Link Ports

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 “Module 2 (DeviceState Inputs)”
- In the CoE object 0xF100 [▶ 91]

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

Read the note on the addition of values in case of simultaneously occurring diagnostic messages in the chapter PDO assignment [▶ 91].



Display of the process data object “Module 2 (DeviceState Inputs)” and meaning of the status bytes in the “Comment” field.

12.2 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

12.3 Further error diagnostics

Device State Inputs Device (0x1A81)

It is indicated in the PDO "Device Diag [▸ 68]" (0xF101:0D) that at least one event has occurred in the "Diag History".
"Device State [▸ 68]" 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 field [▸ 106] 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 [▸ 90] 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.

13 Appendix

13.1 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

More Information:
www.beckhoff.com/EJ6xxx

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