BECKHOFF New Automation Technology

Operating manual | EN

ELX3181 and ELX3184

One and four channel analog input terminals, 4 ... 20 mA, single-ended, 16 bit, HART, Ex i

2024-08-23 | Version: 2.2.0

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

Beckhoff®, 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

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

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

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

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

Warning of damage to property or environment

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

Information on handling the product

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

1.3 Documentation Issue Status

Version	Comment
2.2.0	• Chapter *Identification of ELX terminals* updated • Technical data updated • Chapter *Configuration of ELX terminals in bus terminal block* extended • Chapter *Disposal* added • Hyperlinks updated
2.1.0	• ELX3184 added
2.0.0	• Chapter *Basic function principles*, *Parameterization and programming*, *HART* and *Field Device Tool (FDT)* added • Technical data updated • Chapter *Identification of ELX terminals* updated • Chapter *Connection technology* updated • Safety instructions adapted to IEC 82079-1 • New title page
1.4.0	• FM notes regarding ANSI/ISA EX added • Chapter *Identification of ELX terminals* updated
1.3.0	• Connection extended with sensor display • Chapter *Configuration of ELX terminals in bus terminal block* updated • Chapter *Identification of ELX terminals* updated • Technical data updated
1.2.0	• Chapter *Configuration of ELX terminals in bus terminal block* updated
1.1.0	• Complements, corrections
1.0.0	• First release
0.1	• First preliminary version

1.4 Suggestions or proposals for documentation

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

1.5 Marking of ELX terminals

Designation

An ELX terminal has a 15-digit technical designation, composed of

- Family key
- Type
- Software variant
- Revision

Example	Family	Type	Software variant	Revision
ELX1052-0000-0001	ELX terminal	1052: Two-channel digital input terminal for NAMUR sensors, Ex i	0000: Basic type	0001
ELX9560-0000-0001	ELX terminal	9560: Power supply terminal	0000: Basic type	0001

Notes

- The elements mentioned above result in the **technical designation**. ELX1052-0000-0001 is used in the example below.
- Of these, ELX1052-0000 is the order identifier, commonly called just ELX1052 in the "-0000" revision. “-0001” is the EtherCAT revision.
- The **order identifier** is made up of
  - family key (ELX)
  - type (1052)
  - software version (-0000)
- The **Revision** -0001 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 website.
  The revision has been applied to the terminals on the outside, see *Fig. ELX1052 with date code 3218FMFM, BTN 10000100 and Ex marking.*
- The hyphen is omitted in the labeling on the side of the terminal. Example:
  Name: ELX1052-0000
  Label: ELX1052 $_{0000}$
- The type, software version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

Identification numbers

ELX terminals have two different identification numbers:

- date code (batch number)
- **B**eckhoff **T**raceability **N**umber, or BTN for short (as a serial number it clearly identifies each terminal)

Date code

The date code is an eight-digit number given by Beckhoff and printed on the ELX terminal. The date code indicates the build version in the delivery state and thus identifies an entire production batch but does not distinguish between the terminals in a batch.

Structure of the date code: **WW YY FF HH**	Example with date code 02180100:
WW - week of production (calendar week)	02 - week of production 02
YY - year of production	18 - year of production 2018
FF - firmware version	01 - firmware version 01
HH - hardware version	00 - hardware version 00

Beckhoff Traceability Number (BTN)

In addition, each ELX terminal has a unique **B**eckhoff **T**raceability **N**umber (BTN).

Ex marking

The Ex marking can be found at the top left on the terminal:

II 3 (1) G Ex ec [ia Ga] IIC T4 Gc
II (1) D [Ex ia Da] IIIC
I (M1) [Ex ia Ma] I
IECEx BVS 18.0005X
BVS 18 ATEX E 005 X

Examples

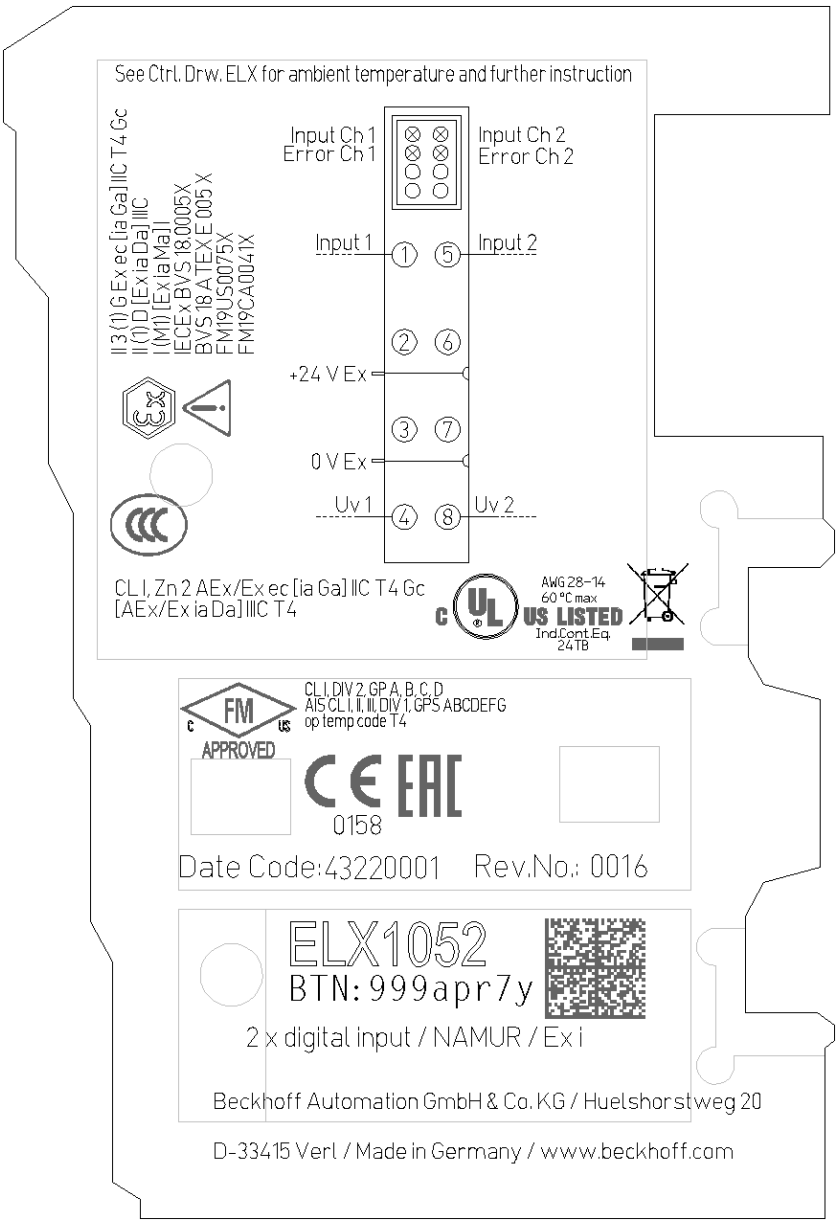


Fig. 1: ELX1052-0000 with date code 43220001, BTN 999apr7y and Ex marking

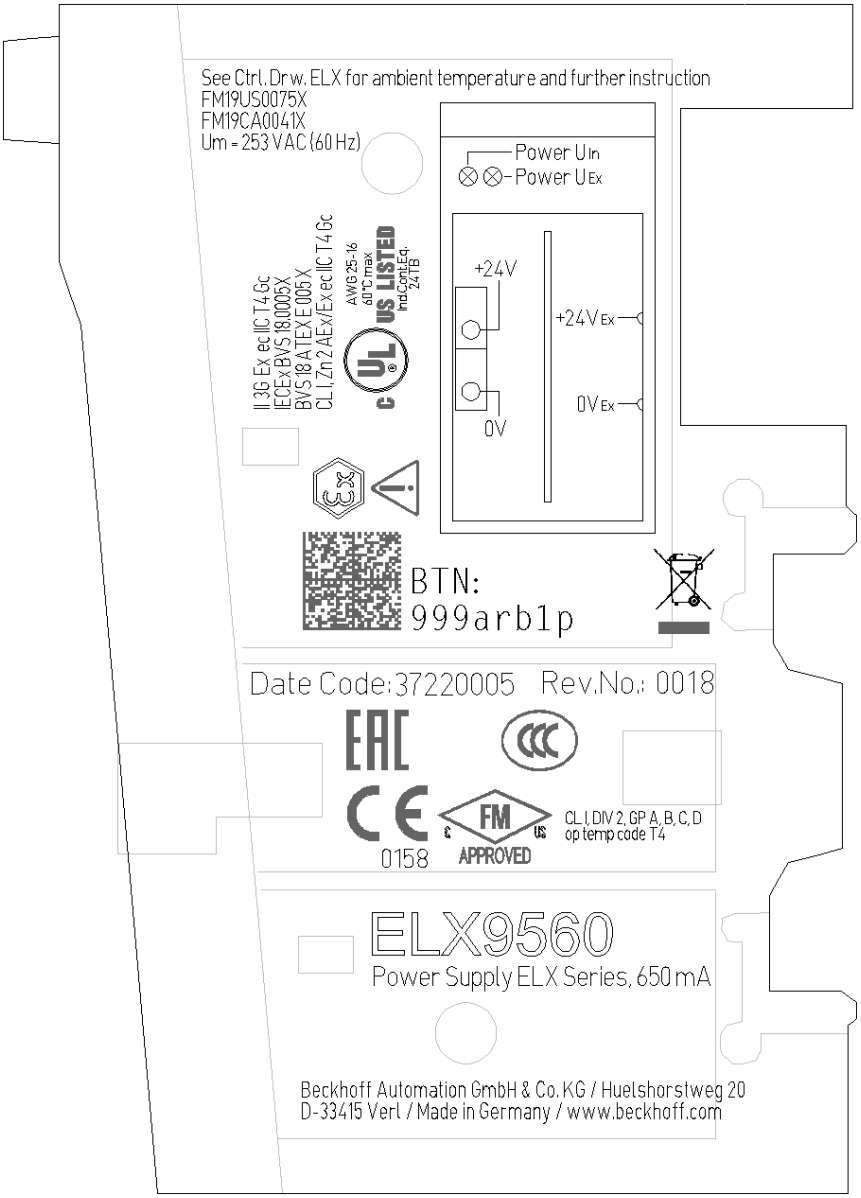


Fig. 2: ELX9560-0000 with date code 37220005, BTN 999arb1p and Ex marking

Fig. 3: ELX9012 with date code 36230000, BTN 000bh4yr and Ex marking

2 Product overview

2.1 ELX3181 - Introduction

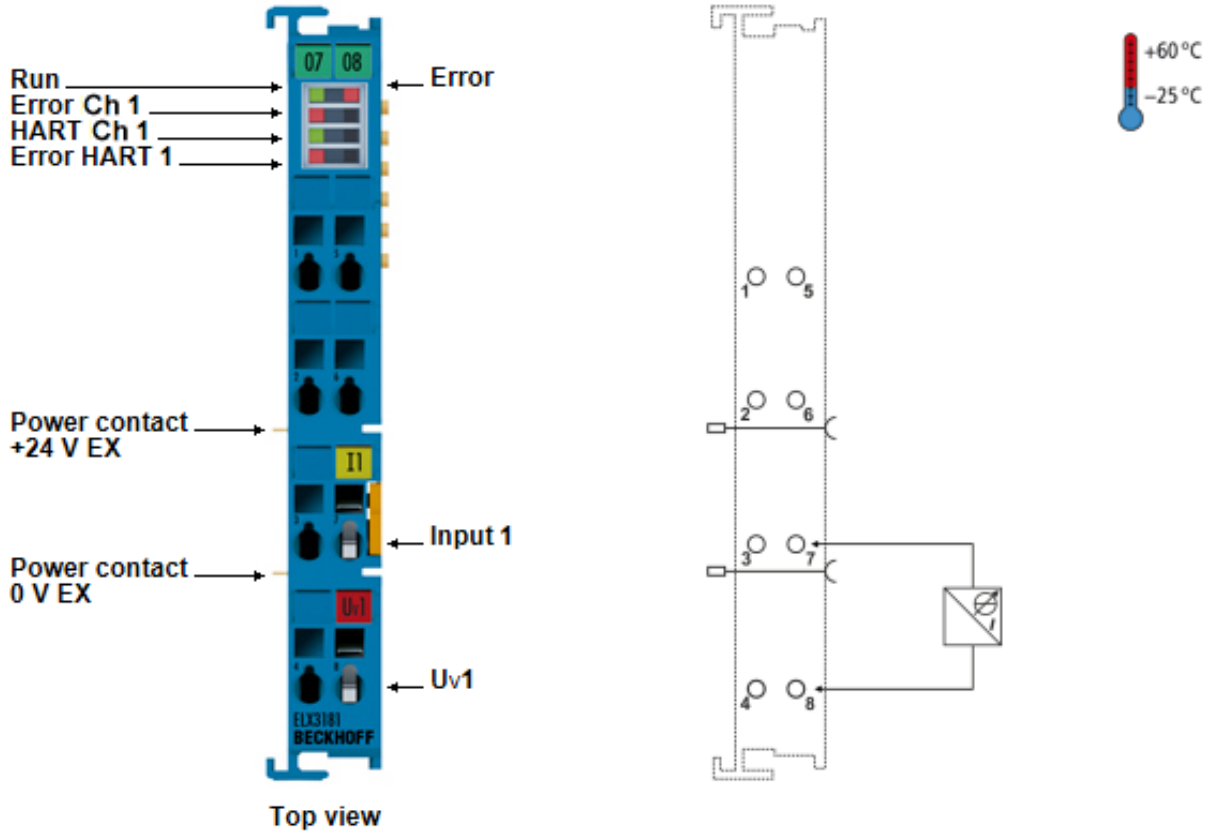


Fig. 4: ELX3181 - 1-channel analog input terminal 4…20 mA, single-ended, 16 bit, HART, Ex i

The ELX3181 analog input terminal for intrinsically safe HART-capable field devices supplies measuring transducers located in the field and transmits their analog measuring signals electrically isolated to the automation device.

The HART (Highway Addressable Remote Transducer) protocol enables two-way communication through digital data transfer via the sensor wiring. The advantages of the simple and robust analog interface (4… 20 mA) are combined with the diagnostic and parameterization options of a digital interface.

The ELX3181 will be supplied via the ELX9560 potential feeding-in terminal. The EtherCAT Terminal indicates the signal state by means of light emitting diodes. Error LEDs signal overload and wire breakage, in addition to which LEDs provide information about the state of the HART communication and signal any communication errors.

2.2 ELX3184 - Introduction

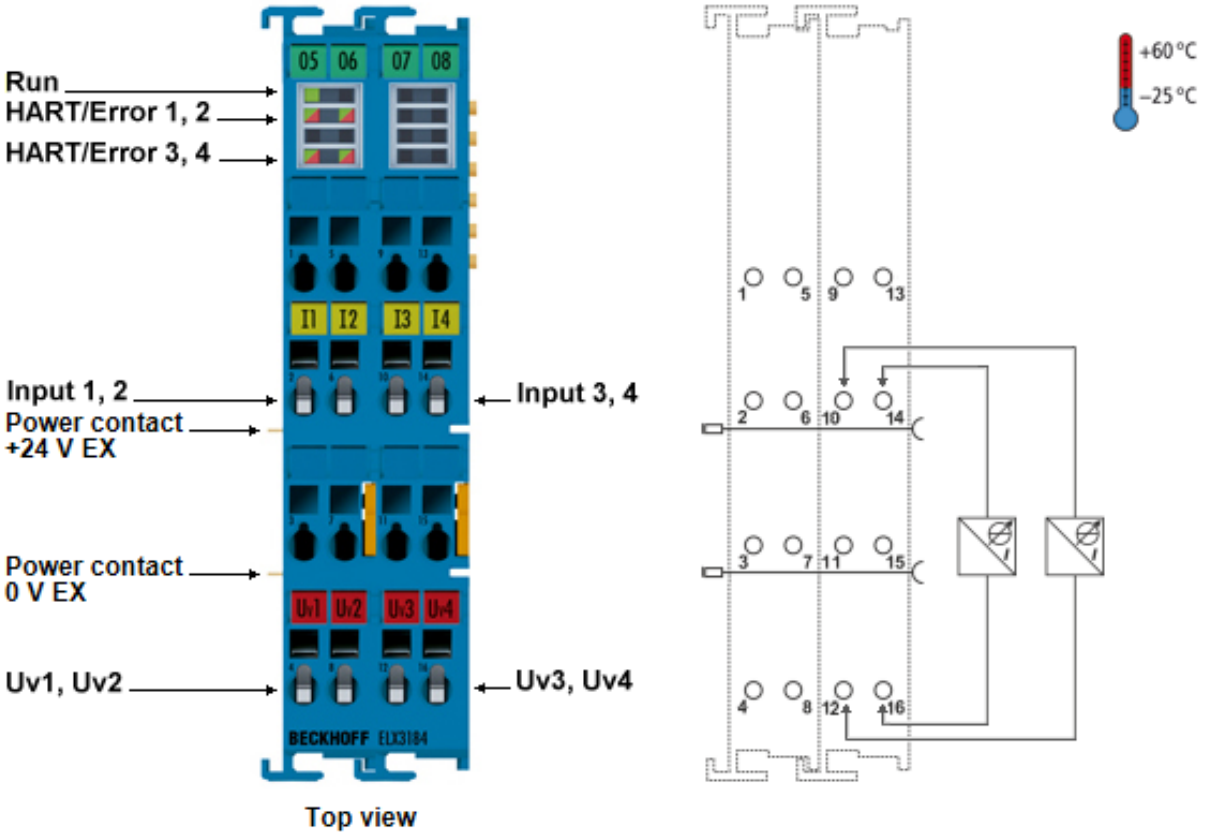


Fig. 5: ELX3184 - 4-channel analog input terminal 4…20 mA, single-ended, 16 bit, HART, Ex i

The ELX3184 analog input terminal for intrinsically safe HART-capable field devices supplies measuring transducers located in the field and transmits their analog measuring signals electrically isolated to the automation device.

The HART (Highway Addressable Remote Transducer) protocol enables two-way communication through digital data transfer via the sensor wiring. The advantages of the simple and robust analog interface (4… 20 mA) are combined with the diagnostic and parameterization options of a digital interface.

The ELX3184 will be supplied via the ELX9560 potential feeding-in terminal. The EtherCAT Terminal indicates the status of the HART communication by means of LEDs and signals any communication errors.

2.3 Technical data

Technical data	ELX3181-0000	ELX3184-0000
Technology	HART	
Number of inputs	1 (single-ended)	4 (single-ended)
Signal current	4 … 20 mA	
Internal resistance	typ. 250 Ω	
Input filter cut-off frequency	25 Hz	
Conversion time	typ. 1 ms	
Resolution	16 bits (including sign)	
Measuring error	< ±0.3 % (relative to full scale value)	
Power supply of the electronics	from the E-bus (5 V_{DC}) and the power contacts (24 V_{DC} Ex, supply by ELX9560)	
Current consumption via E-bus	typ. 85 mA	typ. 60 mA
Current consumption from the power contacts	typ. 15 mA + load	typ. 20 mA + load
Distributed clocks	yes	
Bit width in process image	Inputs: 4 bytes HART: 28 bytes (optional)	Inputs: 4 x 4 bytes HART: 28 bytes per channel (optional)
Configuration	No address settings required, HART setup via TwinCAT	
Special features	- Standard and compact process image - Activatable FIR/IIR filters - Limit value monitoring - NE43 NAMUR	
Weight	approx. 60 g	approx. 90 g
Permissible ambient temperature range during operation	-25 °C ... + 60 °C	
Permissible ambient temperature range during storage	-40 °C ... + 85 °C	
Permissible relative air humidity	95 %, no condensation	
Permissible air pressure (operation, storage, transport)	800 hPa ... 1100 hPa (this is equivalent to an altitude of approx. -690 m to 2000 m above sea level assuming an international standard atmosphere)	
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27	
EMC immunity / emission	conforms to EN 61000-6-2 / EN 61000-6-4	
Protection rating	IP20	
Installation position	See chapter Installation position and minimum distances [▶ 24]	
Approvals / markings*	CE, cULus, CCC, ATEX, IECEx, cFMus	CE, cULus, ATEX, IECEx, cFMus

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

Housing data

Technical data	ELX3181-0000	ELX3184-0000
Design	compact terminal housing with signal LEDs	
Material	Polycarbonate, blue	
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 68 mm (width aligned: 12 mm)	approx. 27 mm x 100 mm x 68 mm (width aligned: 24 mm)
Installation [▶ 25]	on 35 mm mounting rail according to EN 60715 with locking	
Stackable by	double groove-tongue connection	
Labelling	Labeling of the BZxxx series	
Power contacts	2 blade/spring contacts	

ELX3181-0000 - Technical data for explosion protection

Technical data for explosion protection		ELX3181-0000
Ex marking	ATEX	II 3 (1) G Ex ec [ia Ga] IIC T4 Gc II (1) D [Ex ia Da] IIIC I (M1) [Ex ia Ma] I
	IECEx	Ex ec [ia Ga] IIC T4 Gc [Ex ia Da] IIIC [Ex ia Ma] I
	cFMus	AIS Class I, II, III, Division 1, Groups A thru G Class I, Division 2, Groups A, B, C, D Class I, Zone 2, AEx/Ex ec [ia Ga] IIC T4 Gc [AEx/Ex ia Da] IIIC T4
Certificate number		IECEx BVS 18.0005X BVS 18 ATEX E 005 X FM19US0075X, FM19CA0041X
Power supply		without exception in connection with the ELX9560

Use in connection with an ELX9560 from HW05*		ELX3181-0000 from HW02**		ELX3181-0000 to HW01**	
Field interfaces		U_o = 27.0 V I_o = 74 mA P_o = 496 mW Characteristic curve: linear		U_o = 27.0 V I_o = 80 mA P_o = 535 mW Characteristic curve: linear	
Reactances (without consideration of simultaneity)		**L_o**	**C_o**	**L_o**	**C_o**
	Ex ia I	59 mH	3.75 µF	49 mH	3.75 µF
	Ex ia IIA	42 mH	2.33 µF	35 mH	2.33 µF
	Ex ia IIB	25 mH	705 nF	21 mH	705 nF
	Ex ia IIC	3.7 mH	90 nF	2.8 mH	90 nF
	Ex ia IIIC	25 mH	705 nF	21 mH	705 nF

Use in connection with an ELX9560 to HW04*		ELX3181-0000 from HW02**		ELX3181-0000 to HW01**	
Field interfaces		U_o = 27.7 V I_o = 76 mA P_o = 522 mW Characteristic curve: linear		U_o = 27.7 V I_o = 85 mA P_o = 565 mW Characteristic curve: linear	
Reactances (without consideration of simultaneity)		**L_o**	**C_o**	**L_o**	**C_o**
	Ex ia I	55 mH	3.45 µF	43 mH	3.45 µF
	Ex ia IIA	39 mH	2.2 µF	30 mH	2.2 µF
	Ex ia IIB	23 mH	663 nF	18 mH	663 nF
	Ex ia IIC	3.1 mH	85 nF	2 mH	85 nF
	Ex ia IIIC	21 mH	663 nF	18 mH	663 nF

***) Hardware Version of the ELX9560 power supply terminal**

The hardware version of the ELX9560 can be found on the front side of your power supply terminal from hardware version 04 onwards.

****) Hardware Version of the ELX terminal**

The hardware version of the ELX terminal can be found in the date code [▶ 8] on the side of your signal terminal.

ELX3184-0000 - Technical data for explosion protection

Technical data for explosion protection		ELX3184-0000
Ex marking	ATEX	II 3 (1) G Ex ec [ia Ga] IIC T4 Gc II (1) D [Ex ia Da] IIIC I (M1) [Ex ia Ma] I
	IECEx	Ex ec [ia Ga] IIC T4 Gc [Ex ia Da] IIIC [Ex ia Ma] I
	cFMus	AIS Class I, II, III, Division 1, Groups A thru G Class I, Division 2, Groups A, B, C, D Class I, Zone 2, AEx/Ex ec [ia Ga] IIC T4 Gc [AEx/Ex ia Da] IIIC T4
Certificate number		IECEx BVS 18.0005X BVS 18 ATEX E 005 X FM19US0075X, FM19CA0041X
Power supply		without exception in connection with the ELX9560

Use in connection with an ELX9560 from HW05*		ELX3184-0000 from HW01**		ELX3184-0000, HW00**	
Field interfaces		U_o = 27.0 V I_o = 74 mA P_o = 496 mW Characteristic curve: linear		U_o = 27.0 V I_o = 79 mA P_o = 534 mW Characteristic curve: linear	
Reactances (without consideration of simultaneity)		**L_o**	**C_o**	**L_o**	**C_o**
	Ex ia I	59 mH	3.75 µF	49 mH	3.75 µF
	Ex ia IIA	42 mH	2.33 µF	36 mH	2.33 µF
	Ex ia IIB	25 mH	705 nF	20 mH	705 nF
	Ex ia IIC	3.7 mH	90 nF	3 mH	90 nF
	Ex ia IIIC	25 mH	705 nF	20 mH	705 nF

Use in connection with an ELX9560 to HW04*		ELX3184-0000 from HW01**		ELX3184-0000, HW00**	
Field interfaces		U_o = 27.7 V I_o = 76 mA P_o = 522 mW Characteristic curve: linear		U_o = 27.7 V I_o = 81 mA P_o = 561 mW Characteristic curve: linear	
Reactances (without consideration of simultaneity)		**L_o**	**C_o**	**L_o**	**C_o**
	Ex ia I	55 mH	3.45 µF	43 mH	3.45 µF
	Ex ia IIA	39 mH	2.2 µF	34 mH	2.2 µF
	Ex ia IIB	23 mH	663 nF	20 mH	663 nF
	Ex ia IIC	3.1 mH	85 nF	2.4 mH	85 nF
	Ex ia IIIC	23 mH	663 nF	20 mH	663 nF

***) Hardware Version of the ELX9560 power supply terminal**

The hardware version of the ELX9560 can be found on the front side of your power supply terminal from hardware version 04 onwards.

****) Hardware Version of the ELX terminal**

The hardware version of the ELX terminal can be found in the date code [▶ 8] on the side of your signal terminal.

2.4 Intended use

⚠ WARNING

Endangering the safety of persons and equipment!

The ELX components may only be used for the purposes described below!

⚠ CAUTION

Observe ATEX and IECEx!

The ELX components may only be used in accordance with the ATEX directive and the IECEx scheme!

The ELX terminals extend the field of application of the Beckhoff bus terminal system with functions for integrating intrinsically safe field devices from hazardous areas. The intended field of application is data acquisition and control tasks in discrete and process engineering automation, taking into account explosion protection requirements.

The ELX terminals are protected by the type of protection "Increased safety" (Ex e) according to IEC 60079-7 and must only be operated in hazardous areas of Zone 2 or in non-hazardous areas.

The field interfaces of the ELX terminals achieve explosion protection through the type of protection "intrinsic safety" (Ex i) according to IEC 60079-11. For this reason, only appropriately certified, intrinsically safe devices may be connected to the ELX terminals. Observe the maximum permissible connection values for voltages, currents and reactances. Any infringement can damage the ELX terminals and thus eliminate the explosion protection.

The ELX terminals are open, electrical equipment for installation in lockable cabinets, enclosures or operating rooms. Make sure that access to the equipment is only possible for authorized personnel.

⚠ CAUTION

Ensure traceability!

The buyer has to ensure the traceability of the device via the Beckhoff Traceability Number (BTN).

3 Mounting and wiring

3.1 Special conditions of use for ELX terminals

⚠ WARNING

Observe the special conditions of use for the intended use of Beckhoff ELX terminals in potentially explosive areas (ATEX directive 2014/34/EU)!

- The certified components are to be installed in a suitable housing that guarantees an ingress protection of at least IP54 in accordance with EN 60079-0 and EN 60529! The prescribed environmental conditions during installation, operation and maintenance are thereby to be taken into account! Inside the housing, pollution degree 1 and 2 are permissible.
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of -25 to +60°C of Beckhoff ELX terminals!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages! The power supply of the ELX9560 power supply terminal must correspond to overvoltage category II according to EN 60664-1
- The individual terminals may only be unplugged or removed from the bus terminal system if all supply voltages have been switched off or if a non-explosive atmosphere is ensured!
- The connections of the ELX9560 power supply terminal may only be connected or disconnected if all supply voltages have been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and switches may only be adjusted if all supply voltages have been switched off or if a non-explosive atmosphere is ensured!

3.2 Installation notes for ELX terminals

NOTICE

Storage, transport and mounting

- Transport and storage are permitted only in the original packaging!
- Store in a dry place, free from vibrations.
- A brand new ELX terminal with a certified build version is delivered only in a sealed carton. Therefore, check that the carton and all seals are intact before unpacking.
- Do not use the ELX terminal if
  - its packaging is damaged
  - the terminal is visibly damaged or
  - you cannot be sure of the origin of the terminal.
- ELX terminals with a damaged packaging seal are regarded as used.

⚠ WARNING

Observe the accident prevention regulations

During mounting, commissioning, operation and maintenance, adhere to the safety regulations, accident prevention regulations and general technical rules applicable to your devices, machines and plants.

⚠ CAUTION

Observe the erection regulations

Observe the applicable erection regulations.

NOTICE

Protect the terminals against electrostatic discharge (ESD)

Electronic components can be destroyed by electrostatic discharge. Therefore, take the safety measures to protect against electrostatic discharge as described in DIN EN 61340-5-1 among others. In conjunction with this, ensure that the personnel and surroundings are suitably earthed.

NOTICE

Do not place terminals on E-bus contacts

Do not place the ELX terminals on the E-bus contacts located on the right-hand side. The function of the E-bus contacts can be negatively affected by damage caused by this, e.g. scratches.

NOTICE

Protect the terminals against dirt

To ensure the functionality of the ELX terminals they must be protected against dirt, especially on the contact points. For this reason use only clean tools and materials.

NOTICE

Handling

- It is forbidden to insert conductive or non-conductive objects of any kind into the interior of the housing (e.g. through the ventilation slots in the housing).
- Use only the openings provided in the housing front and appropriate tools to actuate the spring-loaded terminal contacts on the front side for attaching connection cables to the terminal; see chapter Wiring [▶ 28].
- The opening of the housing, the removal of parts and any mechanical deformation or machining of an ELX terminal are not permitted!

If an ELX terminal is defective or damaged it must be replaced by an equivalent terminal. Do not carry out any repairs to the devices. For safety reasons repairs may only be carried out by the manufacturer.

NOTICE

Contact marking and pin assignment

The colored inscription labels above the front connection contacts shown in the illustrations in the introduction chapter are only examples and are not part of the scope of delivery!
A clear assignment of channel and terminal designation according to the chapter contact assignment to the actual terminal point can be made via the lasered channel numbers 1 to 8 on the left above the respective terminal point as well as via the laser image.
Observe any possible polarity dependency of connected intrinsically safe circuits!

3.3 Arrangement of ELX terminals within a bus terminal block

⚠ WARNING

Observe the following notes on the configuration of ELX terminals!

- ELX signal terminals may only be mounted behind an ELX9560 power supply terminal without exception!
- Only signal terminals from the ELX series may be installed behind an ELX9560 power supply terminal!
- Several ELX9560 power supply terminals may be set in a terminal block as long as an ELX9410 is set before each additional ELX9560!
- An ELX9410 power supply terminal must not be mounted to the right of an ELX9560 or to the left of an ELX signal terminal!
- The last terminal of each ELX terminal segment must be covered with an ELX9012 bus cap or an EK1110 EtherCAT extension, unless two ELX9410 power supply terminals are installed directly behind each other in order to continue the terminal segment with standard Beckhoff EtherCAT Terminals (e.g. EL/ES/EK)!

Examples for the arrangement of ELX terminals

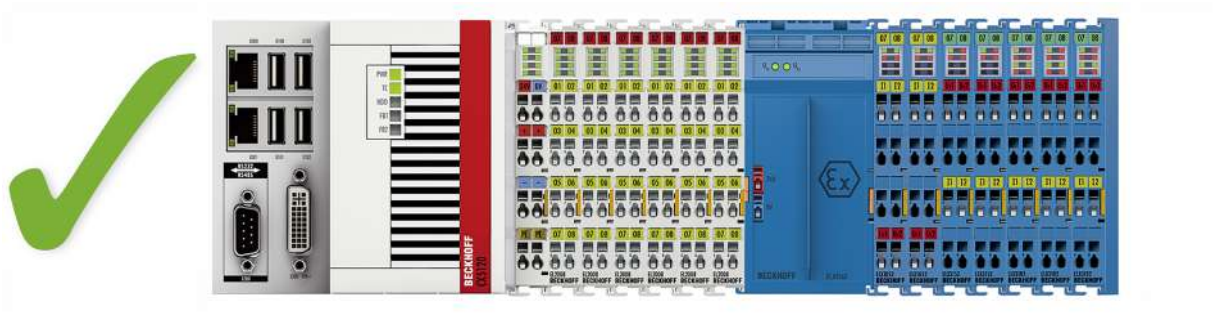


Fig. 6: Permissible arrangement of the ELX terminals (right terminal block).

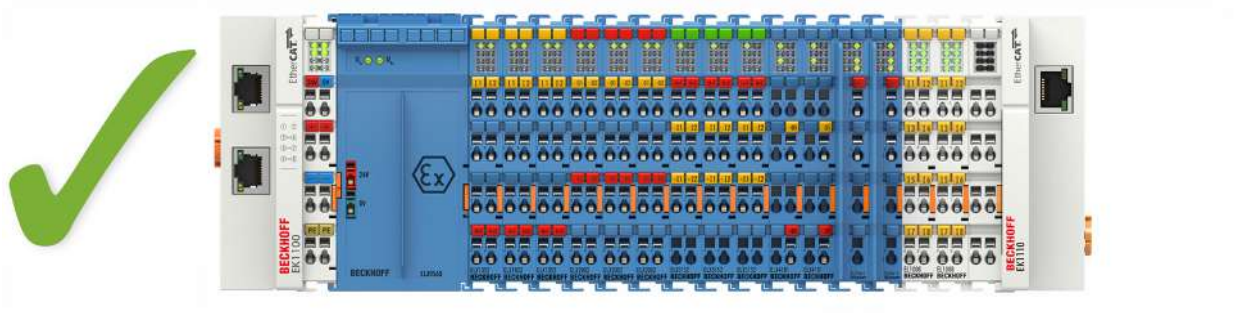


Fig. 7: Permitted arrangement - terminals that do not belong to the ELX series are placed before and after the ELX terminal segment. Isolation is provided by the ELX9560 at the beginning of the ELX terminal segment and two ELX9410 at the end of the ELX terminal segment.

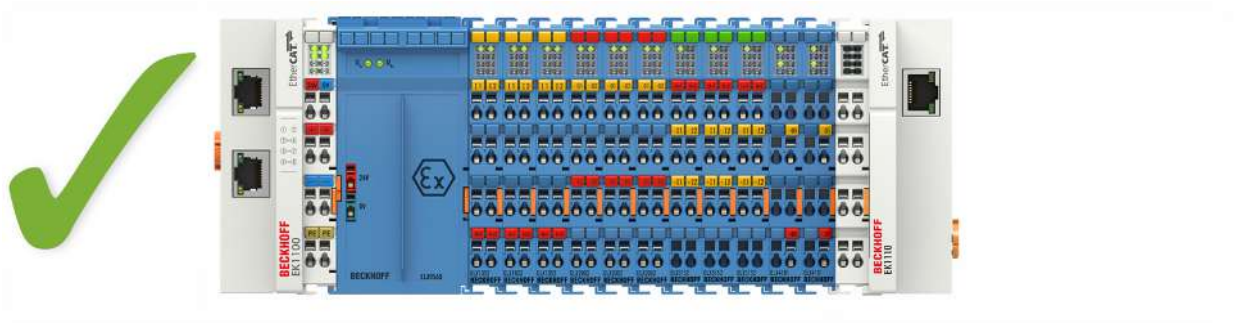


Fig. 8: Permitted arrangement - terminals that do not belong to the ELX series are placed before and after the ELX terminal segment. Isolation is provided by the ELX9560 at the beginning of the ELX terminal segment and the EK1110 at the end of the ELX terminal segment.

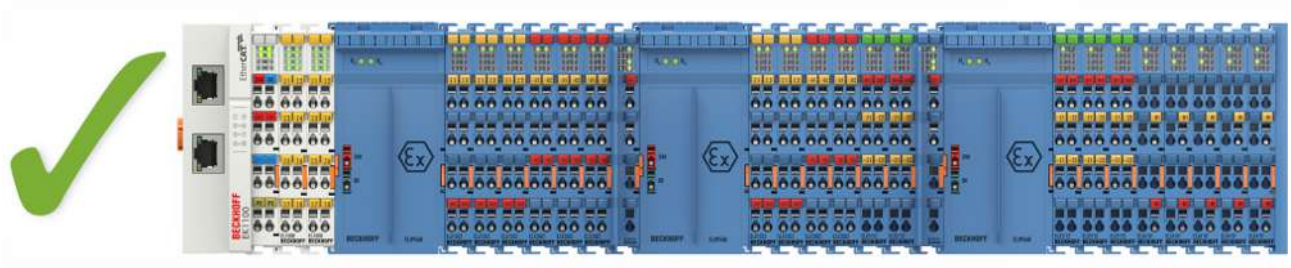


Fig. 9: Permitted arrangement - multiple resupplies by ELX9560 with an upstream ELX9410 in each case.

Fig. 10: Permitted arrangement - ELX9410 in front of an ELX9560 power supply terminal.

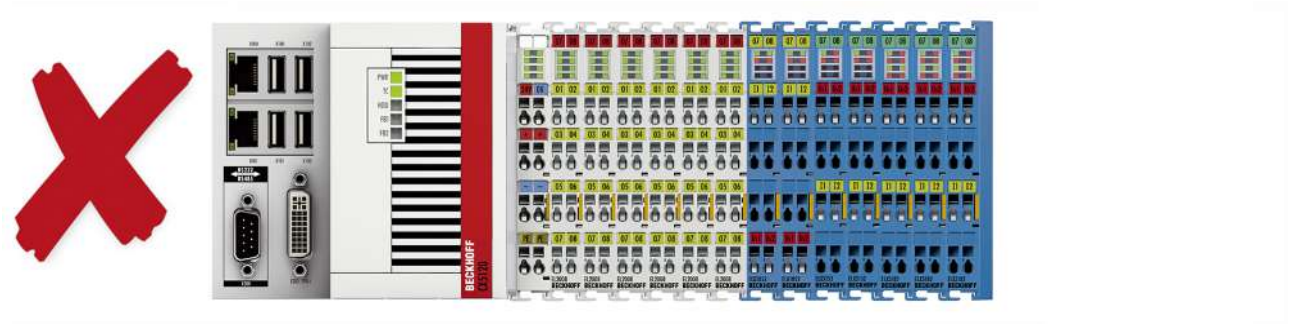


Fig. 11: Illegal arrangement - missing ELX9560 power supply terminal.

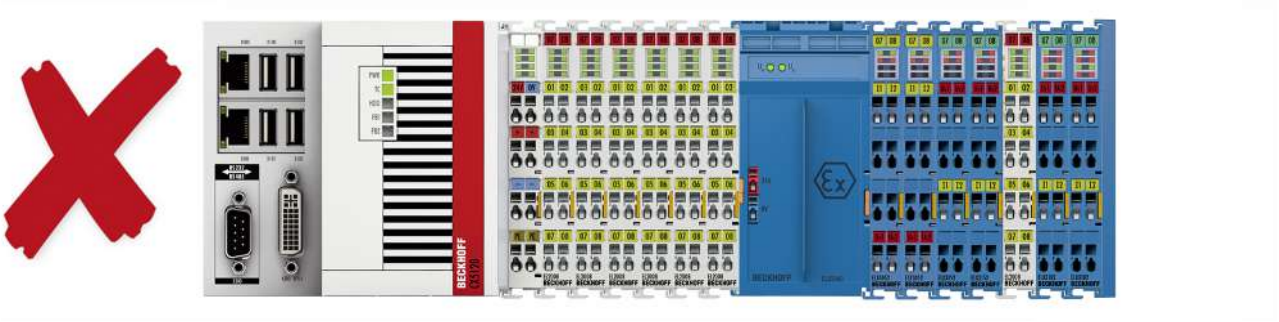


Fig. 12: Impermissible arrangement - terminal in the ELX terminal segment that does not belong to the ELX series

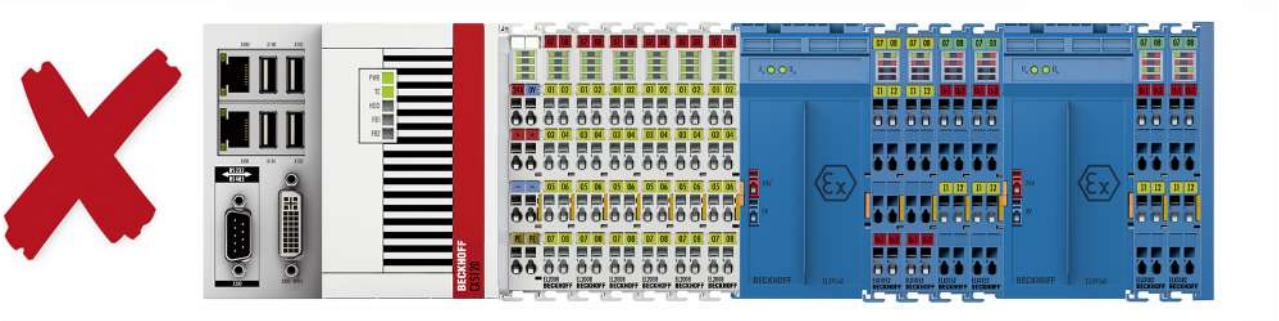


Fig. 13: Impermissible arrangement - second ELX9560 power supply terminal in the ELX terminal segment without upstream ELX9410.

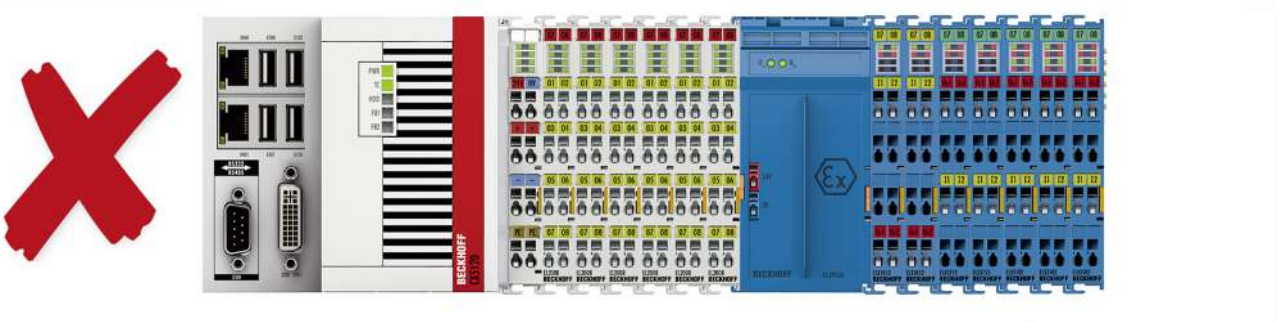


Fig. 14: Illegal arrangement - missing ELX9012 bus end cap.

NOTICE

Note the maximum output current of the ELX9560

When configuring the terminal segment, please observe the maximum available output current of the ELX9560 power supply terminal according to the specified technical data.
If necessary, an additional ELX9560 power supply terminal with upstream ELX9410 (see installation examples) must be installed or a completely new bus terminal block must be configured.

3.4 Installation position and minimum distances

Installation position

For the prescribed installation position the mounting rail is installed horizontally and the mating surfaces of the ELX terminals point toward the front (see illustration below). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. The direction indication “down” corresponds to the direction of positive acceleration due to gravity.

Minimum distances

Observe the following minimum distances to ensure optimum convection cooling:

- above and below the ELX terminals: 35 mm (required!)
- besides the bus terminal block: 20 mm (recommended)

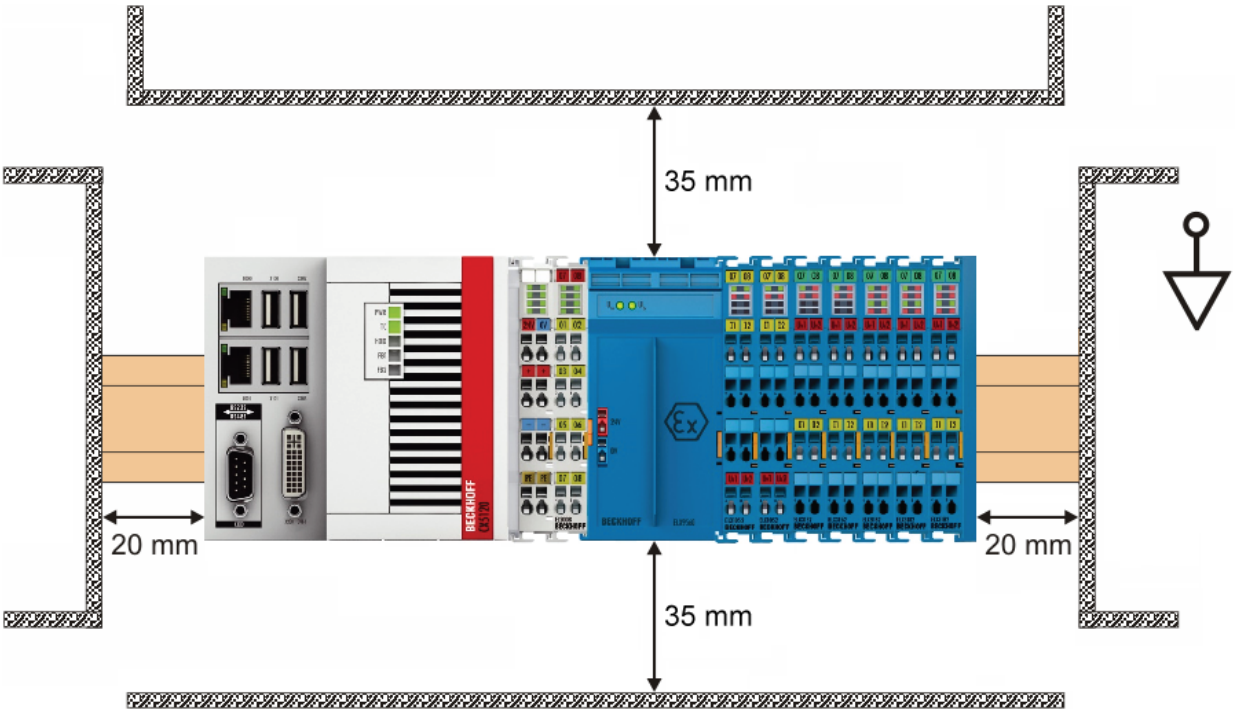


Fig. 15: Installation position and minimum distances

⚠ WARNING

Observe the minimum separation distances according to IEC 60079-14!

Observe the prescribed minimum separation distances between intrinsically safe and non-intrinsically safe circuits according to IEC 60079-14.

3.5 Installation of ELX terminals on mounting rails

⚠ WARNING

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the bus terminals!

⚠ CAUTION

Danger of injury due to power contacts!

For your own protection, pay attention to careful and careful handling of the ELX terminals. In particular, the left side mounted, sharp-edged blade contacts pose a potential risk of injury.

Assembly

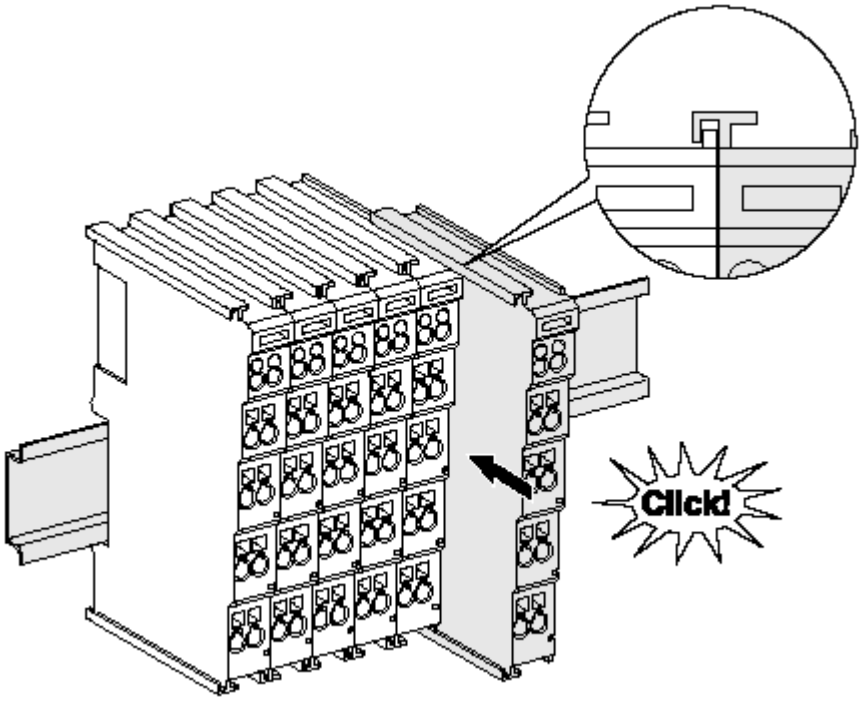


Fig. 16: Attaching on mounting rail

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

1. First attach the fieldbus coupler to the mounting rail.
2. The bus terminals are now attached on the right-hand side of the fieldbus coupler. Join the components with tongue and groove and push the terminals against the mounting rail, until the lock clicks onto the mounting rail.
   If the terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

Fixing of mounting rails

The locking mechanism of the terminals and couplers extends to the profile of the mounting rail. At the installation, the locking mechanism of the components must not come into conflict with the fixing bolts of the mounting rail. To mount the mounting rails with a height of 7.5 mm under the terminals and couplers, you should use flat mounting connections (e.g. countersunk screws or blind rivets).

Disassembly

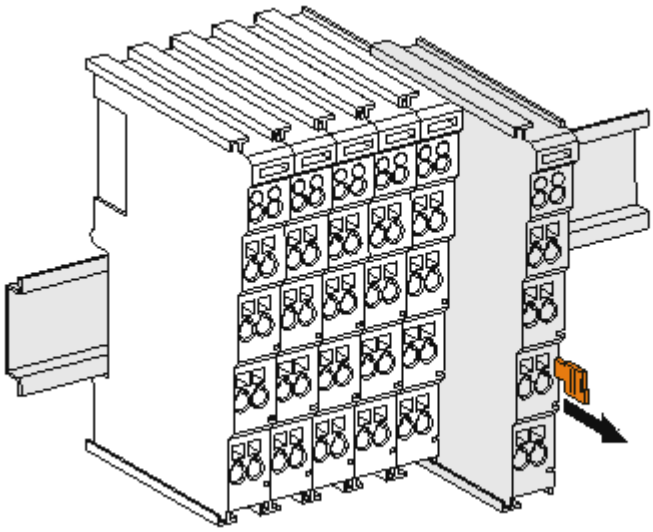


Fig. 17: Disassembling of terminal

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

1. Pull the terminal by its orange-colored lugs approximately 1 cm away from the mounting rail. In doing so for this terminal the mounting rail lock is released automatically and you can pull the terminal out of the bus terminal block easily without excessive force.
2. Grasp the released terminal with thumb and index finger simultaneous at the upper and lower grooved housing surfaces and pull the terminal out of the bus terminal block.

Connections within a bus terminal block

The electric connections between the Bus Coupler and the Bus Terminals are automatically realized by joining the components:

- The six spring contacts of the E-Bus deal with the transfer of the data and the supply of the Bus Terminal electronics.
- The power contacts deal with the supply for the field electronics and thus represent a supply rail within the bus terminal block.
  The power contacts of the ELX terminals are supplied by the ELX9560 power terminal. This interrupts the power contacts and thus represents the beginning of a new supply rail.

Power Contacts

During the design of a bus terminal block, the pin assignment of the individual Bus Terminals must be taken account of, since some types (e.g. analog Bus Terminals or digital 4-channel Bus Terminals) do not or not fully loop through the power contacts.

3.6 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.

3.7 Connection

3.7.1 Connection system

⚠ WARNING

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the bus terminals!

The terminals of ELXxxxx series include electronics and connection level in a single enclosure.

Standard wiring

Fig. 18: Standard wiring

The terminals of ELXxxxx series feature integrated screwless spring force technology for fast and simple assembly.

High Density Terminals (HD Terminals)

Fig. 19: *High Density Terminals*

The Bus Terminals from these series with 16 connection points are distinguished by a particularly compact design, as the packaging density is twice as large as that of the standard 12 mm Bus Terminals. Massive conductors and conductors with a wire end sleeve can be inserted directly into the spring loaded terminal point without tools.

Ultrasonically "bonded" (ultrasonically welded) conductors

Ultrasonically “bonded" conductors

It is also possible to connect the Standard and High Density Terminals with ultrasonically "bonded" (ultrasonically welded) conductors. In this case, please note the tables concerning the wire-size width below!

3.7.2 Wiring

⚠ WARNING

Risk of electric shock and damage of device!

Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the bus terminals!

Terminals for standard wiring

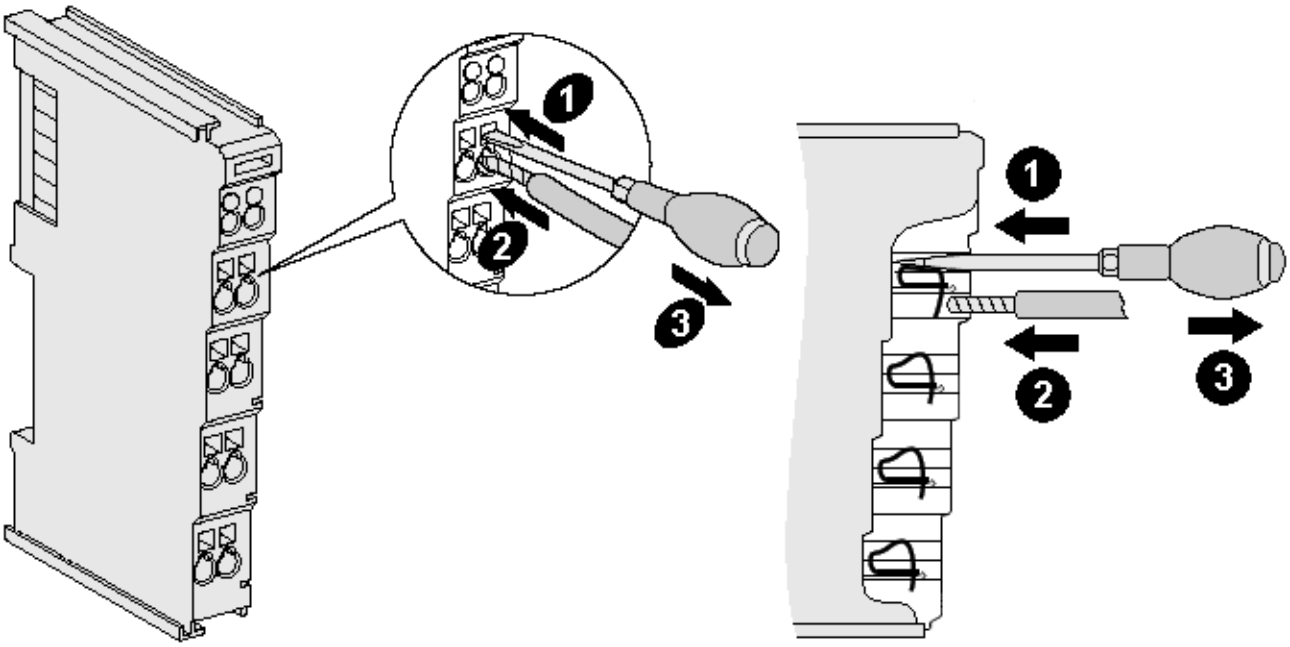


Fig. 20: Connecting a cable on a terminal point

Up to eight terminal points enable the connection of solid or finely stranded cables to the Bus Terminal. The terminal points are implemented in spring force technology. Connect the cables as follows:

1. Open a terminal point by pushing a screwdriver straight against the stop into the square opening above the terminal point. Do not turn the screwdriver or move it alternately (don't toggle).
2. The wire can now be inserted into the round terminal opening without any force.
3. The terminal point closes automatically when the pressure is released, holding the wire securely and permanently.

Observe the requirements for connecting cables and cross sections according to IEC 60079-7 and IEC 60079-11. See the following tables for the suitable wire size width.

Terminal housing	Standard wiring	ELX9560
Wire size width (single core wires)	0.08 ... 2.5 mm²	0.14 ... 1.5 mm²
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm²	0.14 ... 1.5 mm²
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm²	0.14 ... 1.0 mm²
Wire stripping length	8 ... 9 mm	8 ... 9 mm

NOTICE

Maximum screwdriver width for ELX9560

Use a screwdriver with a maximum width of 2 mm to wire the ELX9560 power supply terminal. Wider screwdrivers can damage the terminal points.

High Density Terminals (HD Terminals) with 16 terminal points

The conductors of the HD Terminals are connected without tools for single-wire conductors using the direct plug-in technique, i.e. after stripping the wire is simply plugged into the terminal point. The cables are released, as usual, using the contact release with the aid of a screwdriver. See the following table for the suitable wire size width.

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm^2
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm^2
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm^2
Wire size width (ultrasonically “bonded" conductors)	only 1.5 mm^2
Wire stripping length	8 ... 9 mm

3.7.3 Proper line connection

Always connect only one wire per terminal point.

When using fine-wire conductors it is recommended to connect them with wire end sleeves in order to establish a safe, conductive connection.

In addition, make sure that the pin assignment is correct to prevent damage to the ELX terminals and the connected devices.

3.7.4 Shielding and potential separation

Shielding

Encoder, analog sensors and actors should always be connected with shielded, twisted paired wires!

⚠ CAUTION

Observe installation requirements in areas of potentially explosive atmospheres!

During installation, observe the requirements for cables, shielding and earth potential equalization in areas of potentially explosive atmospheres according to IEC 60079-11, IEC 60079-14 and IEC 60079-25!

⚠ WARNING

Ensure potential separation of the 24 V Ex busbar!

In any case, make sure that the galvanic isolation made by the ELX9560 between the 24 V Ex busbar (power contacts +24 V Ex and 0 V Ex) and other system potentials (if applicable also functional or protective earths) is not removed!

3.7.5 ELX3181 - Contact assignment and LEDs

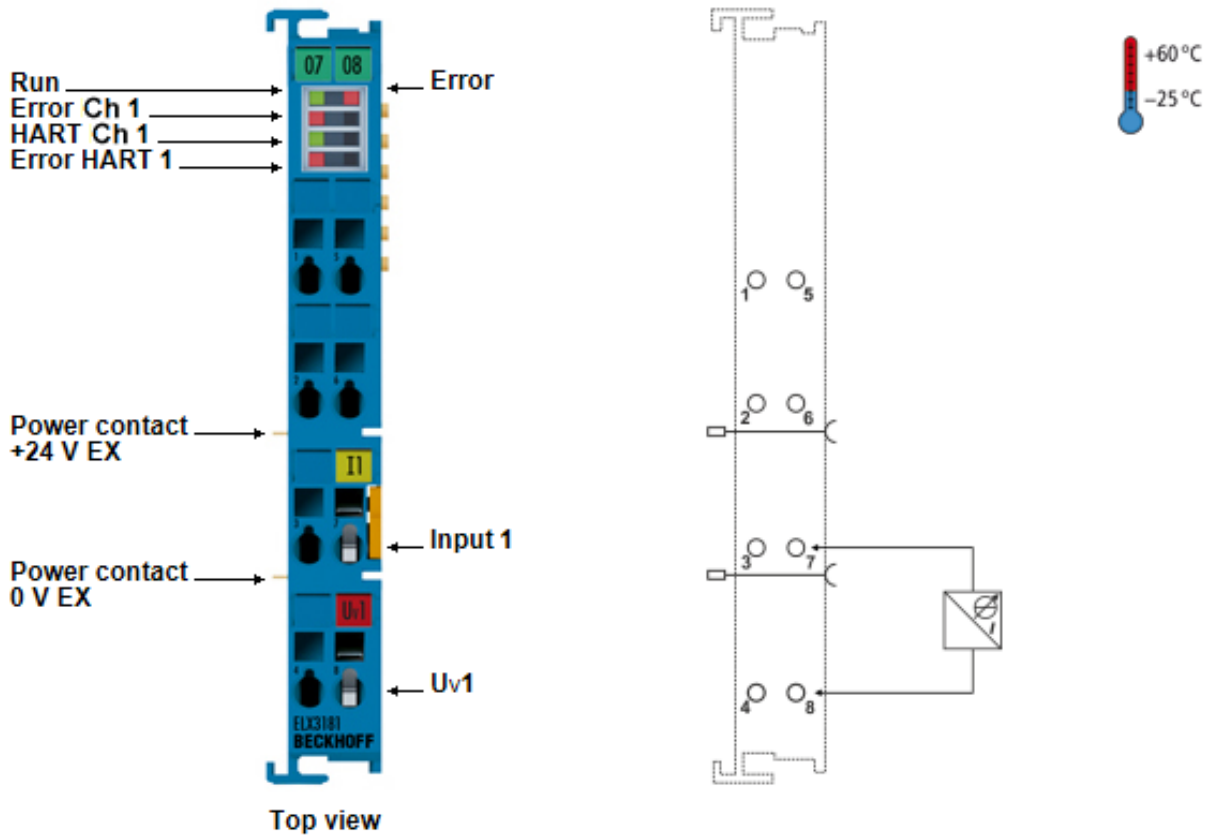


Fig. 21: ELX3181 - Contact assignment

Terminal point		Description
Name	No.	
	1	not implemented
	2	not implemented
	3	not implemented
	4	not implemented
	5	not implemented
	6	not implemented
Input 1	7	signal input Channel 1
Uv1	8	supply voltage channel 1

ELX3181 - LED display

LED	Color	Meaning	
Run	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal or **BOOTSTRAP** = function for firmware updates of the terminal
		flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync-Managers channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
Error	red	General error of the A/D converter	
Error Ch1	red	Fault indication for broken wire and if the measuring range for channel 1 is exceeded (under- or overrun)	
HART Ch1	green	off	No HART communication
		on	HART communication active
Error HART 1	red	HART communication error	

3.7.6 ELX3184 - Contact assignment

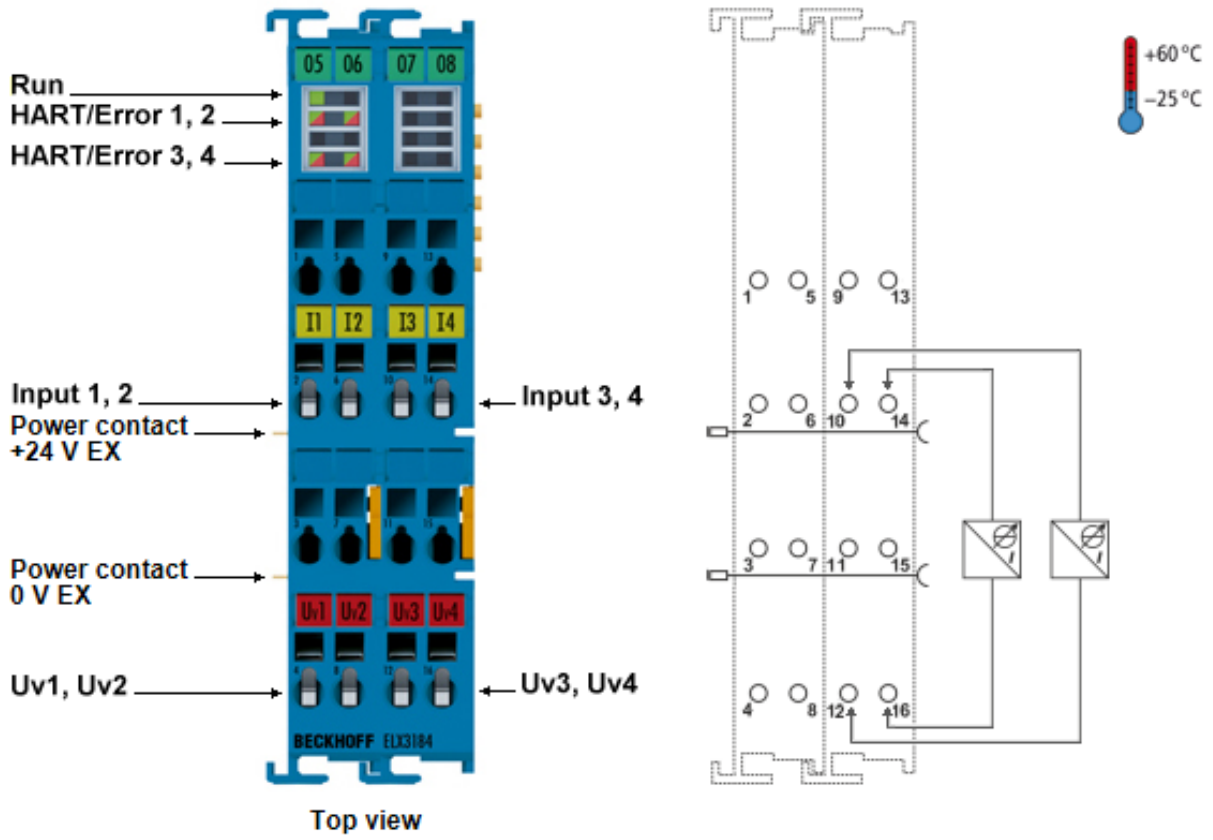


Fig. 22: ELX3184 - Contact assignment

Terminal point		Description
Name	No.	
	1	not implemented
Input 1	2	signal input channel 1
	3	not implemented
Uv1	4	supply voltage channel 1
	5	not implemented
Input 2	6	signal input channel 2
	7	not implemented
Uv2	8	supply voltage channel 2
	9	not implemented
Input 3	10	signal input channel 3
	11	not implemented
Uv3	12	supply voltage channel 3
	13	not implemented
Input 4	14	signal input channel 4
	15	not implemented
Uv4	16	supply voltage channel 4

ELX3184 - LED display

LED	Color	Meaning	
Run	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal or **BOOTSTRAP** = function for firmware updates of the terminal
		flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync-Managers channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
HART Ch n	green	off	No HART communication
		on	HART communication active
Error HART n	red	HART communication error	

4 Basic function principles

4.1 EtherCAT basics

For basic information about the EtherCAT fieldbus please refer to the EtherCAT system documentation, which is also available as PDF file at www.beckhoff.com in the download area of your EtherCAT device.

4.2 Notices on analog specifications

Beckhoff I/O devices (terminals, boxes, modules) with analog inputs are characterized by a number of technical characteristic data; refer to the technical data in the respective documents.

Some explanations are given below for the correct interpretation of these characteristic data.

4.2.1 Full scale value (FSV)

An I/O device with analog input measures over a nominal measuring range, which is limited by an upper and a lower limit (start value and end value), which can usually already be taken from the device designation. The range between both limits is called measuring span and corresponds to the formula (end value - start value). Analogous to pointing devices this is the measuring scale (see IEC 61131) or also the dynamic range.

For analog I/O devices from Beckhoff, the full scale value (FSV) of the respective product (also: reference value) is selected as the largest limit of the nominal measuring range and is given a positive sign. This applies to both symmetrical and asymmetrical measuring spans.

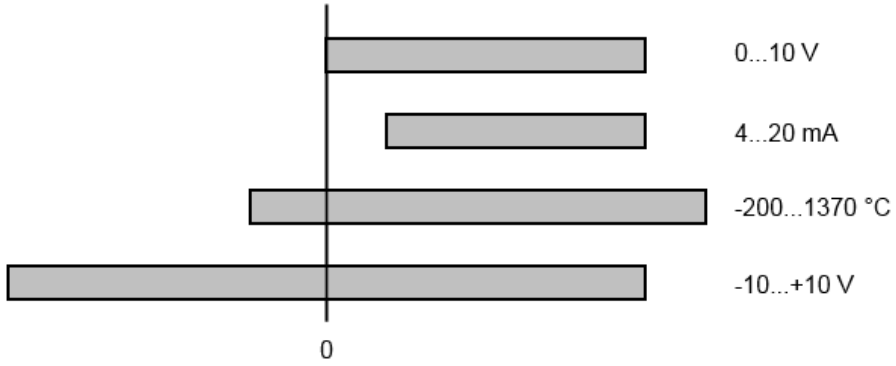


Fig. 23: Full scale value, measuring span

For the above **examples** this means:

- Measuring range 0...10 V: asymmetric unipolar, FSV = 10 V, measuring span = 10 V
- Measuring range 4...20 mA: asymmetric unipolar, FSV = 20 mA, measuring span = 16 mA
- Measuring range -200...1370 °C: asymmetric bipolar, FSV = 1370 °C, measuring span = 1570 °C
- Measuring range -10...+10 V: symmetric bipolar, FSV = 10 V, measuring span = 20 V

Depending on the functionality, an analog input channel may have a technical measuring range that exceeds the nominal measuring range, e.g. to gain more diagnostic information about the signal.
The case-by-case information in the device documentation on the behavior outside the nominal measuring range (measurement uncertainty, display value) must be observed.

The above thoughts are correspondingly applicable to analog output devices:

- The full scale value (FSV) becomes the output end value
- Here, too, there can be a (larger) technical output range in addition to the nominal output range

4.2.2 Measuring error/ measurement deviation

Analog output

The following information also applies analogously to the output end value of analog output devices.

The relative measuring error as a specification value of a Beckhoff analog device is specified in % of the nominal FSV (output end value) and calculated as the quotient of the numerically largest probable deviation from the true measured value (output value) with respect to the FSV (output end value):

$$\text{Measuring error} = \frac{\left|\text{max. deviation}\right|}{\text{full scale value}}$$

It should be noted here that the "true measured value" cannot be determined with infinite accuracy either, but can only be determined via reference devices with a higher expenditure of technology and measuring time and thus a significantly lower measurement uncertainty.

The value therefore describes the result window in which the measured value determined by the device under consideration (Beckhoff analog device) lies with a very high probability in relation to the "true value". Thus, colloquially, this is a "typical" value (typ.); this expresses that the vast statistical majority of values will be within the specification window, but in rare cases there may/will be deviations outside the window.

For this reason, the term "measurement uncertainty" has become established for this window, since "error" is now used to refer to known disturbance effects that can generally be systematically eliminated.

The uncertainty of measurement must always be considered in relation to potential environmental influences:

- invariable electrical channel properties such as temperature sensitivity,
- variable settings of the channel (noise via filters, sampling rate, ...).

Measurement uncertainty specifications without further operational limitation (also called "service error limit") can be assumed as a value "over everything": entire permissible operating temperature range, default setting, etc.

The window is always to be understood as a positive/negative span with "±", even if occasionally indicated as a "half" window without "±".

The maximum deviation can also be specified directly.

Example: measuring range 0...10 V (FSV = 10 V) and measurement uncertainty < ± 0.3%$_{FSV}$ → the expected maximum usual deviation is ± 30 mV in the permissible operating temperature range.

Lower measurement uncertainty possible

If this specification also includes the temperature drift, a significantly lower measuring error can usually be assumed in case of a constant ambient temperature of the device and thermal stabilization after a user calibration.

4.2.3 Temperature coefficient tK [ppm/K]

An electronic circuit is usually temperature dependent to a greater or lesser degree. In analog measurement technology this means that when a measured value is determined by means of an electronic circuit, its deviation from the “true” value is reproducibly dependent on the ambient/operating temperature.

A manufacturer can alleviate this by using components of a higher quality or by software means.

The temperature coefficient, when indicated, specified by Beckhoff allows the user to calculate the expected measuring error outside the basic accuracy. The basic accuracy is usually specified for 23 °C ambient temperature, in special cases also at other temperature.

Due to the extensive uncertainty considerations that are incorporated in the determination of the basic accuracy, Beckhoff recommends a quadratic summation.

Example: Let the basic accuracy be ±0.01% typ. (full scale value), tK = 20 ppm/K typ at 23 °C.; the accuracy A35 at 35 °C is wanted, hence ΔT = 12 K:

$$G35 = \sqrt{(0.01\%)^2 + (12K \cdot 20 \frac{ppm}{K})^2} = 0.026\% \text{ full scale value, typ.}$$

Remarks: ppm ≙ 10^{-6} % ≙ 10^{-2}

4.2.4 Single-ended/differential typification

For analog inputs Beckhoff makes a basic distinction between two types: *single-ended* (SE) and *differential (DIFF)*, referring to the difference in electrical connection with regard to the potential difference.

The diagram shows two-channel versions of an SE module and a DIFF module as examples for all multi-channel versions.

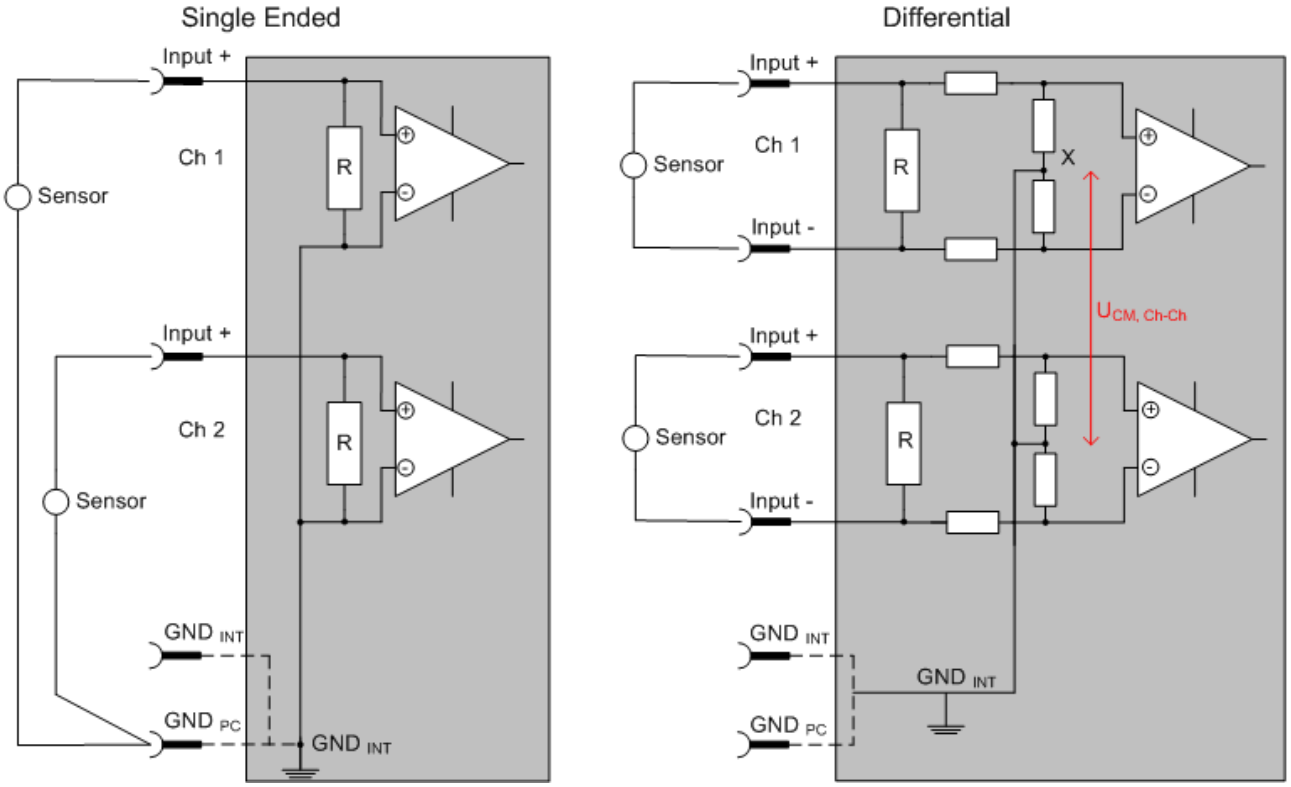


Fig. 24: SE and DIFF module as 2-channel version

Note: Dashed lines indicate that the respective connection may not necessarily be present in each SE or DIFF module. Electrical isolated channels are operating as differential type in general, hence there is no direct relation (voltaic) to ground within the module established at all. Indeed, specified information to recommended and maximum voltage levels have to be taken into account.

The basic rule

- Analog measurements always take the form of voltage measurements between two potential points. For voltage measurements a large R is used, in order to ensure a high impedance. For current measurements a small R is used as shunt. If the purpose is resistance measurement, corresponding considerations are applied.
  - Beckhoff generally refers to these two points as input+/signal potential and input-/reference potential.
  - For measurements between two potential points two potentials have to be supplied.
  - Regarding the terms “single-wire connection” or “three-wire connection”, please note the following for pure analog measurements: three- or four-wire connections can be used for sensor supply, but are not involved in the actual analog measurement, which always takes place between two potentials/wires.
    In particular this also applies to SE, even though the term suggests that only one wire is required.
- The term “electrical isolation” should be clarified in advance.
  Beckhoff I/O devices feature 1 to 8 or more analog channels; with regard to the channel connection a distinction is made in terms of:
  - how the channels WITHIN a module relate to each other, or
  - how the channels of SEVERAL modules relate to each other.

The property of electrical isolation indicates whether the channels are directly connected to each other.

- Beckhoff I/O devices always feature electrical isolation between the field/analog side and the bus/ EtherCAT side. In other words, if two analog I/O devices are not connected via the power contacts (cable), they are effectively electrically isolated.
- If channels within a device are electrically isolated, or if a single-channel device has no power contacts, the channels are effectively always differential. See also explanatory notes below. Differential channels are not necessarily electrically isolated.

- Analog measuring channels are subject to technical limits, both in terms of the recommended operating range (continuous operation) and the destruction limit. Please refer to the respective device documentation for further details.

Explanation

- **Differential (DIFF)**
  - Differential measurement is the most flexible concept. The user can freely choose both connection points, input+/signal potential and input-/reference potential, within the framework of the technical specification.
  - A differential channel can also be operated as SE, if the reference potential of several sensors is linked. This interconnection may take place via the system GND.
  - Since a differential channel is configured symmetrically internally (cf. Fig. SE and DIFF module as 2-channel variant), there will be a mid-potential (X) between the two supplied potentials that is the same as the internal ground/reference ground for this channel. If several DIFF channels are used in a module without electrical isolation, the technical property V_{CM} (common-mode voltage) indicates the degree to which the mean voltage of the channels may differ.
  - The internal reference ground may be accessible as connection point at the device, in order to stabilize a defined GND potential in the device. In this case it is particularly important to pay attention to the quality of this potential (noiselessness, voltage stability). At this GND point a wire may be connected to make sure that $V_{CM,max}$ is not exceeded in the differential sensor cable. If differential channels are not electrically isolated, usually only one $V_{CM,\,max}$ is permitted. If the channels are electrically isolated this limit should not apply, and the channels voltages may differ up to the specified separation limit.
  - Differential measurement in combination with correct sensor wiring has the special advantage that any interference affecting the sensor cable (ideally the feed and return line are arranged side by side, so that interference signals have the same effect on both wires) has very little effect on the measurement, since the potential of both lines varies jointly (hence the term common mode). In simple terms: Common-mode interference has the same effect on both wires in terms of amplitude and phasing.
  - Nevertheless, the suppression of common-mode interference within a channel or between channels is subject to technical limits, which are specified in the technical data.
  - Further helpfully information on this topic can be found on the documentation page *Configuration of 0/4..20 mA differential inputs* (see documentation for the EL30xx terminals, for example).
- **Single-ended (SE)**
  - If the analog circuit is designed as SE, the input/reference wire is internally fixed to a certain potential that cannot be changed. This potential must be accessible from outside of the device on at least one point for connecting the reference potential, e.g. via the power contacts (cable).
  - In other words, in situations with several channels SE offers users the option to avoid returning at least one of the two sensor cables to the device (in contrast to DIFF). Instead, the reference wire can be consolidated at the sensors, e.g. in the system GND.
  - A disadvantage of this approach is that the separate feed and return line can result in voltage/ current variations, which a SE channel may no longer be able to handle. See common-mode interference. A V_{CM} effect cannot occur, since the device channels are internally always 'hard-wired' through the input/reference potential.

Typification of the 2/3/4-wire connection of current sensors

Current transducers/sensors/field devices (referred to in the following simply as 'sensor') with the industrial 0/4-20 mA interface typically have internal transformation electronics for the physical measured variable (temperature, current, etc.) at the current control output. These internal electronics must be supplied with energy (voltage, current). The type of cable for this supply thus separates the sensors into *self-supplied* or *externally supplied* sensors:

Self-supplied sensors

- The sensor draws the energy for its own operation via the sensor/signal cable + and -.
  So that enough energy is always available for the sensor's own operation and open-circuit detection is possible, a lower limit of 4 mA has been specified for the 4-20 mA interface, i.e., the sensor allows a minimum current of 4 mA and a maximum current of 20 mA to pass.
- 2-wire connection see Fig. *2-wire connection*, cf. IEC60381-1
- Such current transducers generally represent a current sink and thus like to sit between + and – as a 'variable load'. See also the data of the sensor manufacturer.

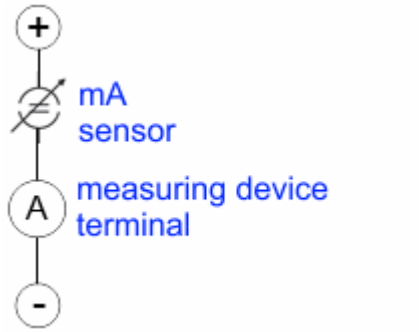


Fig. 25: 2-wire connection

Therefore, they are to be connected according to the Beckhoff terminology as follows:

preferably to **'single-ended' inputs** if the +Supply connections of the device are also to be used - connect to +Supply and Signal.

They can, however, also be connected to **'differential' inputs**, if the termination to GND is then manufactured on the application side – to be connected with the right polarity to +Signal and –Signal
It is important to refer to the information page *Configuration of 0/4…20 mA differential inputs* (see documentation for the EL30xx terminals, for example)!

Externally supplied sensors

No external supply for sensors / actuators

⚠ WARNING

An external supply of sensors / actuators connected to I/O devices of the ELX/EPX series is not permitted!

All I/O devices of the ELX/EPX series are energy-supplying, associated equipment in terms of intrinsic safety. Therefore, connected sensors or actuators are supplied exclusively via the respective channel of the I/O device and must not be externally supplied in any form (e.g. via an additional, external supply voltage).

This limitation is also independent of whether the additional, external supply is energy-limited in the sense of IEC 60079-11.

Connection of any externally powered, intrinsically safe circuits to an I/O device of the ELX/EPX series is contrary to the intended use and the specified technical data on explosion protection. The explosion protection provided by the specified ignition protection type is thus automatically extinguished.

4.2.5 Common-mode voltage and reference ground (based on differential inputs)

Common-mode voltage (V_{cm}) is defined as the average value of the voltages of the individual connections/ inputs and is measured/specified against reference ground.

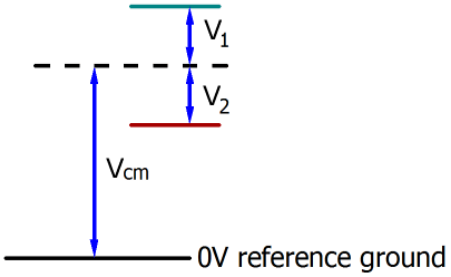


Fig. 26: Common-mode voltage (V_{cm})

The definition of the reference ground is important for the definition of the permitted common-mode voltage range and for measurement of the common-mode rejection ratio (CMRR) for differential inputs.

The reference ground is also the potential against which the input resistance and the input impedance for single-ended inputs or the common-mode resistance and the common-mode impedance for differential inputs is measured.

The reference ground is usually accessible at or near the I/O device. Locations for this can be terminal contacts, power contacts/power line or just a mounting rail.

Please refer to the documentation regarding positioning. The reference ground should be specified for the device under consideration.

For multi-channel I/O devices with resistive (= direct, ohmic, galvanic) or capacitive connection between the channels, the reference ground should preferably be the symmetry point of all channels, taking into account the connection resistances.

Reference ground samples for Beckhoff IO devices:

1. Internal AGND fed out: EL3102/EL3112, resistive connection between the channels
2. 0 V power contact: EL3104/EL3114, resistive connection between the channels and AGND; AGND connected to 0 V power contact with low-resistance
3. Earth or SGND (shield GND):
   - EL3174-0002: Channels have no resistive connection between each other, although they are capacitively coupled to SGND via leakage capacitors
   - EL3314: No internal ground fed out to the terminal points, although capacitive coupling to SGND

4.2.6 Dielectric strength

A distinction should be made between:

- Dielectric strength (destruction limit): Exceedance can result in irreversible changes to the electronics
  - Against a specified reference ground
  - Differential
- Recommended operating voltage range: If the range is exceeded, it can no longer be assumed that the system operates as specified
  - Against a specified reference ground
  - Differential

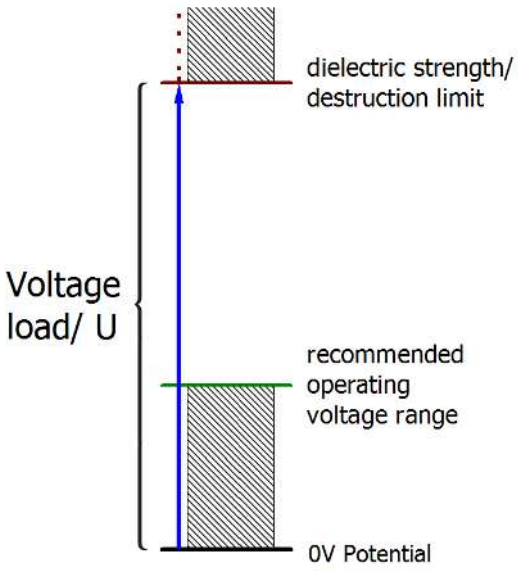


Fig. 27: Recommended operating voltage range

The device documentation may contain particular specifications and timings, taking into account:

- Self-heating
- Rated voltage
- Insulating strength
- Edge steepness of the applied voltage or holding periods
- Normative environment (e.g. PELV)

4.2.7 Temporal aspects of analog/digital conversion

Analog output

The following information applies analogously to analog signal output via DAC (digital-to-analog converter).

The conversion of the constant electrical input signal to a value-discrete digital and machine-readable form takes place in the analog Beckhoff EL/KL/EP input modules with ADC (analog digital converter). Although different ADC technologies are common, from the user's point of view they all have one common feature: after the end of the conversion, a certain digital value is available for further processing in the controller. This digital value, the so-called analog process data, has a fixed temporal relationship with the “original parameter”, i.e. the electrical input value. Therefore, corresponding temporal characteristic data can be determined and specified for Beckhoff analogue input devices.

This process involves several functional components, which act more or less strongly in every AI (analog input) module:

- the electrical input circuit
- the analog/digital conversion
- the digital further processing
- the final provision of the process and diagnostic data for collection at the fieldbus (EtherCAT, K-bus, etc.)

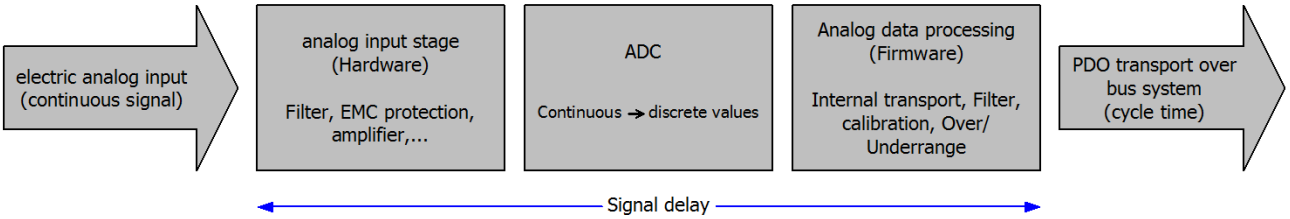


Fig. 28: Signal processing analog input

Two aspects are crucial from a user perspective:

- “How often do I receive new values?”, i.e. a sampling rate in terms of speed with regard to the device/ channel
- What delay does the (whole) AD conversion of the device/channel cause?
  So hardware and firmware parts in toto. For technological reasons, the signal characteristics must be considered to determine this specification: depending on the signal frequency, there may be different propagation times through the system.

This is the “external” view of the “Beckhoff AI channel” system – internally the signal delay in particular is composed of different components: hardware, amplifier, conversion itself, data transport and processing. Internally a higher sampling rate may be used (e.g. in the deltaSigma converters) than is offered “externally” from the user perspective. From a user perspective of the “Beckhoff AI channel” component this is usually irrelevant or is specified accordingly, if it is relevant for the function.

For Beckhoff AI devices the following specification parameters for the AI channel are available for the user from a temporal perspective:

1. Minimum conversion time [ms, µs]

This is the reciprocal value of the maximum **sampling rate** [Sps, samples per second]:
Specifies how often the analog channel provides a newly detected process data value for collection by the fieldbus. Whether the fieldbus (EtherCAT, K-bus) fetches the value with the same speed (i.e. synchronous), or more quickly (if the AI channel operates in slow FreeRun mode) or more slowly (e.g. with oversampling), is then a question of the fieldbus setting and which modes the AI device supports.
For EtherCAT devices the so-called toggle bit indicates (by toggling) for the diagnostic PDOs when a newly determined analog value is available.
Accordingly, a maximum conversion time, i.e. a smallest sampling rate supported by the AI device, can be specified.
Corresponds to IEC 61131-2 Chap. 7.10.2 2, "Sampling repeat time"

2. Typical signal delay

Corresponds to IEC 61131-2, Chapter 7.10.2 1, "Sampling duration". From this perspective it includes all internal hardware and firmware components, but not "external" delay components from the fieldbus or the controller (TwinCAT).
This delay is particularly relevant for absolute time considerations, if AI channels also provide a timestamp that corresponds to the amplitude value – which can be assumed to match the physically prevailing amplitude value at the time.
Due to the frequency-dependent runtime of a signal, a dedicated value can only be specified for a given signal. The value also depends on possibly changing filter settings of the channel.
A typical characterization in the device documentation can be:

2.1 Signal delay (step response)

Keyword settling time:
The square wave signal can be generated externally with a frequency generator (note impedance!).
The 90% limit is used as detection threshold.
The signal delay [ms, µs] is then the time interval between the (ideal) electrical square wave signal and the time when the analog process value has reached the 90% amplitude.

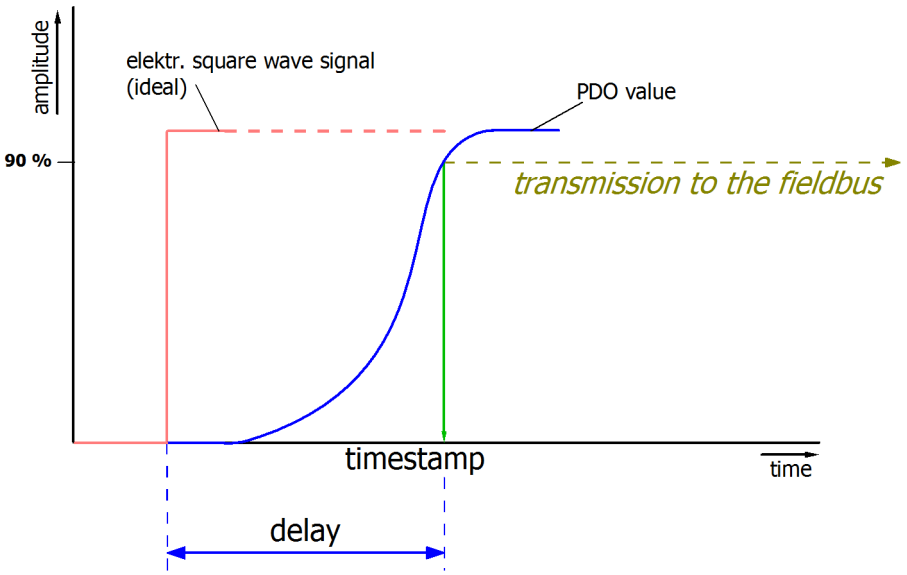


Fig. 29: Diagram Signal delay (step response)

2.2 Signal delay (linear)

Keyword group delay:
Describes the delay of a frequency-constant signal
Test signal can be generated externally with a frequency generator, e.g. as sawtooth or sine. Reference would then be a simultaneous square wave signal.
The signal delay [ms, µs] is then the time interval between the applied electrical signal of a certain amplitude and the moment when the analog process value reaches the same value.
For this purpose, the test frequency must be selected in a reasonable range; this can be, for example, 1/20 of the maximum sampling rate.

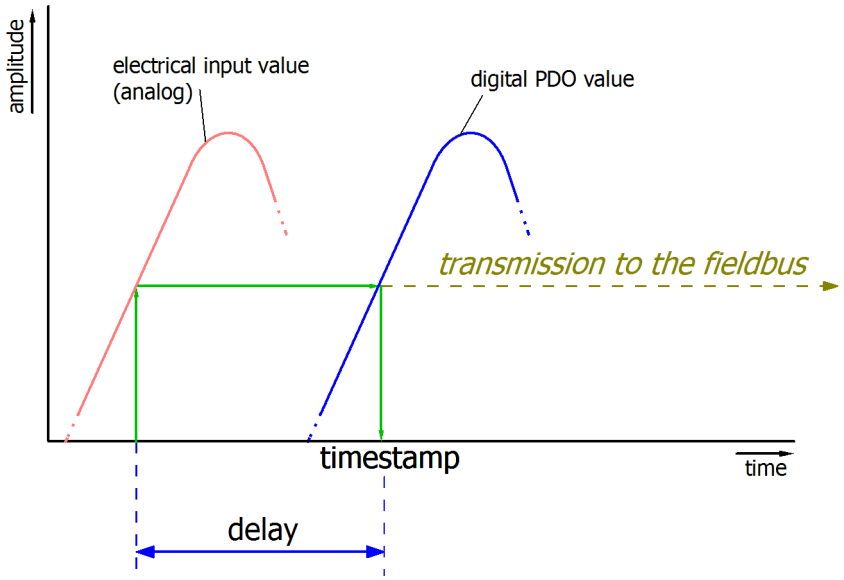


Fig. 30: Diagram Signal delay (linear)

3. Additional information

Additional information may be provided in the specification, e.g.

- actual sampling rate of the ADC (if different from the channel sampling rate)
- time correction values for runtimes with different filter settings
- etc.

4.3 NAMUR basic information

The abbreviation of NAMUR, “User Association of Automation Technology in Process Industries” identifies an international association for users of automation technology that considers the interests related to standardization, devices and measurement control (or similar) of the Process Industries as its major task. In this role, the NAMUR releases the so called NE (proposed standards), each numbered continuously.

Information with regard to the implementation of this recommendation in Beckhoff products are specified in sections *Technical data* and *Process data* of this documentation.

Analog measured values

The analog output value of a sensor that can be measured among other things as a certain current value represents the measurement information (M).

By means of NAMUR NE43 (“Standardization of the signal level for the failure information of digital transmitters with analog output signal“) a recommendation – irrespective of the sensor manufacturer – for standardized failure information (A) is defined in addition to the measurement information (e.g. malfunction of a measurement converter, error in connective wires, failure of an auxiliary energy etc.). The failure information states that there is an error in the measuring system. This concerns the analog output signal of sensors in a current loop and therefore in the form of a current value. A current value lying outside of the limits defined by NAMUR is defined as invalid and is thus interpreted as failure information. The following diagram illustrates this:

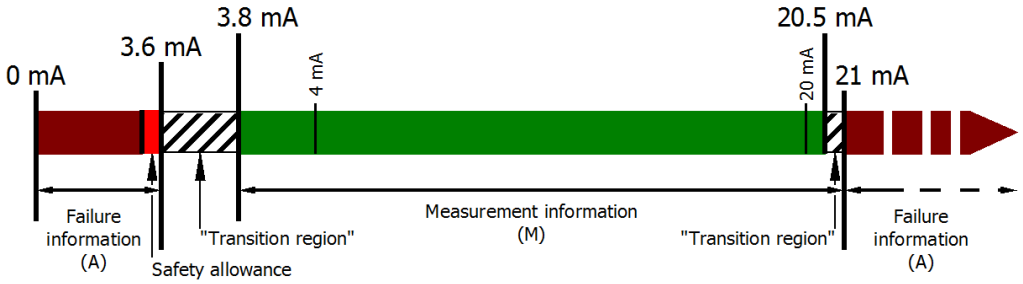


Fig. 31: Representation of the definitions from NAMUR recommendation NE43, version 03/02/2003

5 Parameterization and programming

5.1 TwinCAT Quick Start

TwinCAT is a development environment for real-time control including multi-PLC system, NC axis control, programming and operation. The whole system is mapped through this environment and enables access to a programming environment (including compilation) for the controller. Individual digital or analog inputs or outputs can also be read or written directly, in order to verify their functionality, for example.

For further information please refer to http://infosys.beckhoff.com:

- **EtherCAT Systemmanual:**
  Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → I/O - Configuration
- In particular, TwinCAT driver installation:
  Fieldbus components → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation

Devices contain the terminals for the actual configuration. All configuration data can be entered directly via editor functions (offline) or via the “Scan” function (online):

- **“offline”**: The configuration can be customized by adding and positioning individual components. These can be selected from a directory and configured.
  - The procedure for offline mode can be found under http://infosys.beckhoff.com:
    TwinCAT 2 → TwinCAT System Manager → IO - Configuration → Adding an I/O Device
- **“online”**: The existing hardware configuration is read
  - See also http://infosys.beckhoff.com:
    Fieldbus components → Fieldbus cards and switches → FC900x – PCI Cards for Ethernet → Installation → Searching for devices

The following relationship is envisaged from user PC to the individual control elements:

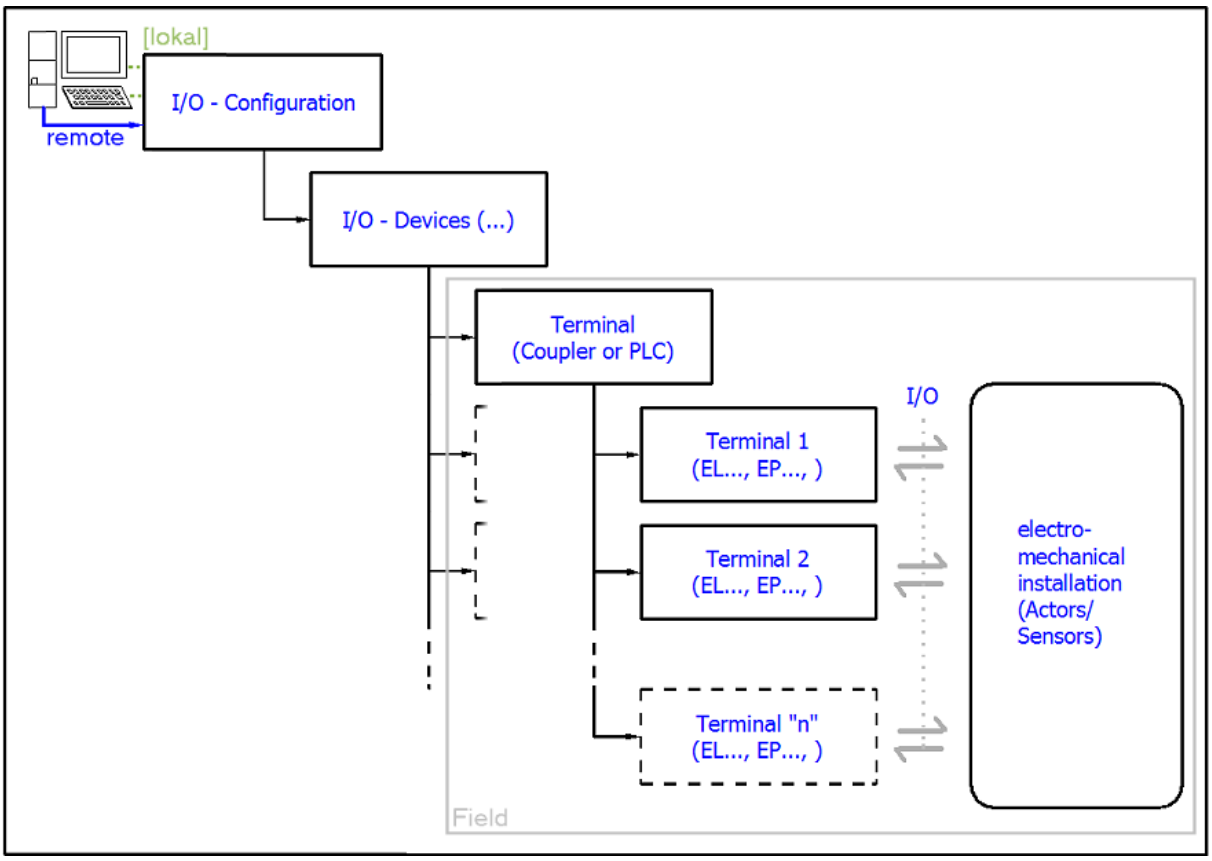


Fig. 32: Relationship between user side (commissioning) and installation

The user inserting of certain components (I/O device, terminal, box...) is the same in TwinCAT 2 and TwinCAT 3. The descriptions below relate to the online procedure.

Sample configuration (actual configuration)

Based on the following sample configuration, the subsequent subsections describe the procedure for TwinCAT 2 and TwinCAT 3:

- Control system (PLC) **CX2040** including **CX2100-0004** power supply unit
- Connected to the CX2040 on the right (E-bus):
  EL1004 (4-channel digital input terminal 24 V_{DC})
- Linked via the X001 port (RJ-45): **EK1100** EtherCAT Coupler
- Connected to the EK1100 EtherCAT coupler on the right (E-bus):
  EL2008 (8-channel digital output terminal 24 V_{DC}; 0.5 A)
- (Optional via X000: a link to an external PC for the user interface)

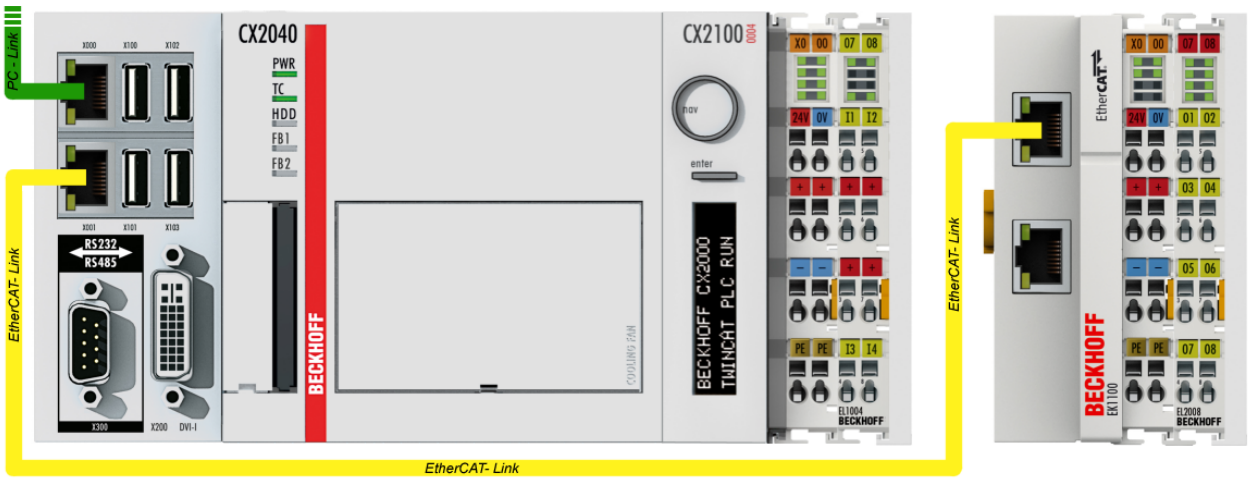


Fig. 33: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

Note that all combinations of a configuration are possible; for example, the EL1004 terminal could also be connected after the coupler, or the EL2008 terminal could additionally be connected to the CX2040 on the right, in which case the EK1100 coupler wouldn't be necessary.

5.1.1 TwinCAT 2

Startup

TwinCAT basically uses two user interfaces: the TwinCAT System Manager for communication with the electromechanical components and TwinCAT PLC Control for the development and compilation of a controller. The starting point is the TwinCAT System Manager.

After successful installation of the TwinCAT system on the PC to be used for development, the TwinCAT 2 System Manager displays the following user interface after startup:

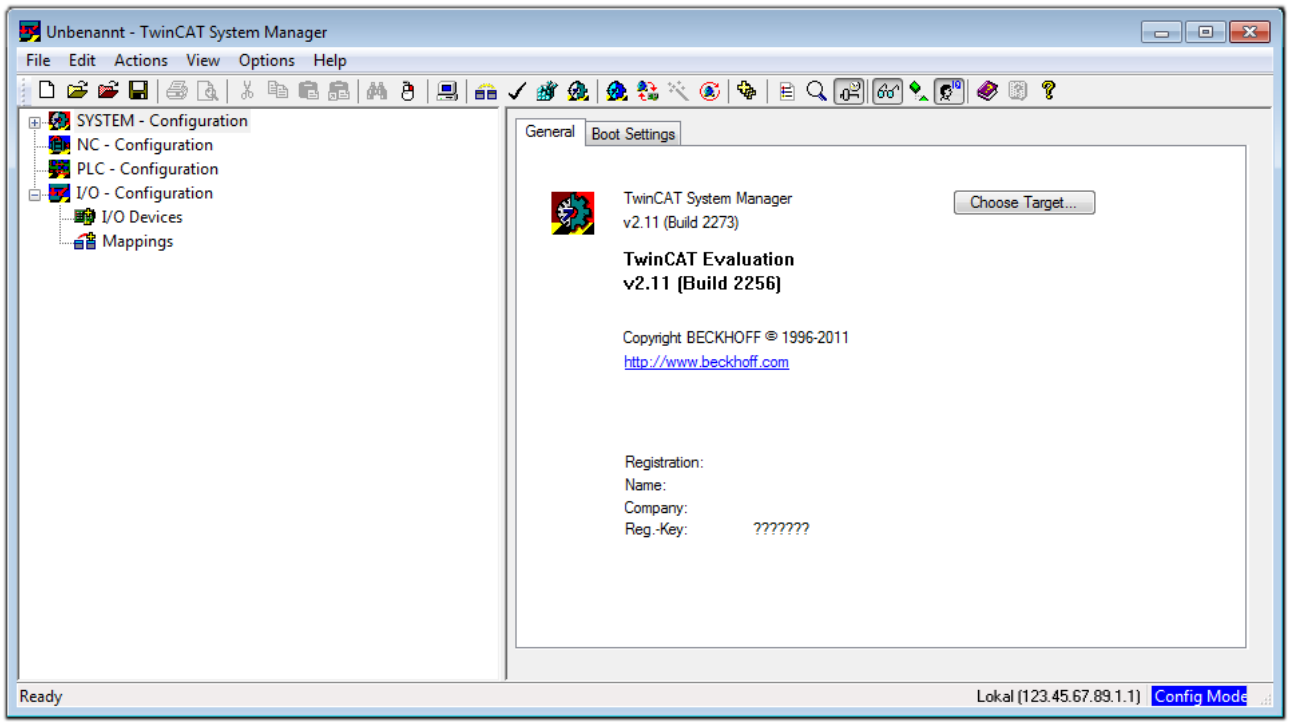


Fig. 34: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is “Insert Device [▶ 50]”.

If the intention is to address the TwinCAT runtime environment installed on a PLC as development environment remotely from another system, the target system must be made known first. In the menu under “Actions” → “Choose Target System...”, via the symbol “  ” or the “F8” key, open the following window:

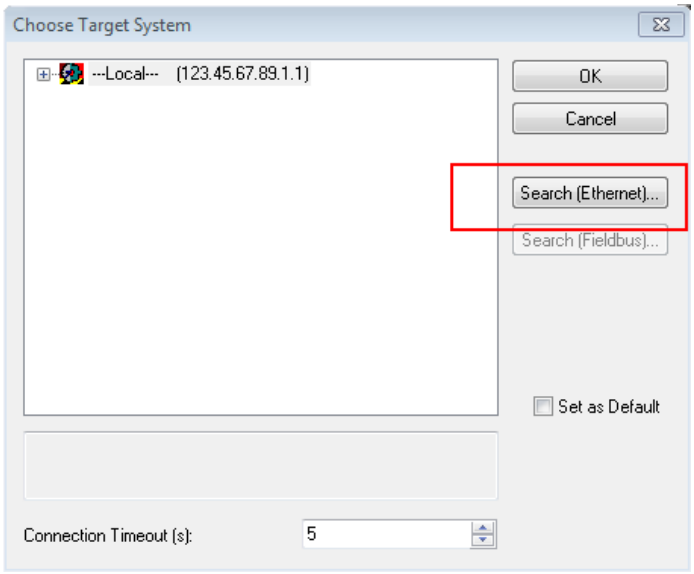


Fig. 35: Selection of the target system

Use “Search (Ethernet)...” to enter the target system. Thus a next dialog opens to either:

- enter the known computer name after “Enter Host Name / IP:” (as shown in red)
- perform a “Broadcast Search” (if the exact computer name is not known)
- enter the known computer IP or AmsNetID.

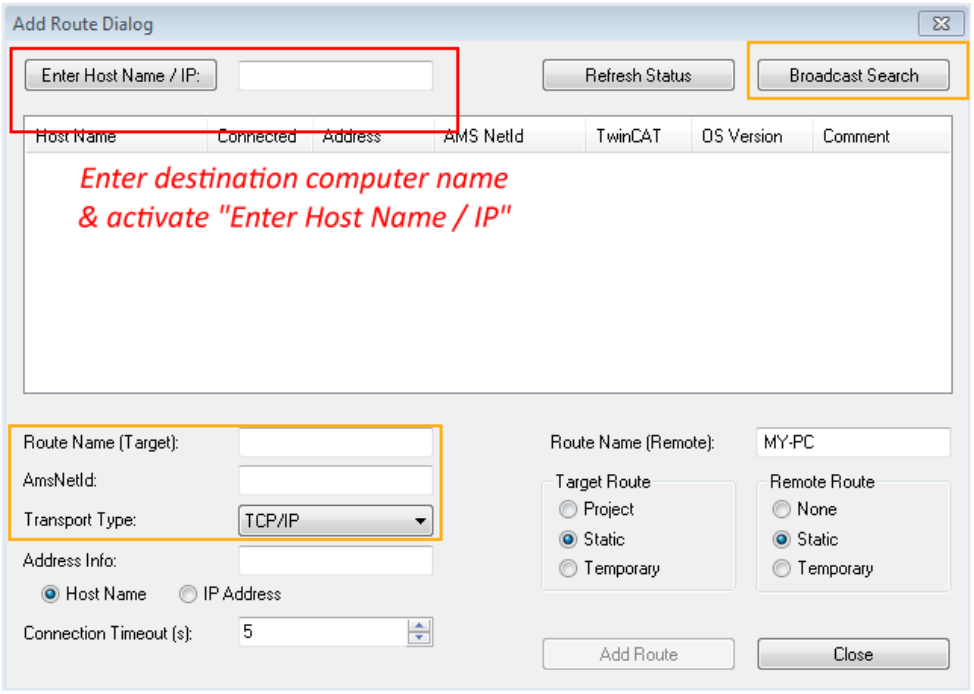
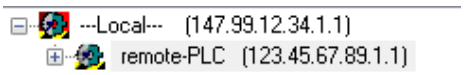


Fig. 36: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (a password may have to be entered):



After confirmation with “OK” the target system can be accessed via the System Manager.

Adding devices

In the configuration tree of the TwinCAT 2 System Manager user interface on the left, select “I/O Devices” and then right-click to open a context menu and select “Scan Devices…”, or start the action in the menu bar via  . The TwinCAT System Manager may first have to be set to “Config mode” via  or via menu “Actions” → “Set/Reset TwinCAT to Config Mode…” (Shift + F4).

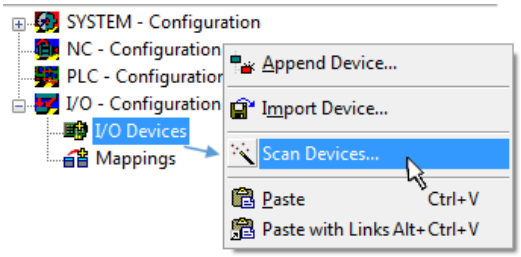


Fig. 37: Select “Scan Devices...”

Confirm the warning message, which follows, and select “EtherCAT” in the dialog:

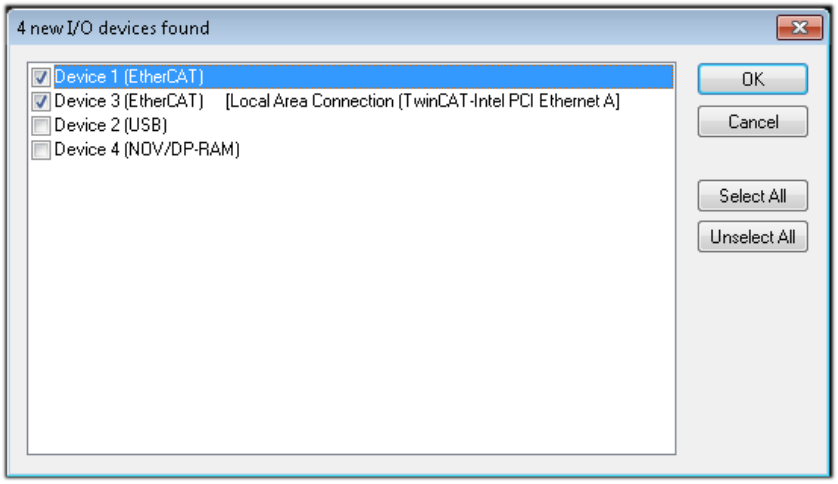


Fig. 38: Automatic detection of I/O devices: selection the devices to be integrated

Confirm the message “Find new boxes”, in order to determine the terminals connected to the devices. “Free Run” enables manipulation of input and output values in “Config mode” and should also be acknowledged.

Based on the sample configuration described at the beginning of this section, the result is as follows:

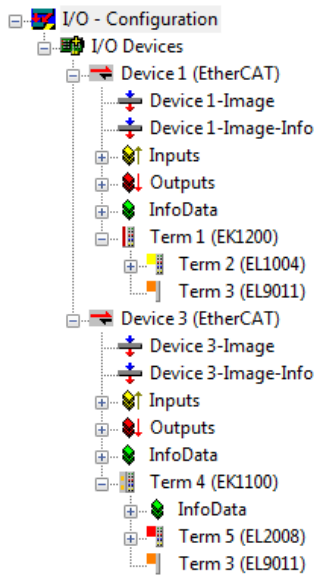


Fig. 39: Mapping of the configuration in the TwinCAT 2 System Manager

The whole process consists of two stages, which may be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan can also be initiated by selecting “Device ...” from the context menu, which then reads the elements present in the configuration below:

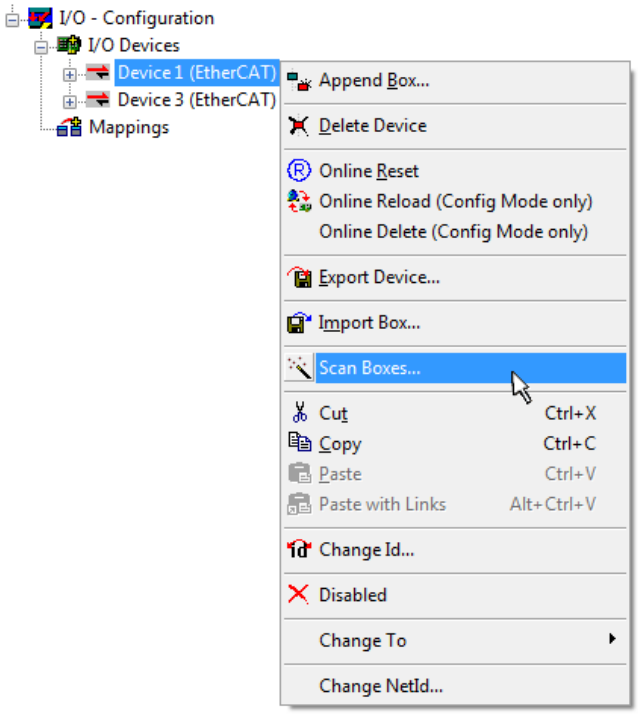


Fig. 40: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

Programming and integrating the PLC

TwinCAT PLC Control is the development environment for the creation of the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)

- **Graphical languages**
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - The Continuous Function Chart Editor (CFC)
  - Sequential Function Chart (SFC)

The following section refers to Structured Text (ST).

After starting TwinCAT PLC Control, the following user interface is shown for an initial project:

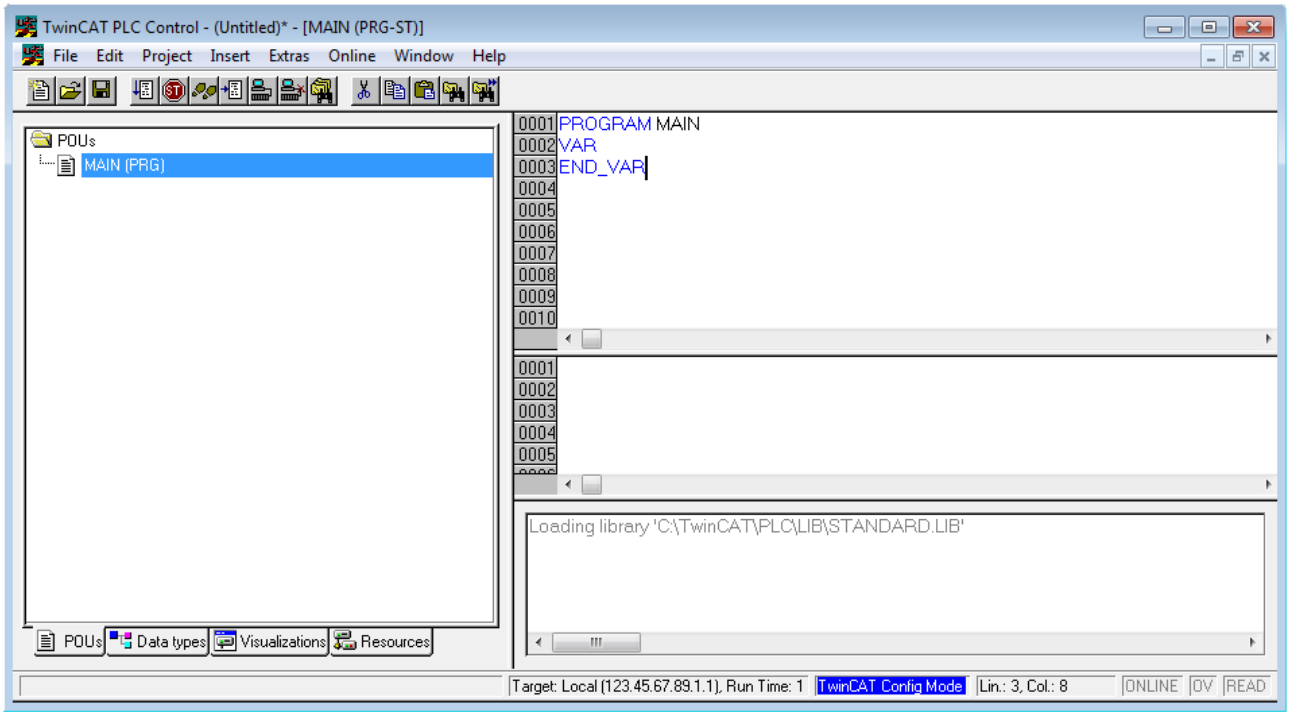


Fig. 41: TwinCAT PLC Control after startup

Sample variables and a sample program have been created and stored under the name “PLC_example.pro”:

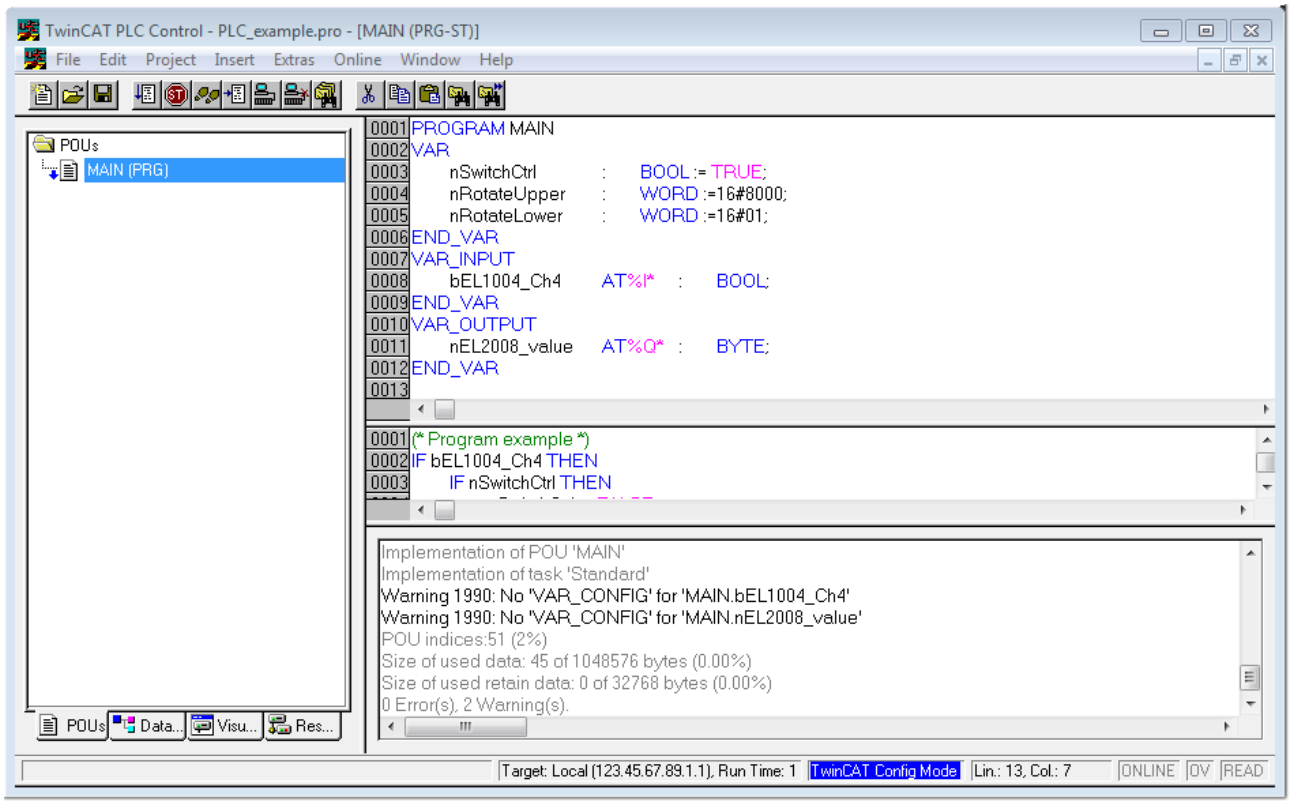


Fig. 42: Sample program with variables after a compile process (without variable integration)

Warning 1990 (missing “VAR_CONFIG”) after a compile process indicates that the variables defined as external (with the ID “AT%I*” or “AT%Q*”) have not been assigned. After successful compilation, TwinCAT PLC Control creates a “*.tpy” file in the directory in which the project was stored. This file (“*.tpy”) contains variable assignments and is not known to the System Manager, hence the warning. Once the System Manager has been notified, the warning no longer appears.

First, integrate the TwinCAT PLC Control project in the **System Manager** via the context menu of the PLC configuration; right-click and select “Append PLC Project…”:

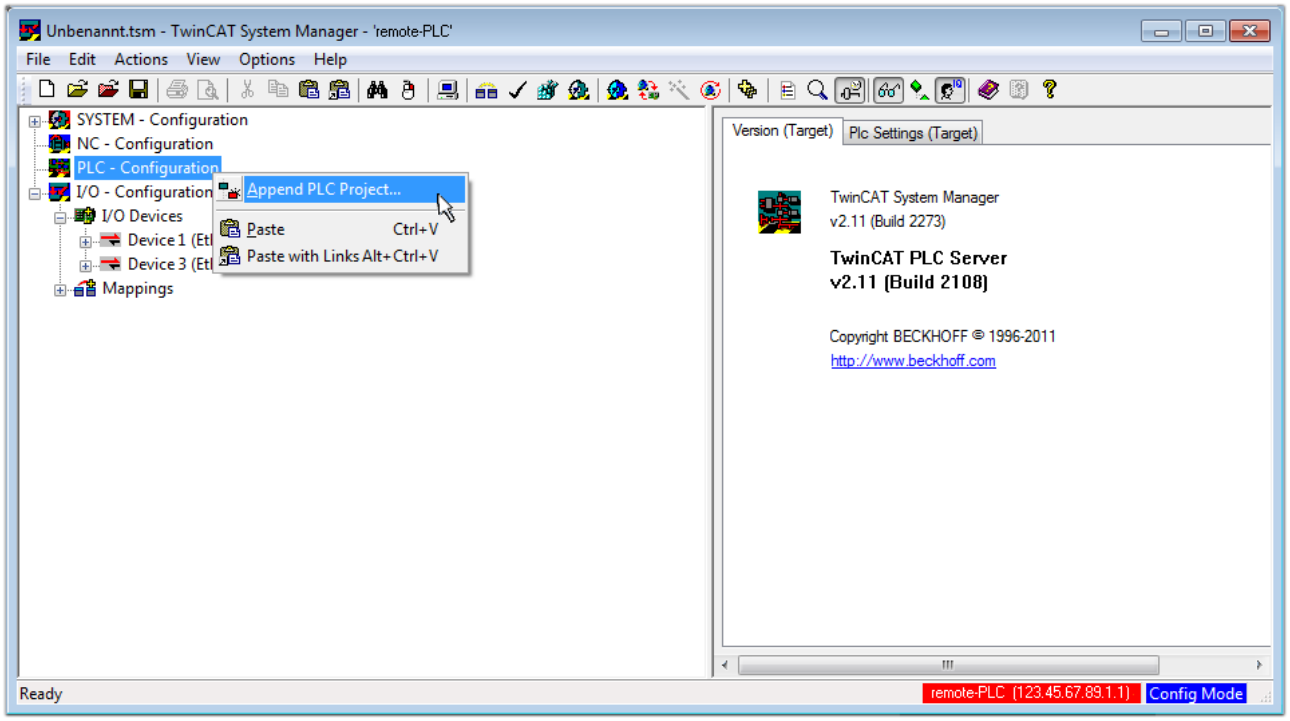


Fig. 43: Appending the TwinCAT PLC Control project

Select the PLC configuration “PLC_example.tpy” in the browser window that opens. The project including the two variables identified with “AT” are then integrated in the configuration tree of the System Manager:

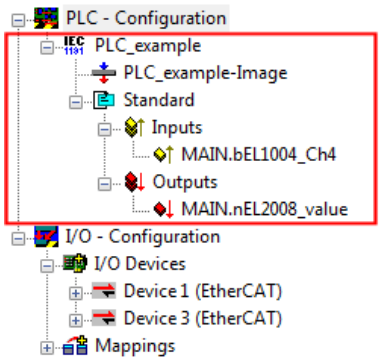


Fig. 44: PLC project integrated in the PLC configuration of the System Manager

The two variables “bEL1004_Ch4” and “nEL2008_value” can now be assigned to certain process objects of the I/O configuration.

Assigning variables

Open a window for selecting a suitable process object (PDO) via the context menu of a variable of the integrated project “PLC_example” and via “Modify Link...” “Standard”:

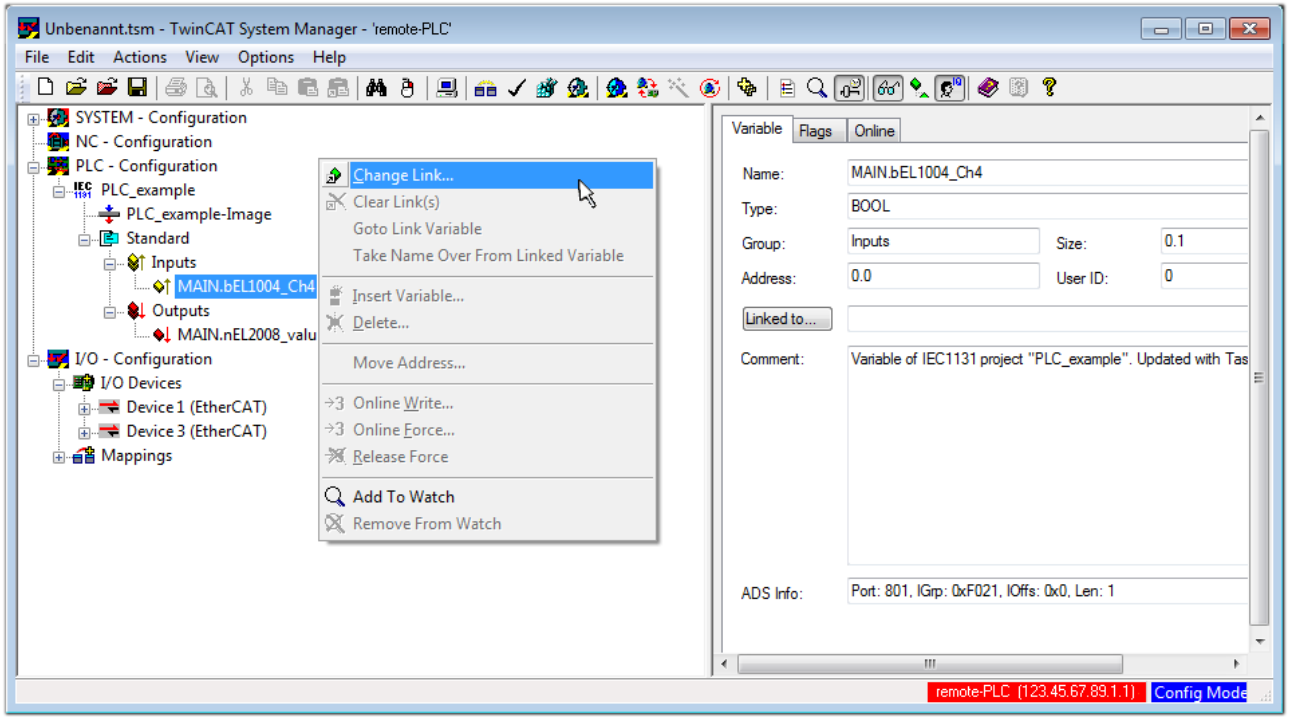


Fig. 45: Creating the links between PLC variables and process objects

In the window that opens, the process object for the variable “bEL1004_Ch4” of type BOOL can be selected from the PLC configuration tree:

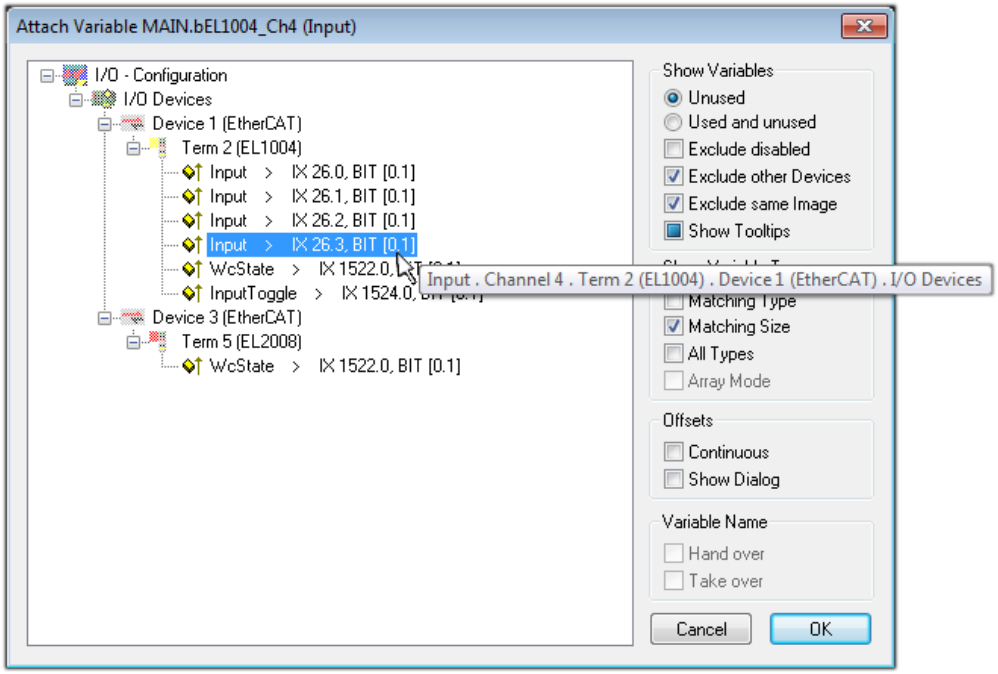


Fig. 46: Selecting PDO of type BOOL

According to the default setting, certain PDO objects are now available for selection. In this sample the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox “All types” must be ticked for creating the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable. The following diagram shows the whole process:

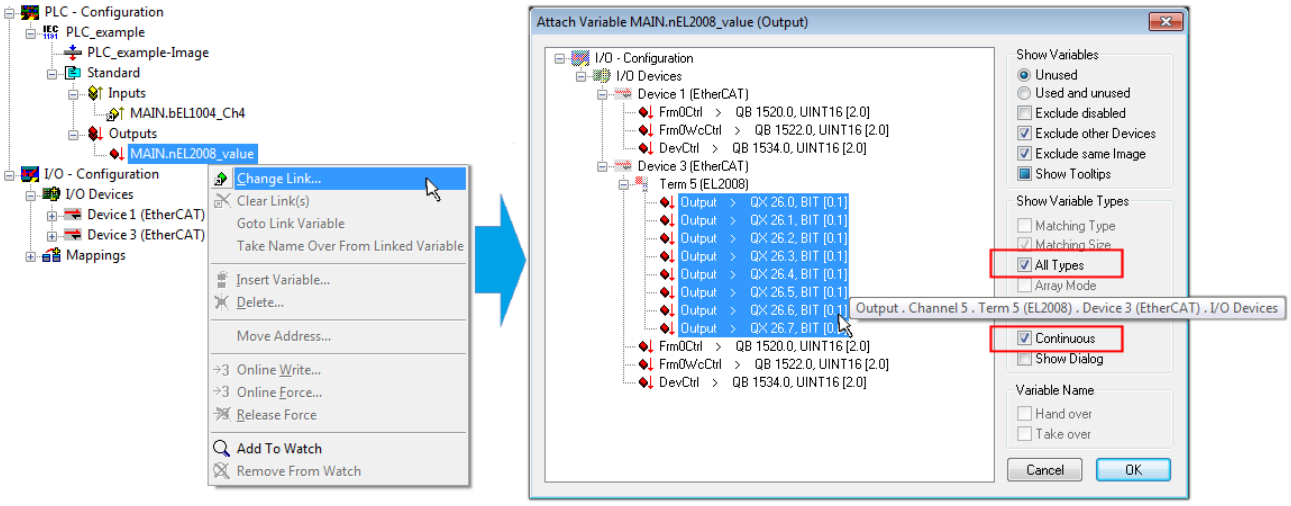


Fig. 47: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the variable “nEL2008_value” sequentially to all eight selected output bits of the EL2008 terminal. In this way it is possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) at the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting a “Goto Link Variable” from the context menu of a variable. The object opposite, in this case the PDO, is automatically selected:

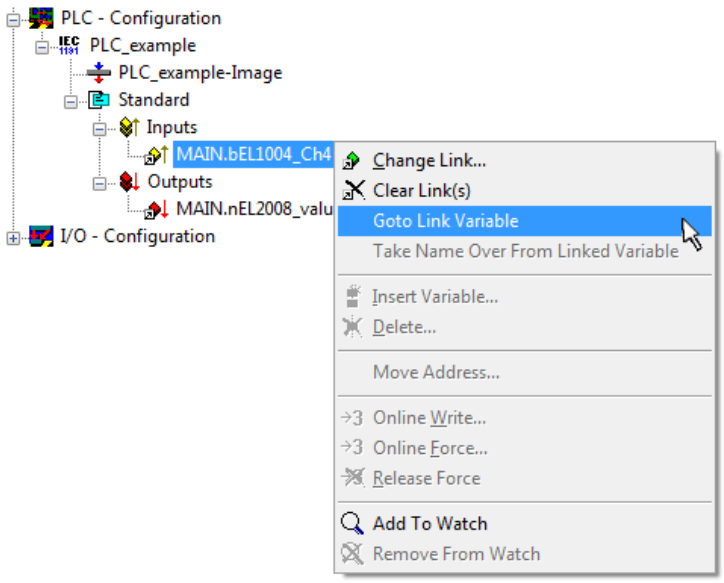
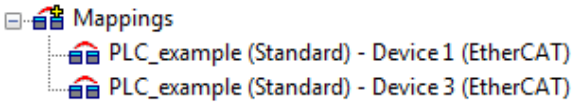


Fig. 48: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

The process of assigning variables to the PDO is completed via the menu selection “Actions” → “Generate Mappings”, key Ctrl+M or by clicking on the symbol  in the menu.

This can be visualized in the configuration:



The process of creating links can also take place in the opposite direction, i.e. starting with individual PDOs to variable. However, in this example it would then not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word, integer or similar PDO, it is possible to allocate this a set of bit-standardized variables (type “BOOL”). Here, too, a “Goto Link Variable” from the context menu of a PDO can be executed in the other direction, so that the respective PLC instance can then be selected.

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated. First, the configuration can be verified via  (or via “Actions” → “Check Configuration”). If no error is present, the configuration can be activated via  (or via “Actions” → “Activate Configuration…”) to transfer the System Manager settings to the runtime system. Confirm the messages “Old configurations are overwritten!” and “Restart TwinCAT system in Run mode” with “OK”.

A few seconds later the real-time status 

is displayed at the bottom right in the System Manager. The PLC system can then be started as described below.

Starting the controller

Starting from a remote system, the PLC control has to be linked with the Embedded PC over Ethernet via “Online” → “Choose Run-Time System…”:

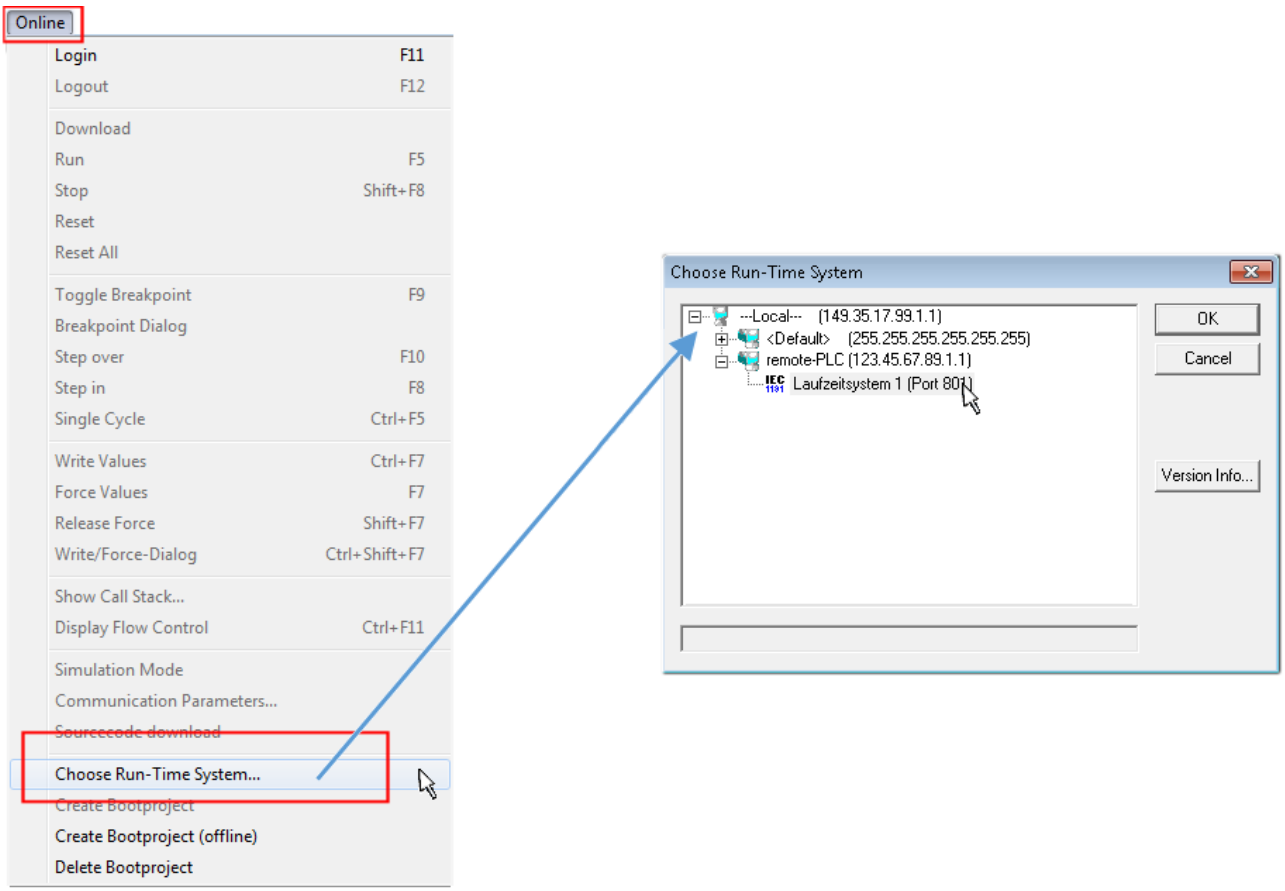


Fig. 49: Choose target system (remote)

In this sample “Runtime system 1 (port 801)” is selected and confirmed. Link the PLC with the real-time system via menu option “Online” → “Login”, the F11 key or by clicking on the symbol  . The control program can then be loaded for execution. This results in the message “No program on the controller! Should the new program be loaded?”, which should be acknowledged with “Yes”. The runtime environment is ready for the program start:

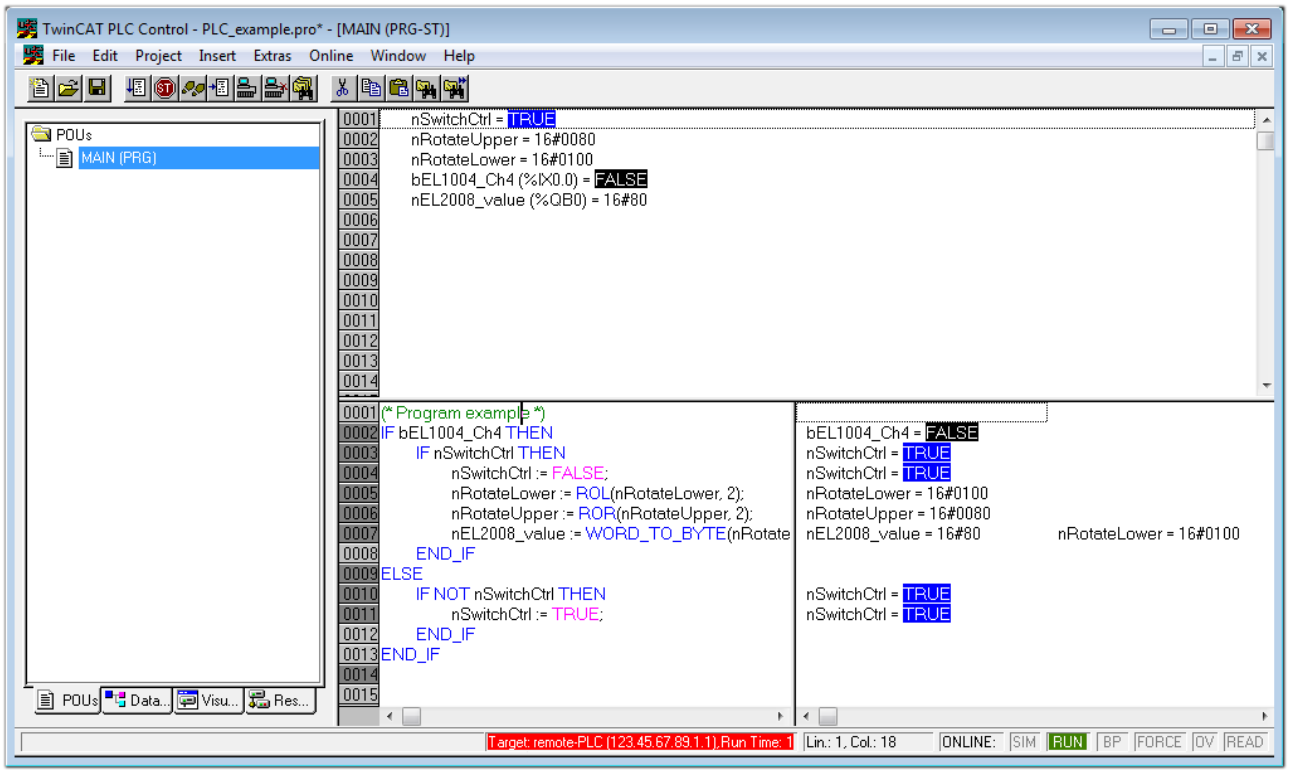


Fig. 50: PLC Control logged in, ready for program startup

The PLC can now be started via “Online” → “Run”, F5 key or  .

5.1.2 TwinCAT 3

Startup

TwinCAT makes the development environment areas available together with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (cf. “TwinCAT System Manager” of TwinCAT 2) for communication with the electromechanical components.

After successful installation of the TwinCAT system on the PC to be used for development, TwinCAT 3 (shell) displays the following user interface after startup:

Fig. 51: Initial TwinCAT 3 user interface

First create a new project via  New TwinCAT Project... (or under “File”→“New”→ “Project…”). In the following dialog make the corresponding entries as required (as shown in the diagram):

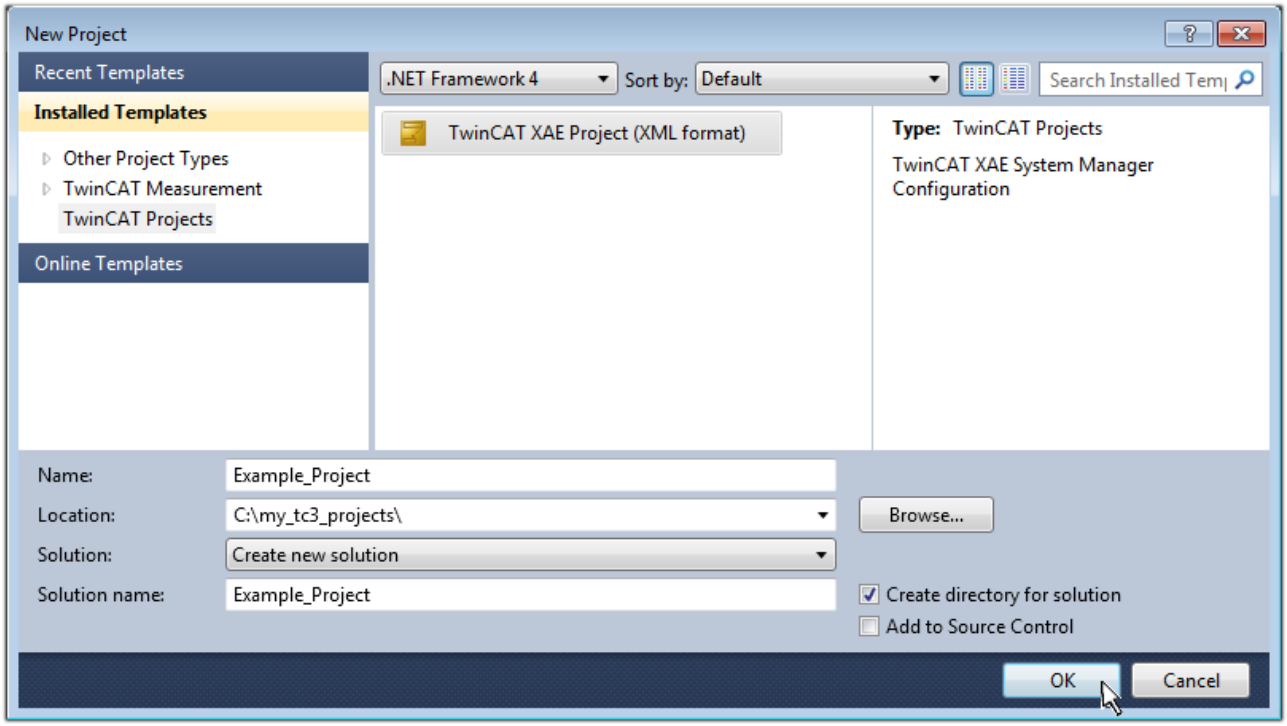


Fig. 52: Create new TwinCAT project

The new project is then available in the project folder explorer:

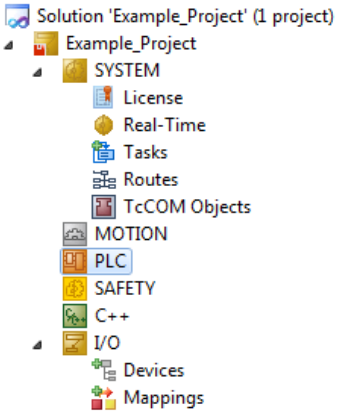
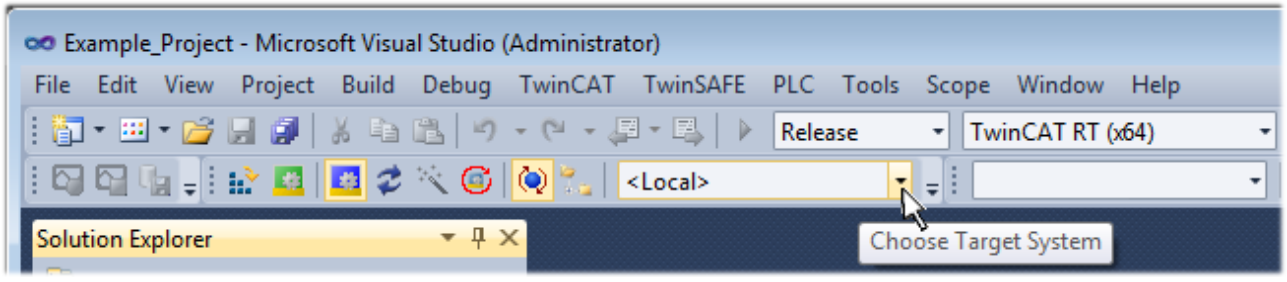


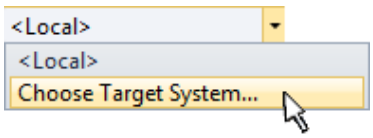
Fig. 53: New TwinCAT3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is “Insert Device [▶ 62]”.

If the intention is to address the TwinCAT runtime environment installed on a PLC as development environment remotely from another system, the target system must be made known first. Via the symbol in the menu bar:



expand the pull-down menu:



and open the following window:

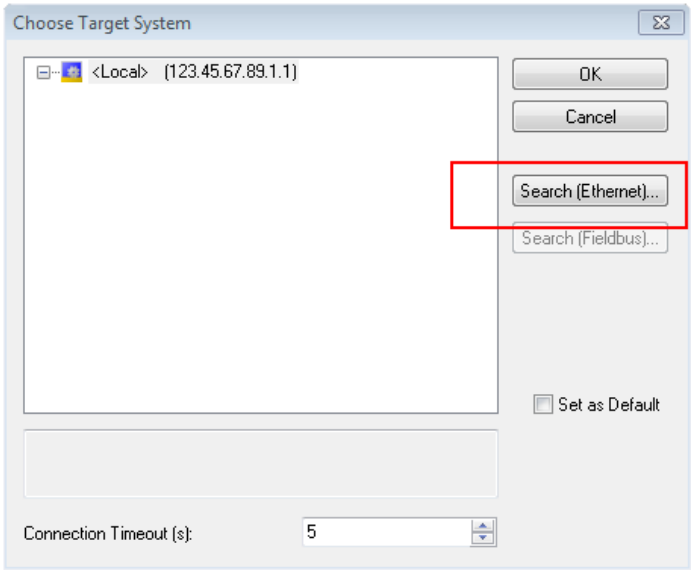


Fig. 54: Selection dialog: Choose the target system

Use “Search (Ethernet)...” to enter the target system. Thus a next dialog opens to either:

- enter the known computer name after “Enter Host Name / IP:” (as shown in red)
- perform a “Broadcast Search” (if the exact computer name is not known)
- enter the known computer IP or AmsNetID.

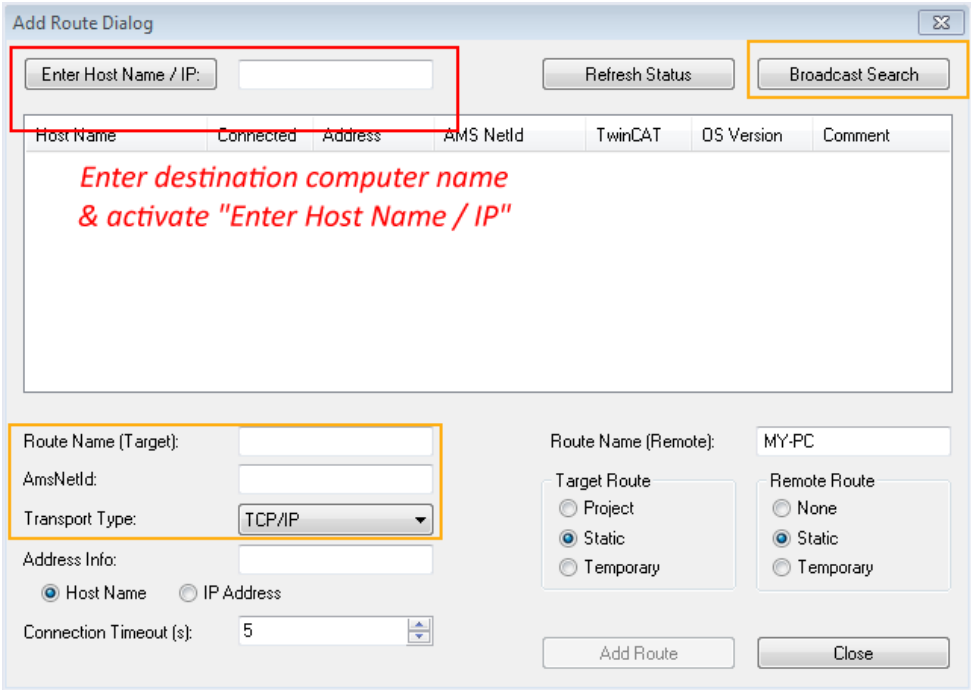
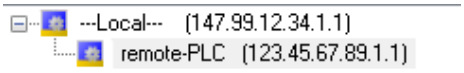


Fig. 55: Specify the PLC for access by the TwinCAT System Manager: selection of the target system

Once the target system has been entered, it is available for selection as follows (a password may have to be entered):



After confirmation with “OK” the target system can be accessed via the Visual Studio shell.

Adding devices

In the project folder explorer of the Visual Studio shell user interface on the left, select “Devices” within element “I/O”, then right-click to open a context menu and select “Scan” or start the action via  in the menu bar. The TwinCAT System Manager may first have to be set to “Config mode” via  or via the menu “TwinCAT” → “Restart TwinCAT (Config mode)”.

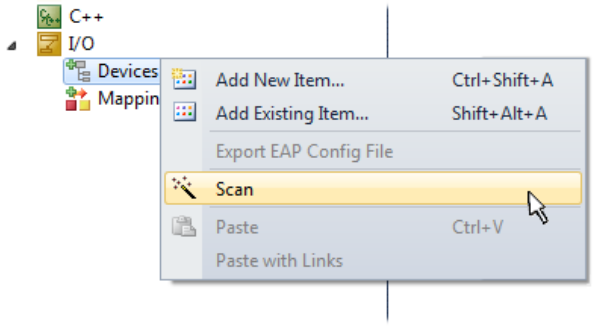


Fig. 56: Select “Scan”

Confirm the warning message, which follows, and select “EtherCAT” in the dialog:

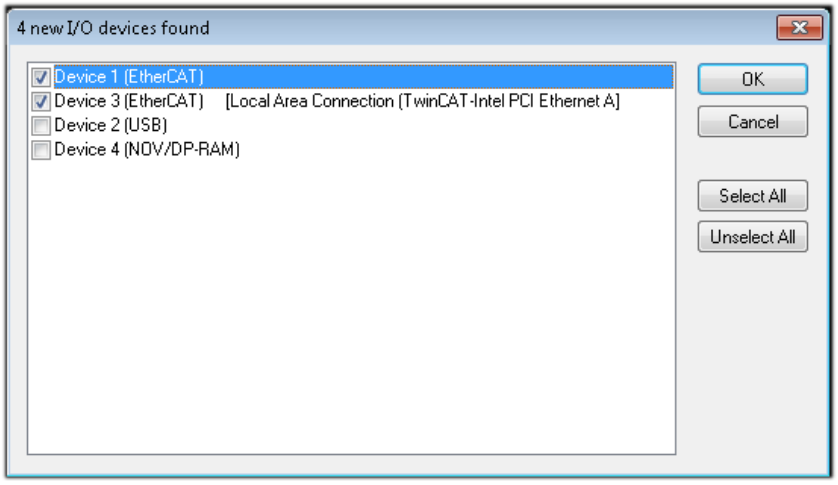


Fig. 57: Automatic detection of I/O devices: selection the devices to be integrated

Confirm the message “Find new boxes”, in order to determine the terminals connected to the devices. “Free Run” enables manipulation of input and output values in “Config mode” and should also be acknowledged.

Based on the sample configuration described at the beginning of this section, the result is as follows:

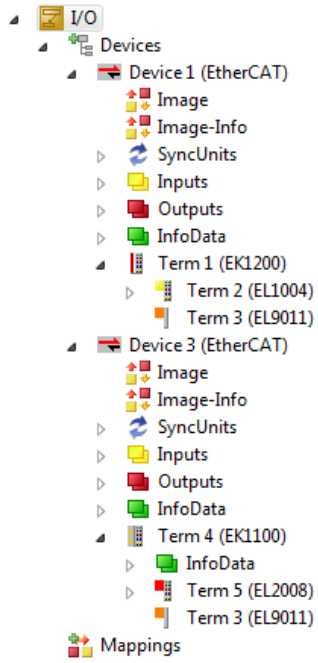


Fig. 58: Mapping of the configuration in VS shell of the TwinCAT3 environment

The whole process consists of two stages, which may be performed separately (first determine the devices, then determine the connected elements such as boxes, terminals, etc.). A scan can also be initiated by selecting “Device ...” from the context menu, which then reads the elements present in the configuration below:

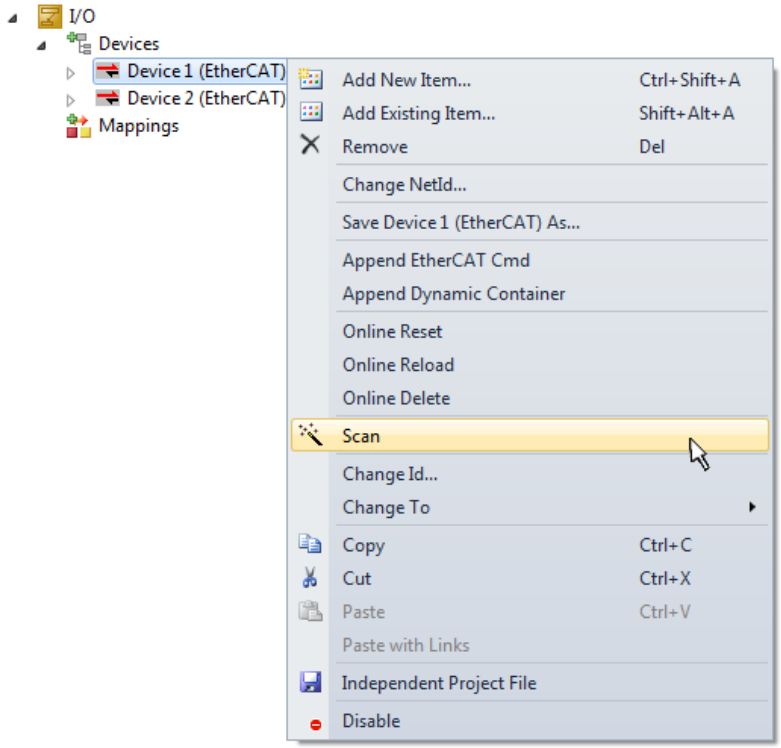


Fig. 59: Reading of individual terminals connected to a device

This functionality is useful if the actual configuration is modified at short notice.

Programming the PLC

TwinCAT PLC Control is the development environment for the creation of the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

- **Text-based languages**
  - Instruction List (IL)
  - Structured Text (ST)
- **Graphical languages**
  - Function Block Diagram (FBD)
  - Ladder Diagram (LD)
  - The Continuous Function Chart Editor (CFC)
  - Sequential Function Chart (SFC)

The following section refers to Structured Text (ST).

In order to create a programming environment, a PLC subproject is added to the project sample via the context menu of “PLC” in the project folder explorer by selecting “Add New Item….”:

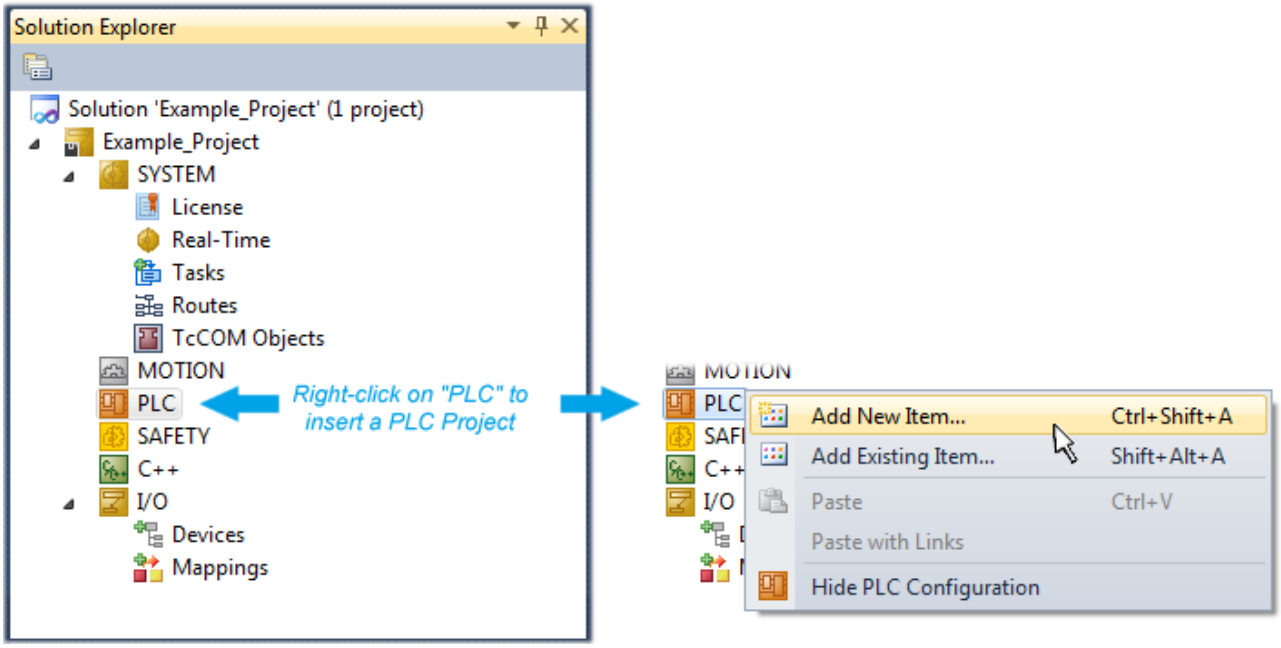


Fig. 60: Adding the programming environment in “PLC”

In the dialog that opens select “Standard PLC project” and enter “PLC_example” as project name, for example, and select a corresponding directory:

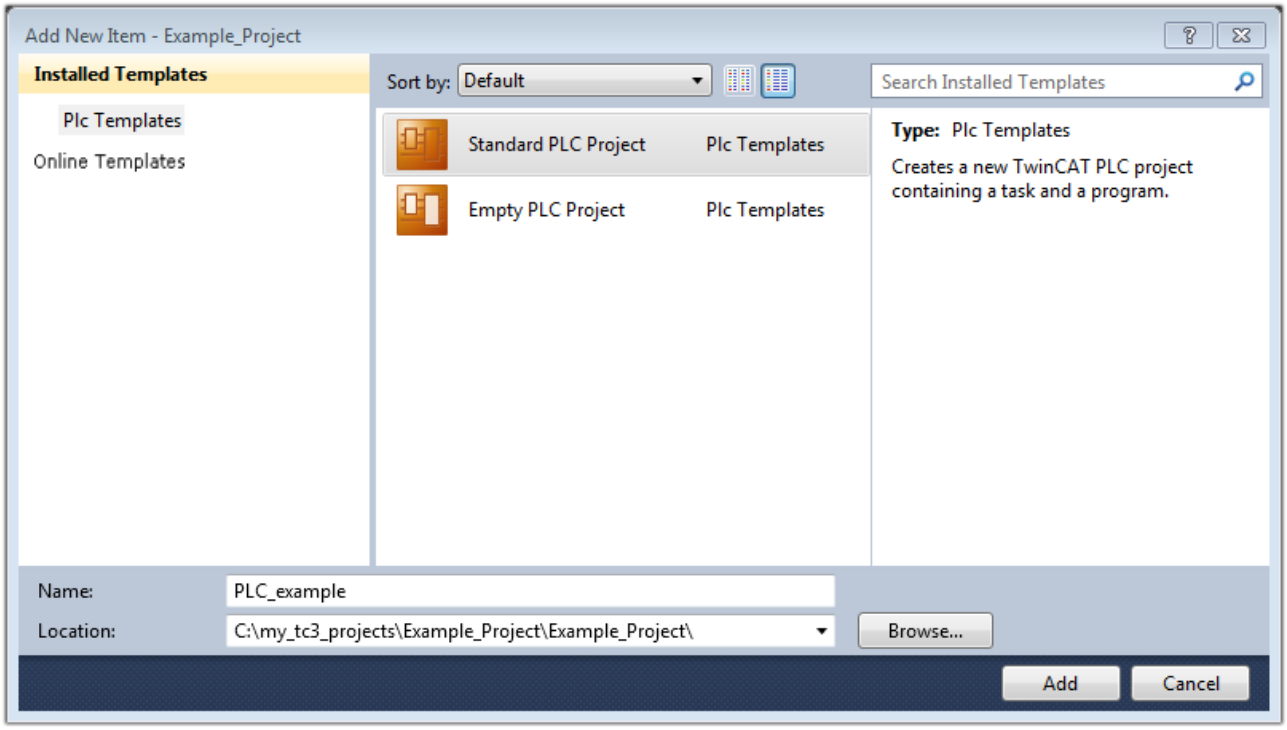


Fig. 61: Specifying the name and directory for the PLC programming environment

The “Main” program, which already exists by selecting “Standard PLC project”, can be opened by double-clicking on “PLC_example_project” in “POUs”. The following user interface is shown for an initial project:

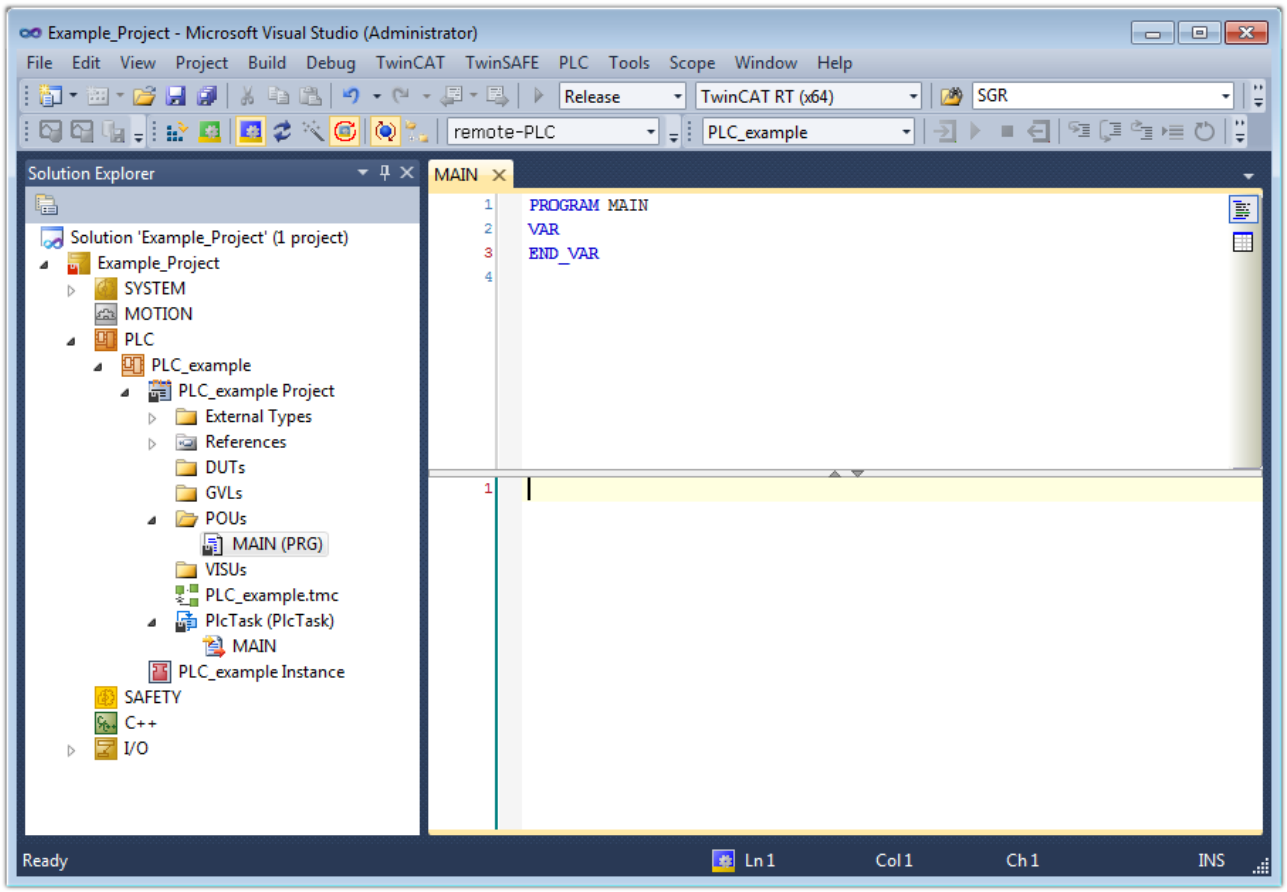


Fig. 62: Initial “Main” program of the standard PLC project

To continue, sample variables and a sample program have now been created:

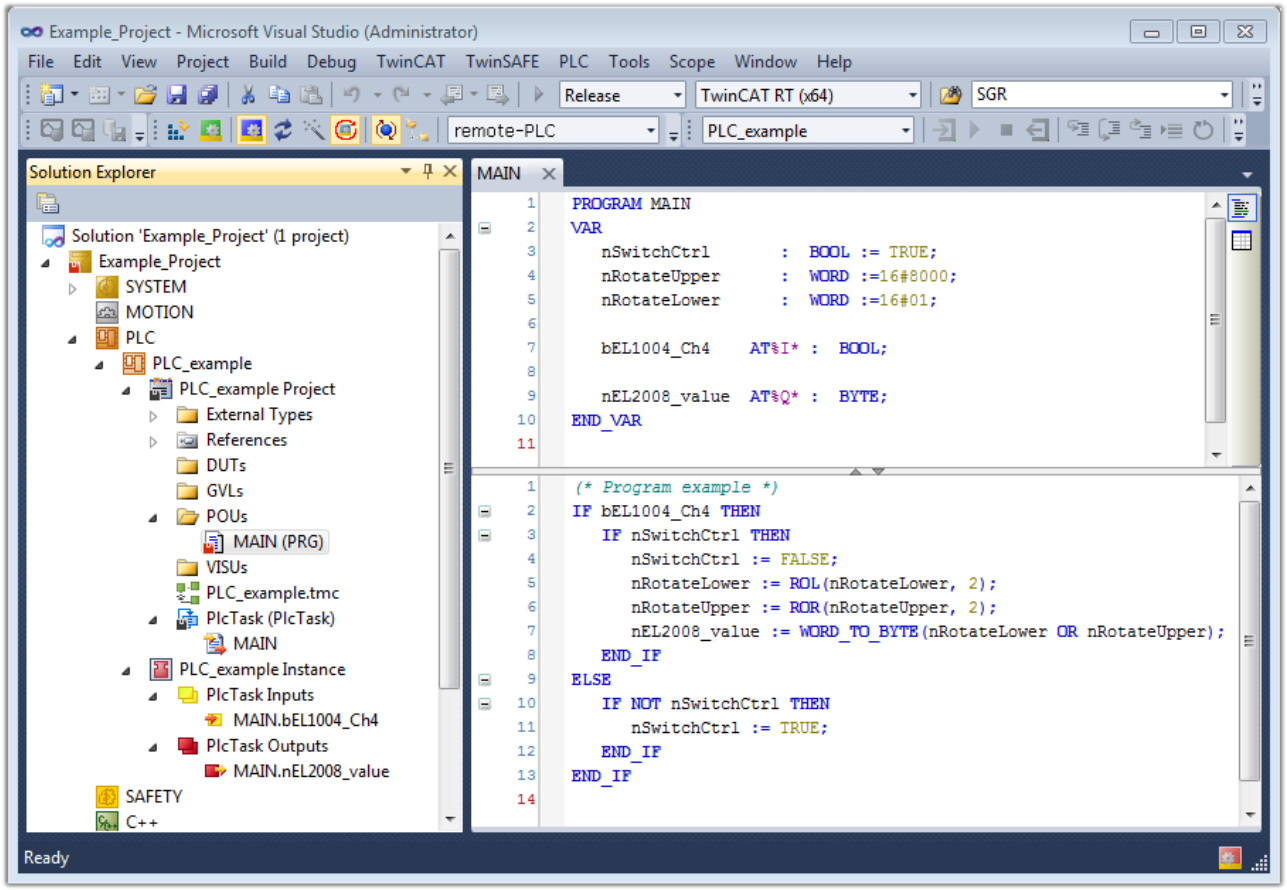


Fig. 63: Sample program with variables after a compile process (without variable integration)

The control program is now created as a project folder, followed by the compile process:

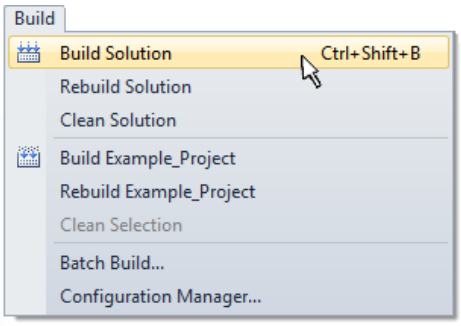
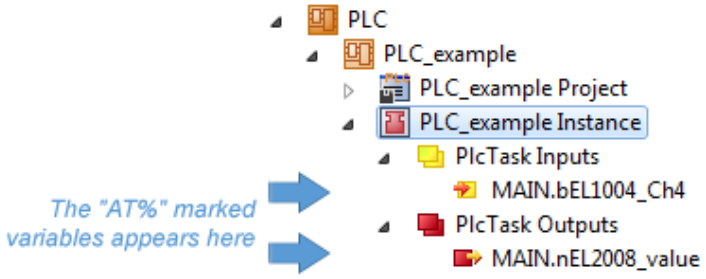


Fig. 64: Start program compilation

The following variables, identified in the ST/ PLC program with “AT%”, are then available in under “Assignments” in the project folder explorer:



Assigning variables

Via the menu of an instance - variables in the “PLC” context, use the “Modify Link…” option to open a window for selecting a suitable process object (PDO) for linking:

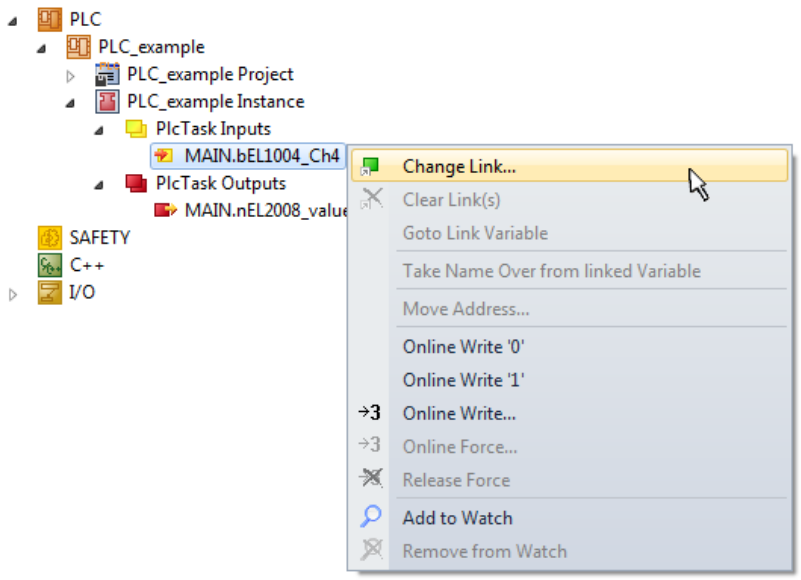


Fig. 65: Creating the links between PLC variables and process objects

In the window that opens, the process object for the variable “bEL1004_Ch4” of type BOOL can be selected from the PLC configuration tree:

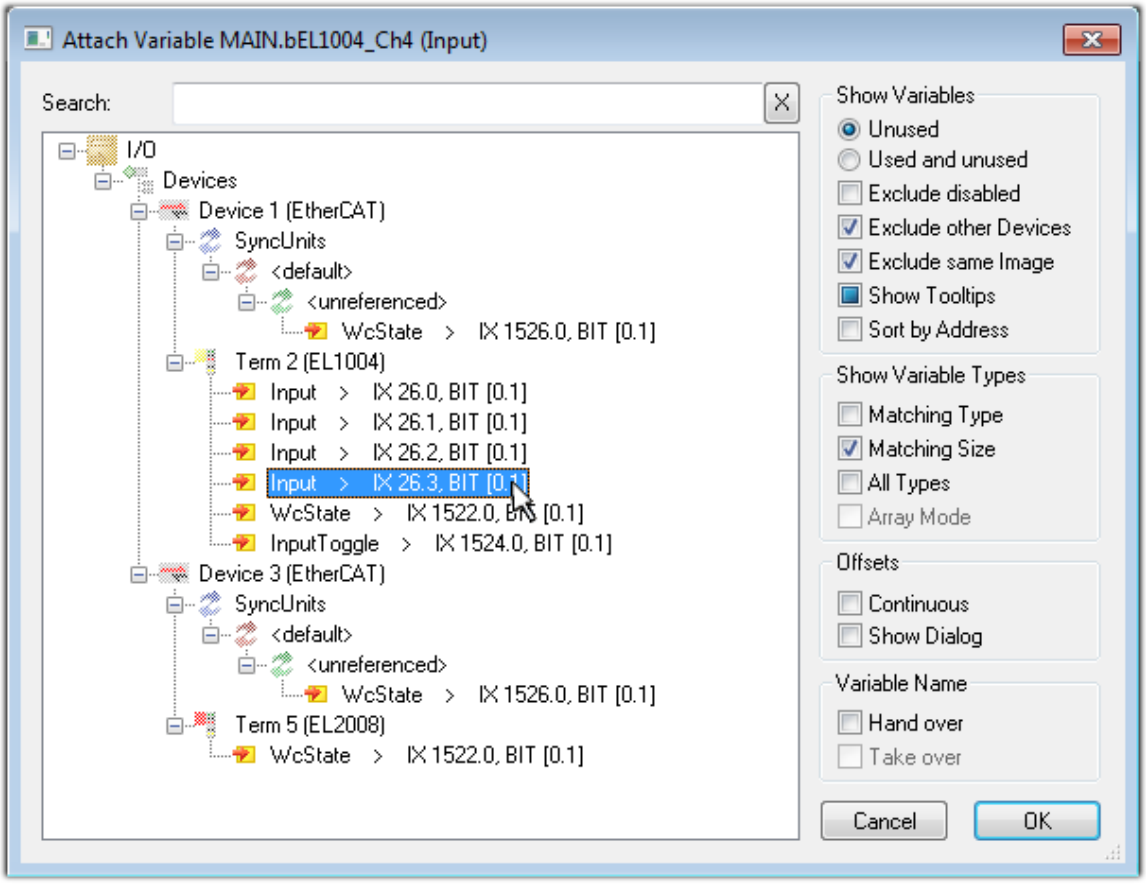


Fig. 66: Selecting PDO of type BOOL

According to the default setting, certain PDO objects are now available for selection. In this sample the input of channel 4 of the EL1004 terminal is selected for linking. In contrast, the checkbox “All types” must be ticked for creating the link for the output variables, in order to allocate a set of eight separate output bits to a byte variable. The following diagram shows the whole process:

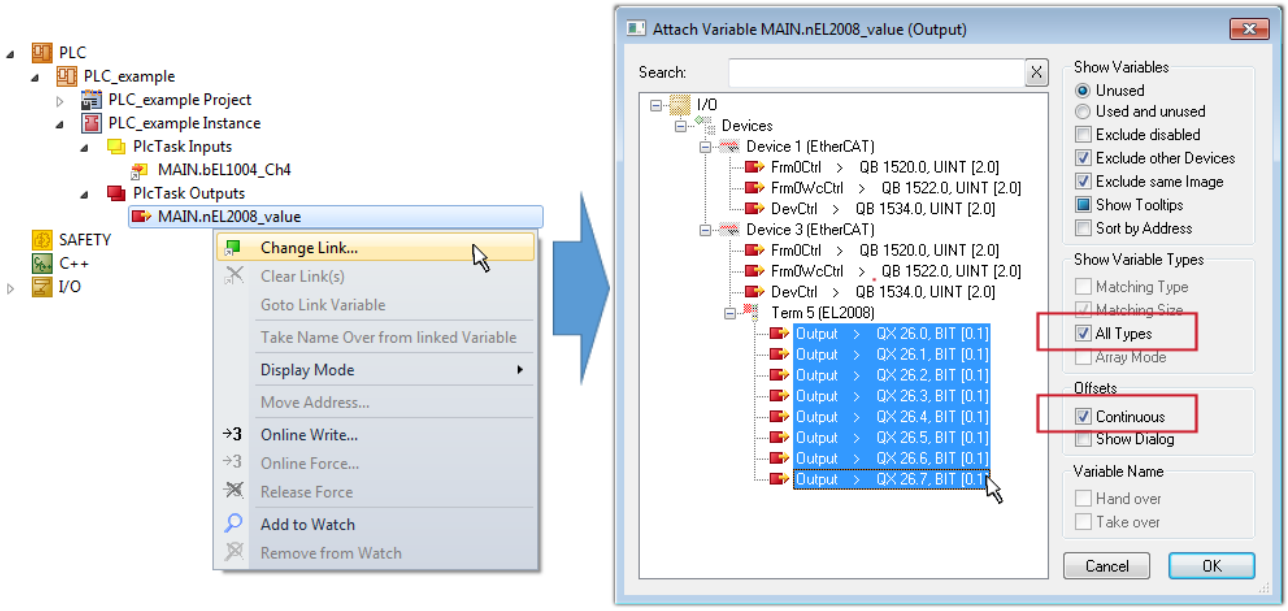


Fig. 67: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the variable “nEL2008_value” sequentially to all eight selected output bits of the EL2008 terminal. In this way it is possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) at the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting a “Goto Link Variable” from the context menu of a variable. The object opposite, in this case the PDO, is automatically selected:

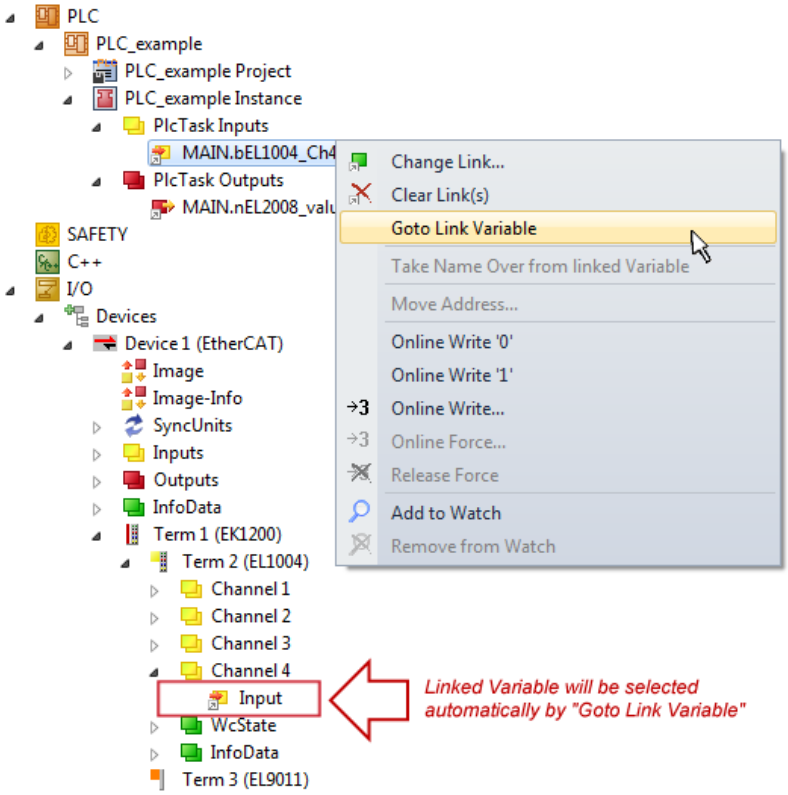


Fig. 68: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

The process of creating links can also take place in the opposite direction, i.e. starting with individual PDOs to variable. However, in this example it would then not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word, integer or

similar PDO, it is possible to allocate this a set of bit-standardized variables (type “BOOL”). Here, too, a “Goto Link Variable” from the context menu of a PDO can be executed in the other direction, so that the respective PLC instance can then be selected.

Note on the type of variable assignment

The following type of variable assignment can only be used from TwinCAT version V3.1.4024.4 onwards and is only available for terminals with a microcontroller.

In TwinCAT it is possible to create a structure from the mapped process data of a terminal. An instance of this structure can then be created in the PLC, so it is possible to access the process data directly from the PLC without having to declare own variables.

The procedure for the EL3001 1-channel analog input terminal -10...+10 V is shown as an example.

1. First the required process data must be selected in the “Process data” tab in TwinCAT.
2. After that, the PLC data type must be generated in the tab “PLC” via the check box.
3. The data type in the “Data Type” field can then be copied using the “Copy” button.

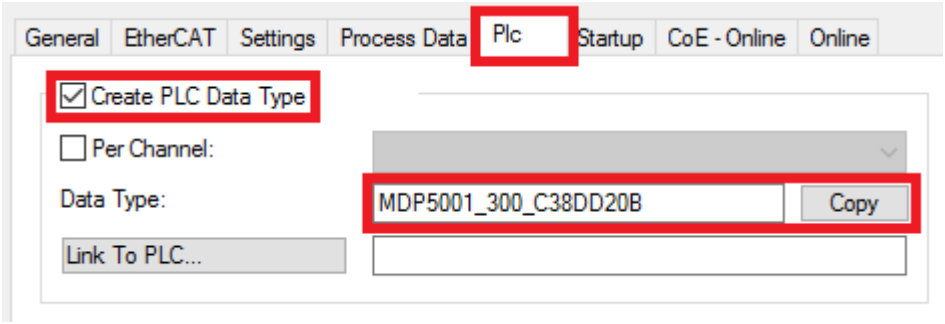


Fig. 69: Creating a PLC data type

4. An instance of the data structure of the copied data type must then be created in the PLC.

```
MAIN
1  PROGRAM MAIN
2  VAR
3      EL3001  : MDP5001_300_C38DD20B;
4  END_VAR
```

Fig. 70: Instance_of_struct

5. Then the project folder must be created. This can be done either via the key combination “CTRL + Shift + B” or via the “Build” tab in TwinCAT.
6. The structure in the “PLC” tab of the terminal must then be linked to the created instance.

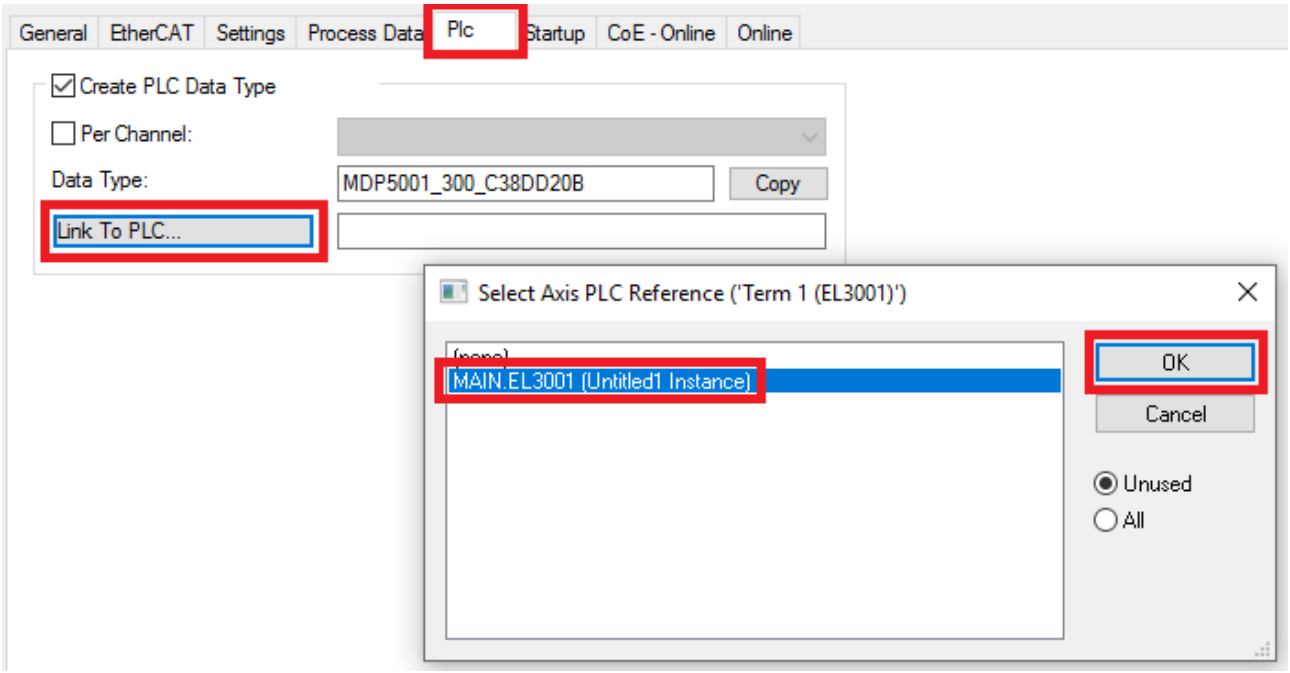


Fig. 71: Linking the structure

7. In the PLC the process data can then be read or written via the structure in the program code.

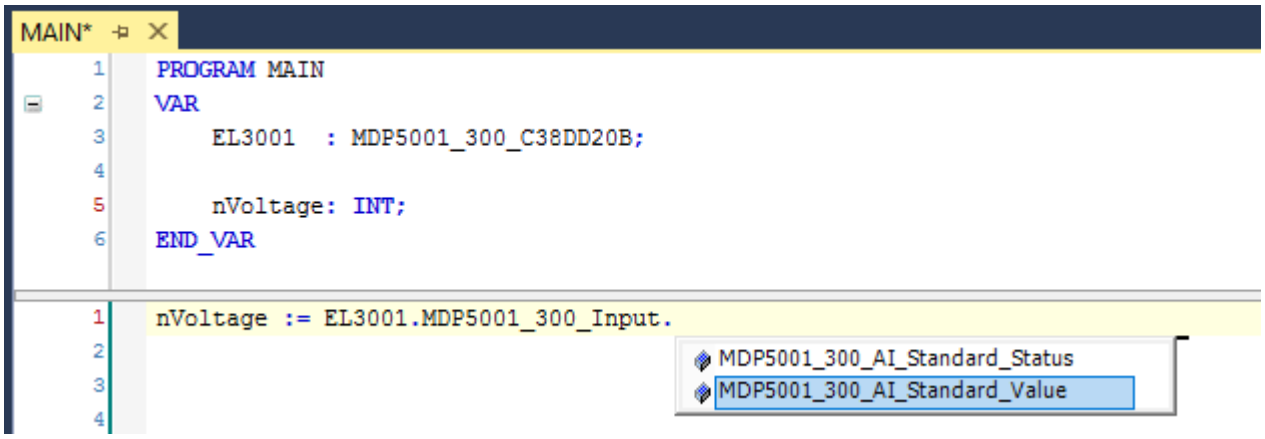
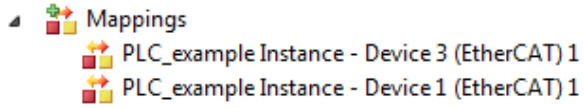


Fig. 72: Reading a variable from the structure of the process data

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated with  or via the menu under “TwinCAT” in order to transfer settings of the development environment to the runtime system. Confirm the messages “Old configurations are overwritten!” and “Restart TwinCAT system in Run mode” with “OK”. The corresponding assignments can be seen in the project folder explorer:



A few seconds later the corresponding status of the Run mode is displayed in the form of a rotating symbol  at the bottom right of the VS shell development environment. The PLC system can then be started as described below.

Starting the controller

Select the menu option “PLC” → “Login” or click on to link the PLC with the real-time system and load the control program for execution. This results in the message *No program on the controller! Should the new program be loaded?,* which should be acknowledged with “Yes”. The runtime environment is ready for program start by click on symbol , the “F5” key or via “PLC” in the menu selecting “Start”. The started programming environment shows the runtime values of individual variables:

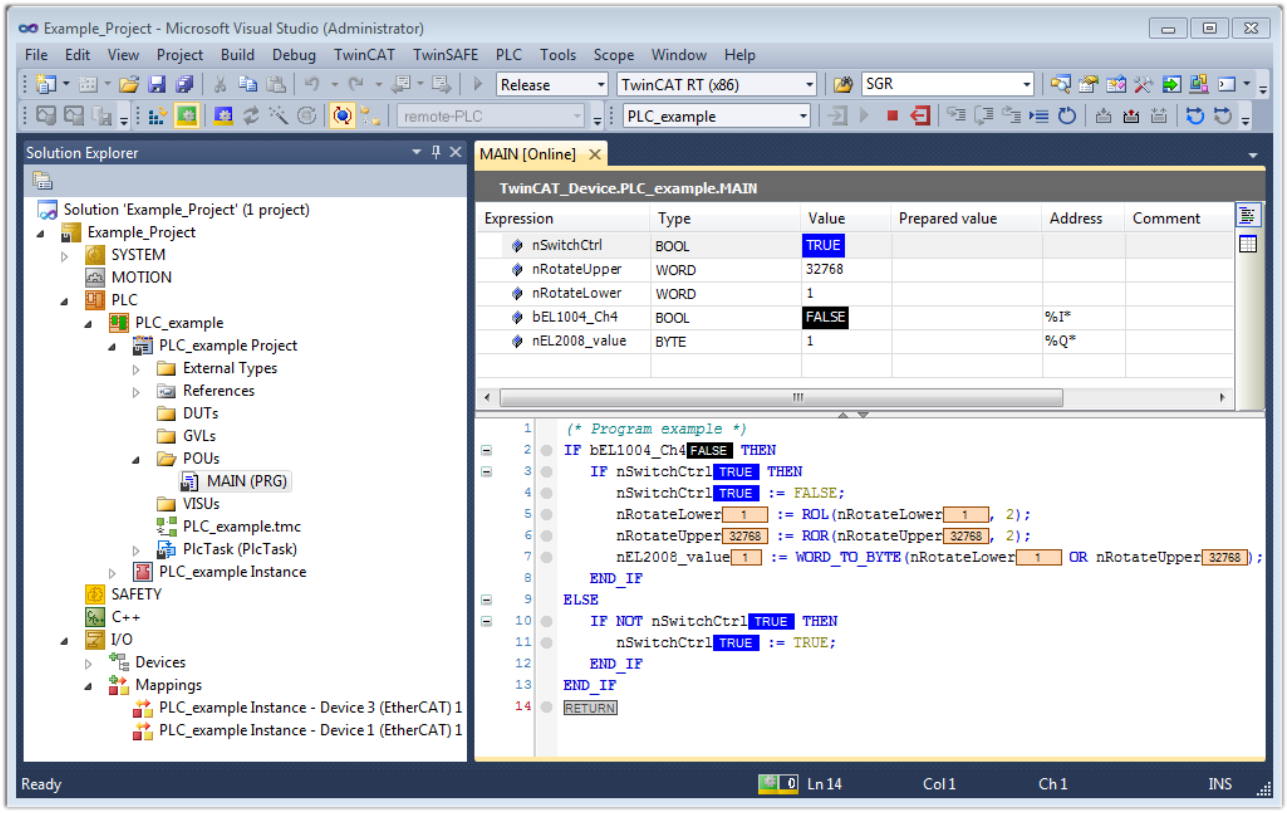


Fig. 73: TwinCAT development environment (VS shell): logged-in, after program startup

The two operator control elements for stopping and logout result in the required action (accordingly also for stop “Shift + F5”, or both actions can be selected via the PLC menu).

5.2 TwinCAT Development Environment

The Software for automation TwinCAT (The Windows Control and Automation Technology) will be distinguished into:

- TwinCAT 2: System Manager (Configuration) & PLC Control (Programming)
- TwinCAT 3: Enhancement of TwinCAT 2 (Programming and Configuration takes place via a common Development Environment)

Details:

- **TwinCAT 2:**
  - Connects I/O devices to tasks in a variable-oriented manner
  - Connects tasks to tasks in a variable-oriented manner
  - Supports units at the bit level
  - Supports synchronous or asynchronous relationships
  - Exchange of consistent data areas and process images

- Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)
- Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/ 2000/XP/Vista, Windows 7, NT/XP Embedded, CE
- Interconnection to all common fieldbusses
- More...

Additional features:

- **TwinCAT 3** (eXtended Automation)**:**
  - Visual-Studio®-Integration
  - Choice of the programming language
  - Supports object orientated extension of IEC 61131-3
  - Usage of C/C++ as programming language for real time applications
  - Connection to MATLAB®/Simulink®
  - Open interface for expandability
  - Flexible run-time environment
  - Active support of Multi-Core- und 64-Bit-Operatingsystem
  - Automatic code generation and project creation with the TwinCAT Automation Interface
  - More...

Within the following sections commissioning of the TwinCAT Development Environment on a PC System for the control and also the basically functions of unique control elements will be explained.

Please see further information to TwinCAT 2 and TwinCAT 3 at http://infosys.beckhoff.com.

5.2.1 Installation of the TwinCAT real-time driver

In order to assign real-time capability to a standard Ethernet port of an IPC controller, the Beckhoff real-time driver has to be installed on this port under Windows.

This can be done in several ways. One option is described here.

In the System Manager call up the TwinCAT overview of the local network interfaces via Options → Show Real Time Ethernet Compatible Devices.

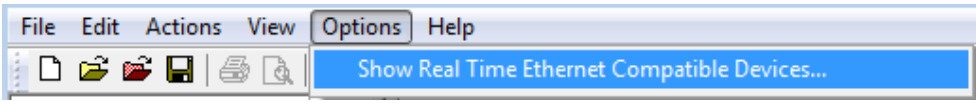


Fig. 74: System Manager “Options” (TwinCAT 2)

This have to be called up by the Menü “TwinCAT” within the TwinCAT 3 environment:

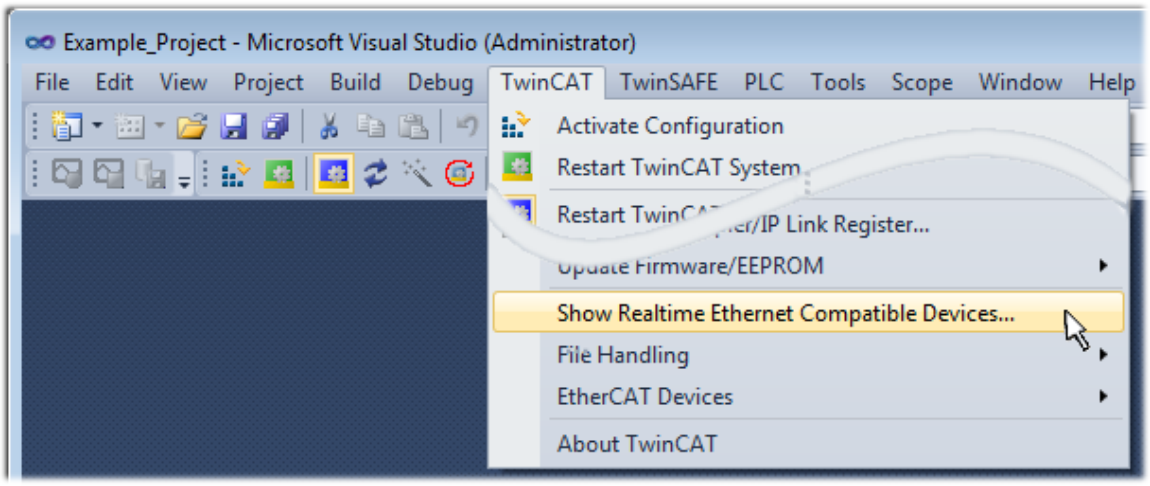


Fig. 75: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

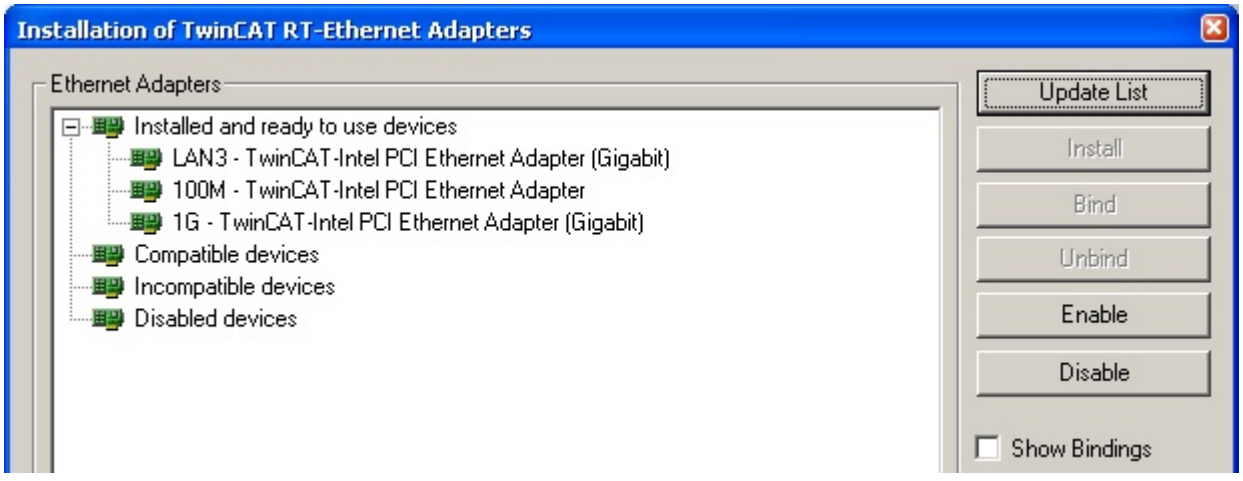


Fig. 76: Overview of network interfaces

Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter Offline configuration creation, section “Creating the EtherCAT device” [▶ 83] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):

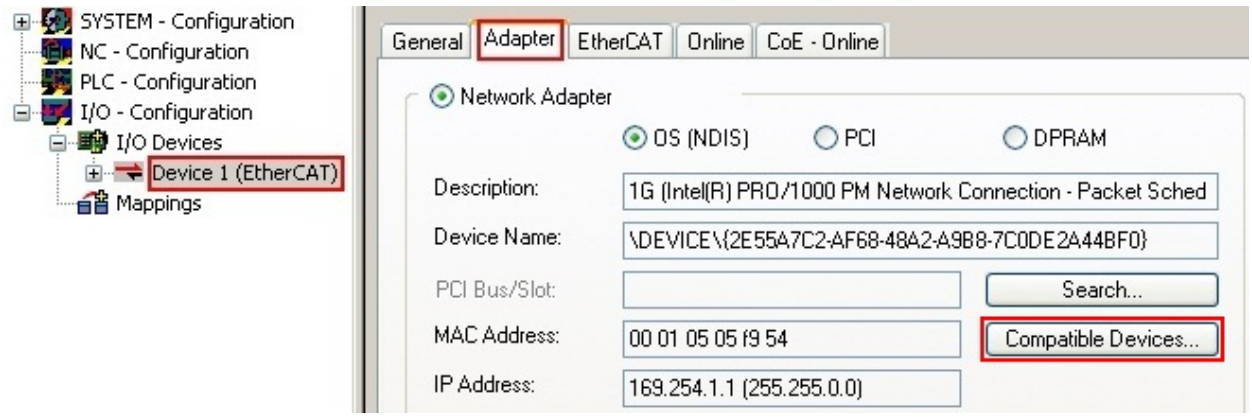
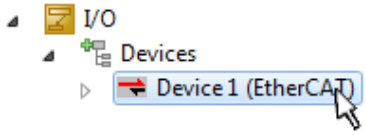


Fig. 77: EtherCAT device properties(TwinCAT 2): click on “Compatible Devices…” of tab “Adapte””

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on “Device .. (EtherCAT)” within the Solution Explorer under “I/O”:



After the installation the driver appears activated in the Windows overview for the network interface (Windows Start → System Properties → Network)

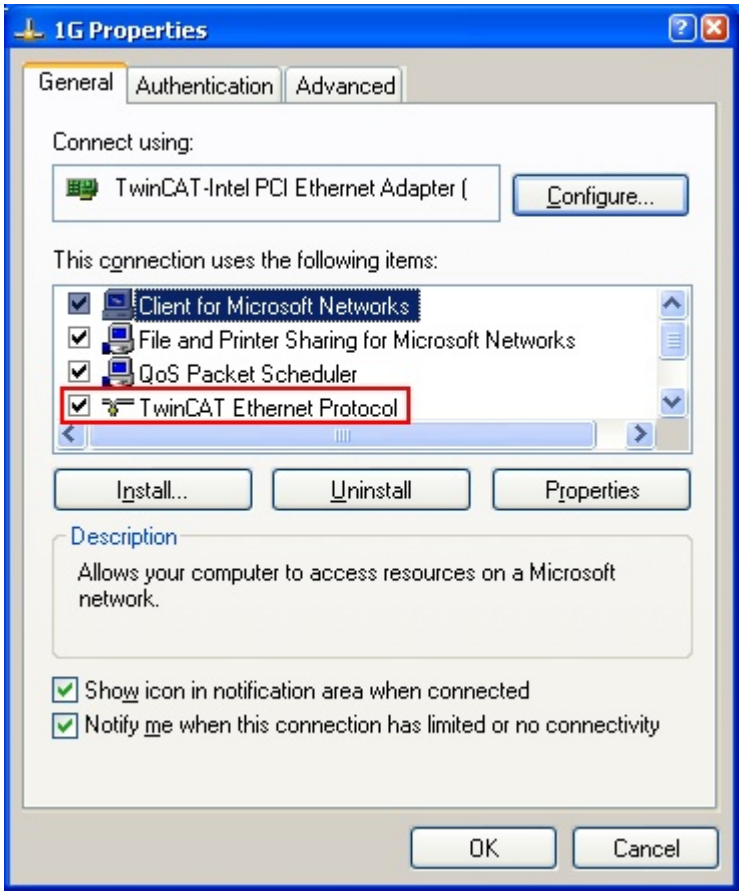


Fig. 78: Windows properties of the network interface

A correct setting of the driver could be:

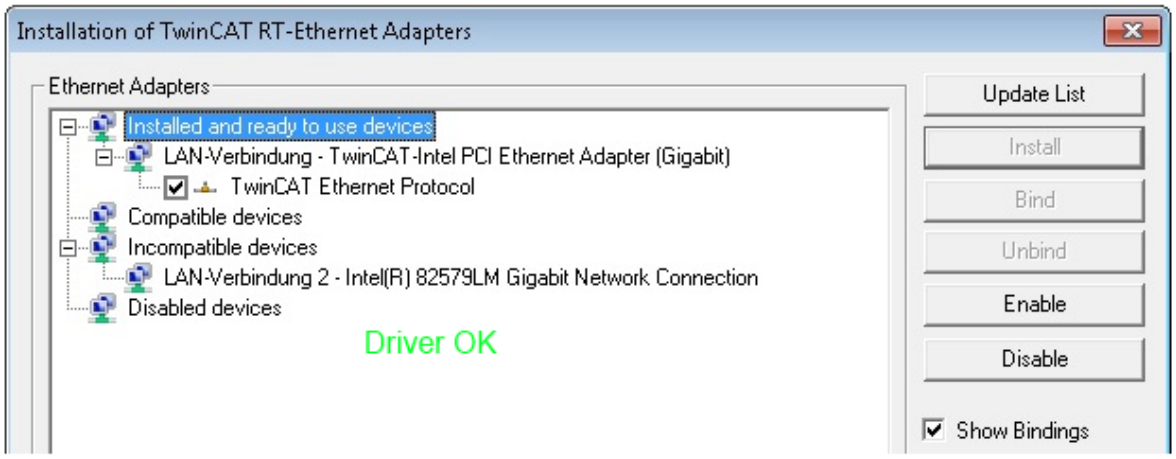


Fig. 79: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

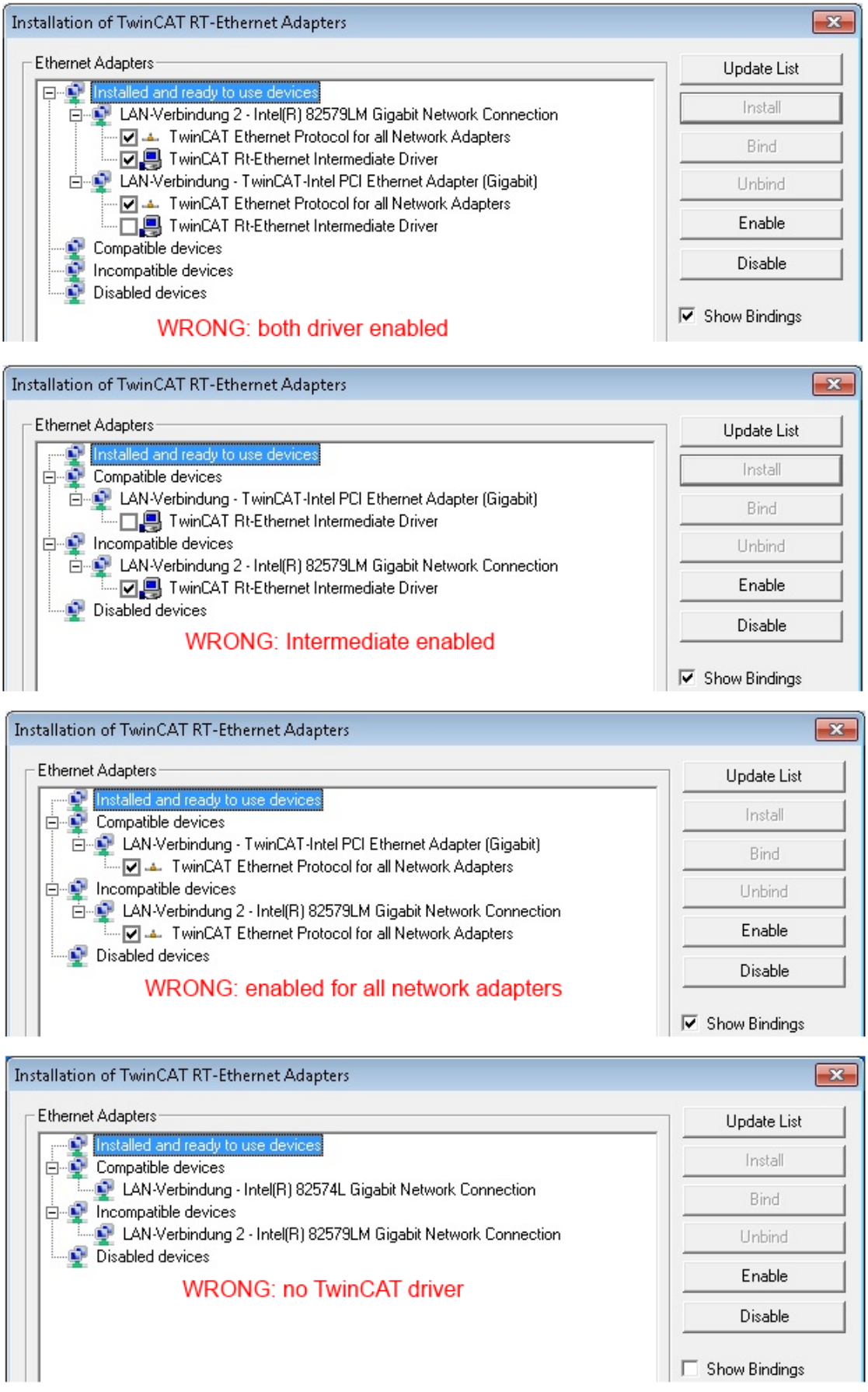


Fig. 80: Incorrect driver settings for the Ethernet port

IP address of the port used

IP address/DHCP

In most cases an Ethernet port that is configured as an EtherCAT device will not transport general IP packets. For this reason and in cases where an EL6601 or similar devices are used it is useful to specify a fixed IP address for this port via the “Internet Protocol TCP/IP” driver setting and to disable DHCP. In this way the delay associated with the DHCP client for the Ethernet port assigning itself a default IP address in the absence of a DHCP server is avoided. A suitable address space is 192.168.x.x, for example.

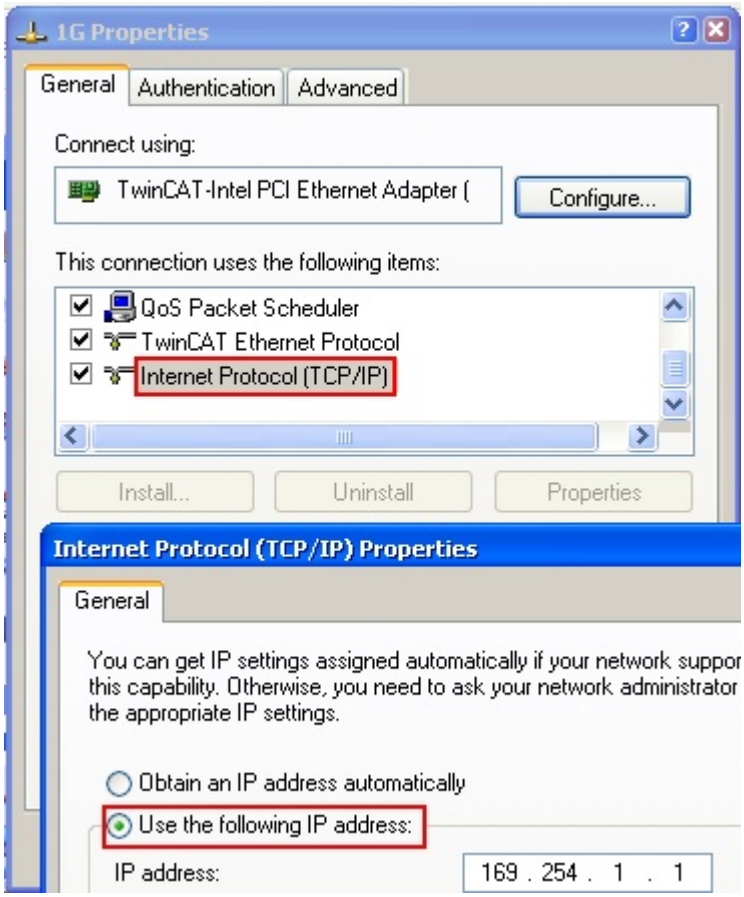


Fig. 81: TCP/IP setting for the Ethernet port

5.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

The TwinCAT EtherCAT master/System Manager needs the device description files for the devices to be used in order to create the configuration in online or offline mode. The device descriptions are contained in the so-called ESI files (EtherCAT Slave Information) in XML format. These files can be requested from the respective vendor and are made available for download. An *.xml file may contain several device descriptions.

The ESIs for Beckhoff EtherCAT devices are provided on the Beckhoff website.

The ESI files must be stored in the TwinCAT installation directory.

Default settings:

- **TwinCAT 2**: C:\TwinCAT\IO\EtherCAT
- **TwinCAT 3**: C:\TwinCAT\3.1\Config\Io\EtherCAT

The files are read (once) when a new System Manager window is opened, if they have changed since the last time the System Manager window was opened.

A TwinCAT installation includes the set of Beckhoff ESI files that was current at the time when the TwinCAT build was created.

From TwinCAT 2.11 / TwinCAT 3 on the ESI directory can be updated from the System Manager if the programming PC is connected to the internet; at

TwinCAT 2: Options → „Update EtherCAT Device Descriptions“

TwinCAT 3: TwinCAT → EtherCAT Devices → “Update Device Descriptions (via ETG Website)…”

The TwinCAT ESI Updater is available for this purpose.

ESI

The *.xml files are associated with *.xsd files, which describe the structure of the ESI XML files. To update the ESI device descriptions, both file types should therefore be updated.

Device differentiation

EtherCAT devices/slaves are distinguished by four properties, which determine the full device identifier. For example, the device identifier EL2521-0025-1018 consists of:

- family key “EL”
- name “2521”
- type “0025”
- and revision “1018”

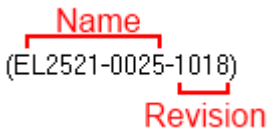


Fig. 82: Identifier structure

The order identifier consisting of name + type (here: EL2521-0010) describes the device function. The revision indicates the technical progress 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. Each revision has its own ESI description. See further notes.

Online description

If the EtherCAT configuration is created online through scanning of real devices (see section Online setup) and no ESI descriptions are available for a slave (specified by name and revision) that was found, the System Manager asks whether the description stored in the device should be used. In any case, the System Manager needs this information for setting up the cyclic and acyclic communication with the slave correctly.

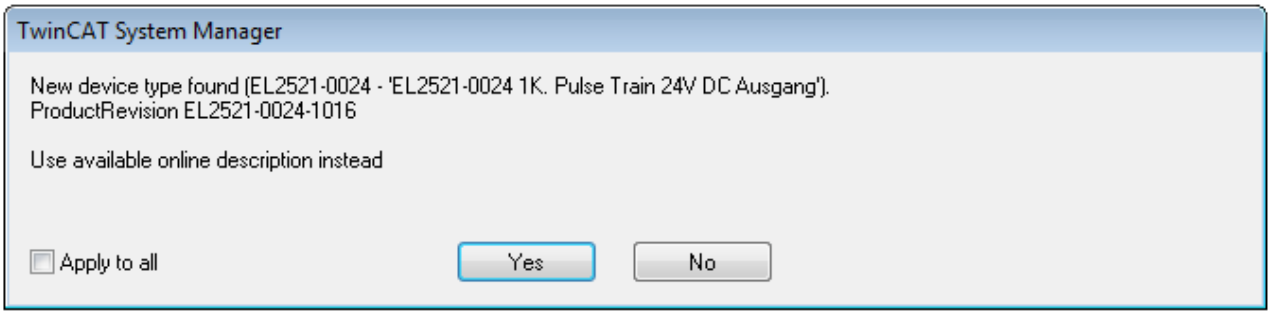


Fig. 83: OnlineDescription information window (TwinCAT 2)

In TwinCAT 3 a similar window appears, which also offers the Web update:

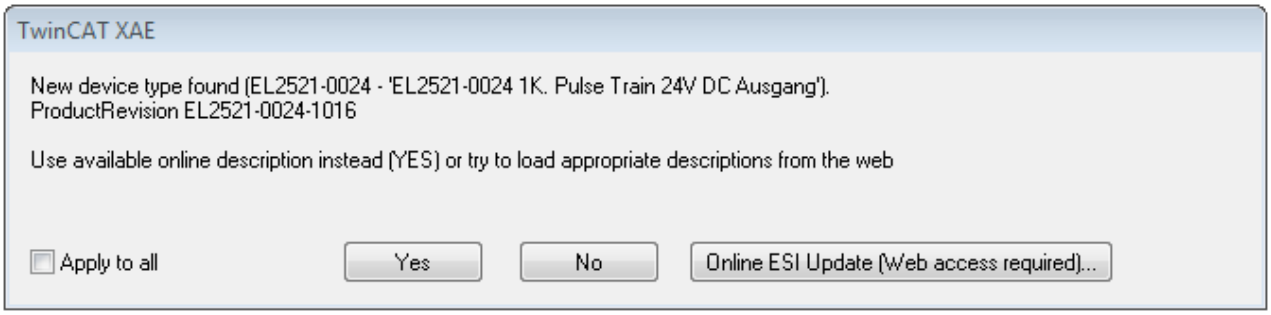


Fig. 84: Information window OnlineDescription (TwinCAT 3)

If possible, the *Yes* is to be rejected and the required ESI is to be requested from the device manufacturer. After installation of the XML/XSD file the configuration process should be repeated.

NOTICE

Changing the "usual" configuration through a scan

✓ If a scan discovers a device that is not yet known to TwinCAT, distinction has to be made between two cases. Taking the example here of the EL2521-0000 in the revision 1019

a) no ESI is present for the EL2521-0000 device at all, either for the revision 1019 or for an older revision. The ESI must then be requested from the manufacturer (in this case Beckhoff).

b) an ESI is present for the EL2521-0000 device, but only in an older revision, e.g. 1018 or 1017. In this case an in-house check should first be performed to determine whether the spare parts stock allows the integration of the increased revision into the configuration at all. A new/higher revision usually also brings along new features. If these are not to be used, work can continue without reservations with the previous revision 1018 in the configuration. This is also stated by the Beckhoff compatibility rule.

Refer in particular to the chapter "General notes on the use of Beckhoff EtherCAT IO components" and for manual configuration to the chapter "Offline configuration creation".

If the OnlineDescription is used regardless, the System Manager reads a copy of the device description from the EEPROM in the EtherCAT slave. In complex slaves the size of the EEPROM may not be sufficient for the complete ESI, in which case the ESI would be *incomplete* in the configurator. Therefore it's recommended using an offline ESI file with priority in such a case.

The System Manager creates for online recorded device descriptions a new file "OnlineDescription0000...xml" in its ESI directory, which contains all ESI descriptions that were read online.

Fig. 85: File OnlineDescription.xml created by the System Manager

Is a slave desired to be added manually to the configuration at a later stage, online created slaves are indicated by a prepended symbol “>” in the selection list (see Figure *Indication of an online recorded ESI of EL2521 as an example)*.

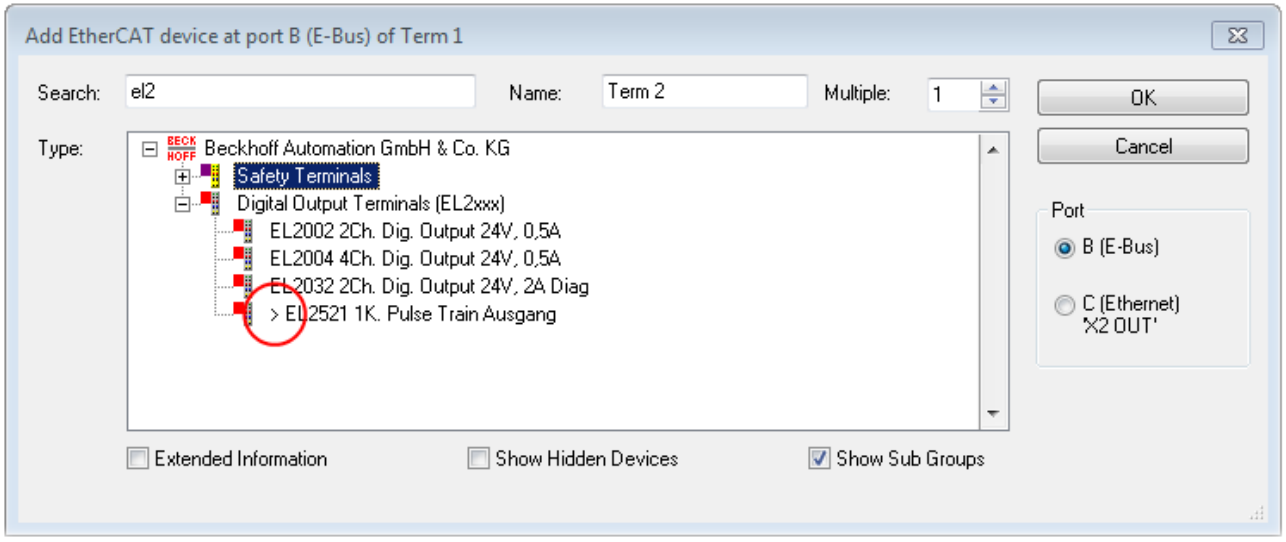


Fig. 86: Indication of an online recorded ESI of EL2521 as an example

If such ESI files are used and the manufacturer's files become available later, the file OnlineDescription.xml should be deleted as follows:

- close all System Manager windows
- restart TwinCAT in Config mode
- delete “OnlineDescription0000...xml”
- restart TwinCAT System Manager

This file should not be visible after this procedure, if necessary press <F5> to update

OnlineDescription for TwinCAT 3.x

In addition to the file described above “OnlineDescription0000...xml”, a so called EtherCAT cache with new discovered devices is created by TwinCAT 3.x, e.g. under Windows 7:

C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml

(Please note the language settings of the OS!)
You have to delete this file, too.

Faulty ESI file

If an ESI file is faulty and the System Manager is unable to read it, the System Manager brings up an information window.

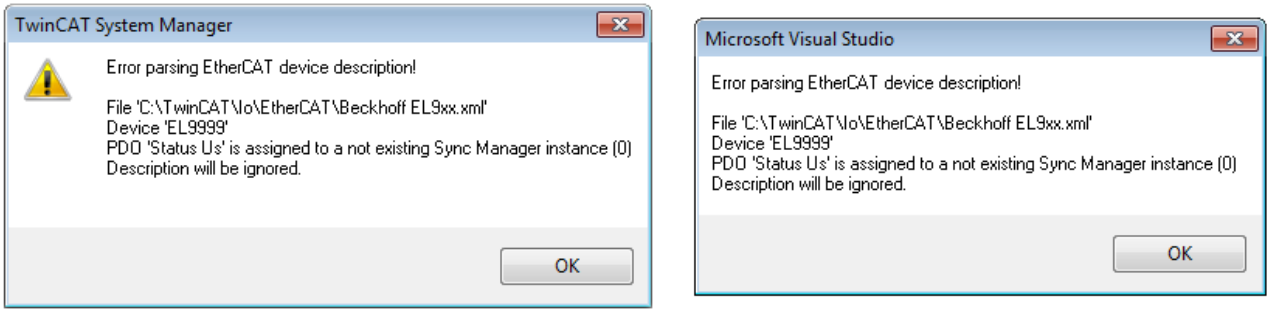


Fig. 87: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

- Structure of the *.xml does not correspond to the associated *.xsd file → check your schematics

- Contents cannot be translated into a device description → contact the file manufacturer

5.2.3 TwinCAT ESI Updater

For TwinCAT 2.11 and higher, the System Manager can search for current Beckhoff ESI files automatically, if an online connection is available:

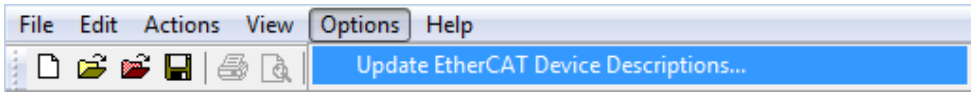


Fig. 88: Using the ESI Updater (>= TwinCAT 2.11)

The call up takes place under:
“Options” → “Update EtherCAT Device Descriptions”

Selection under TwinCAT 3:

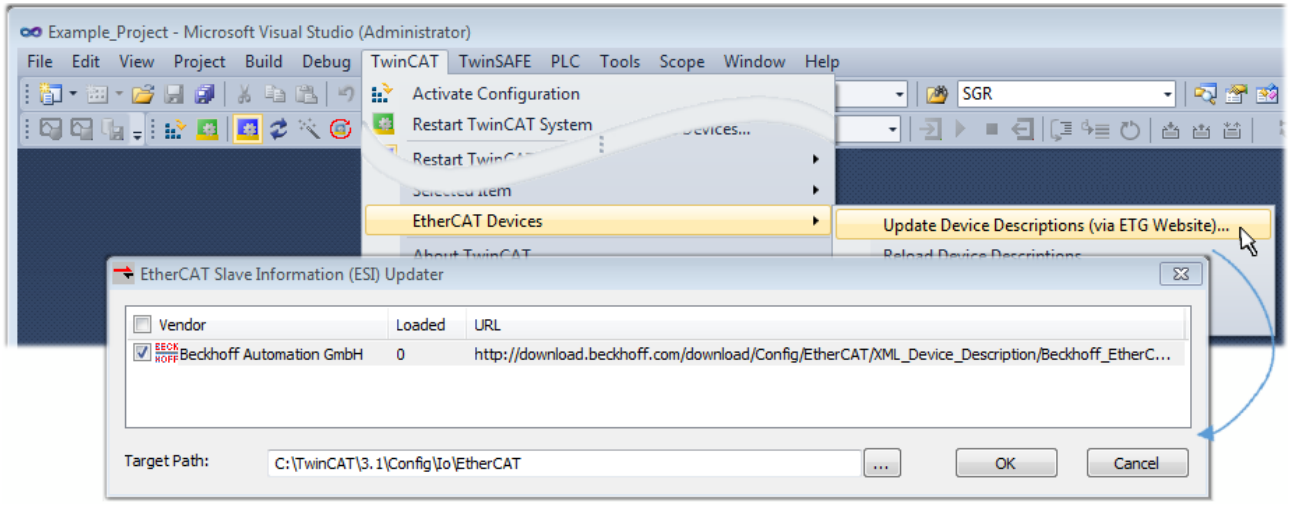


Fig. 89: Using the ESI Updater (TwinCAT 3)

The ESI Updater (TwinCAT 3) is a convenient option for automatic downloading of ESI data provided by EtherCAT manufacturers via the Internet into the TwinCAT directory (ESI = EtherCAT slave information). TwinCAT accesses the central ESI ULR directory list stored at ETG; the entries can then be viewed in the Updater dialog, although they cannot be changed there.

The call up takes place under:
“TwinCAT” → “EtherCAT Devices” → “Update Device Description (via ETG Website)…”.

5.2.4 Distinction between Online and Offline

The distinction between online and offline refers to the presence of the actual I/O environment (drives, terminals, EJ-modules). If the configuration is to be prepared in advance of the system configuration as a programming system, e.g. on a laptop, this is only possible in “Offline configuration” mode. In this case all components have to be entered manually in the configuration, e.g. based on the electrical design.

If the designed control system is already connected to the EtherCAT system and all components are energised and the infrastructure is ready for operation, the TwinCAT configuration can simply be generated through “scanning” from the runtime system. This is referred to as online configuration.

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note “Installation of the latest ESI-XML device description”.

For preparation of a configuration:

- the real EtherCAT hardware (devices, couplers, drives) must be present and installed
- the devices/modules must be connected via EtherCAT cables or in the terminal/ module strand in the same way as they are intended to be used later
- the devices/modules be connected to the power supply and ready for communication

- TwinCAT must be in CONFIG mode on the target system.

The online scan process consists of:

- detecting the EtherCAT device [▶ 88] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 89]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 92]

The scan with existing configuration [▶ 93] can also be carried out for comparison.

5.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

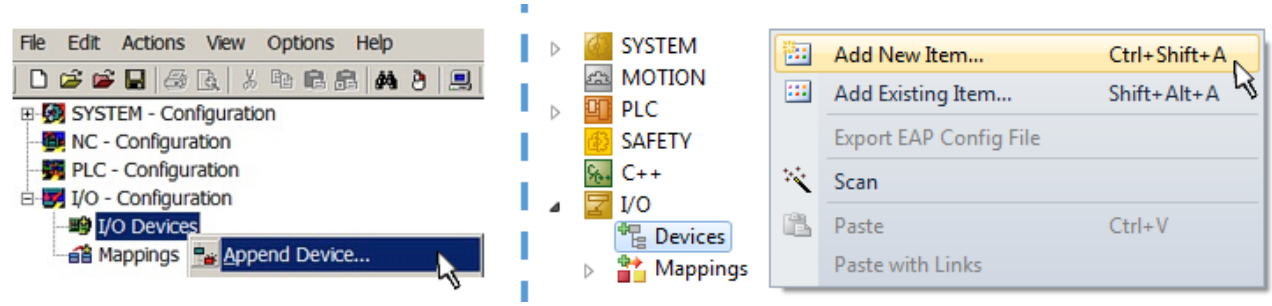


Fig. 90: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

Select type “EtherCAT” for an EtherCAT I/O application with EtherCAT slaves. For the present publisher/subscriber service in combination with an EL6601/EL6614 terminal select “EtherCAT Automation Protocol via EL6601”.

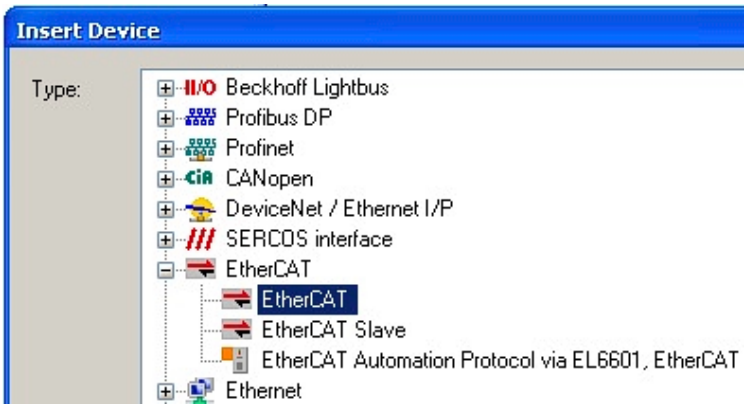


Fig. 91: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

Then assign a real Ethernet port to this virtual device in the runtime system.

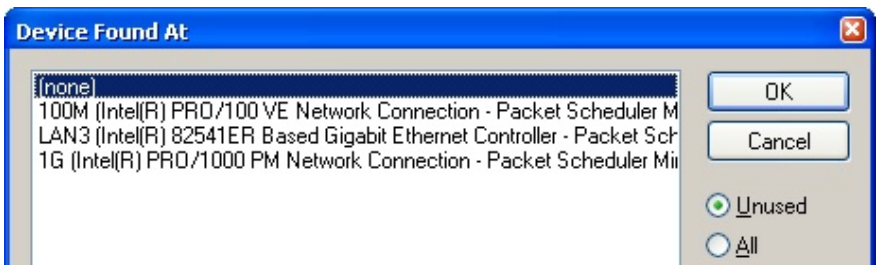


Fig. 92: Selecting the Ethernet port

This query may appear automatically when the EtherCAT device is created, or the assignment can be set/ modified later in the properties dialog; see Fig. “EtherCAT device properties (TwinCAT 2)”.

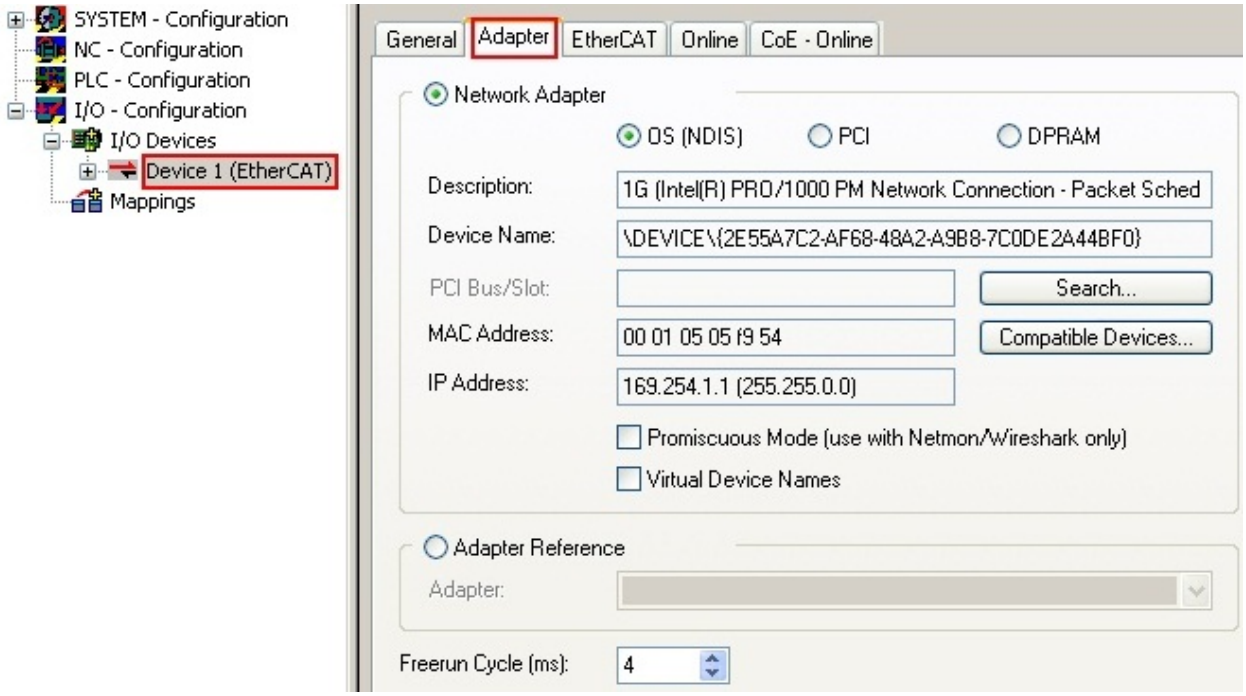
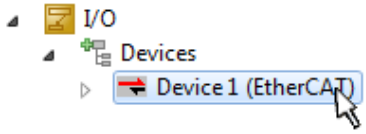


Fig. 93: EtherCAT device properties (TwinCAT 2)

TwinCAT 3: the properties of the EtherCAT device can be opened by double click on “Device .. (EtherCAT)” within the Solution Explorer under “I/O”:



Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page.

Defining EtherCAT slaves

Further devices can be appended by right-clicking on a device in the configuration tree.

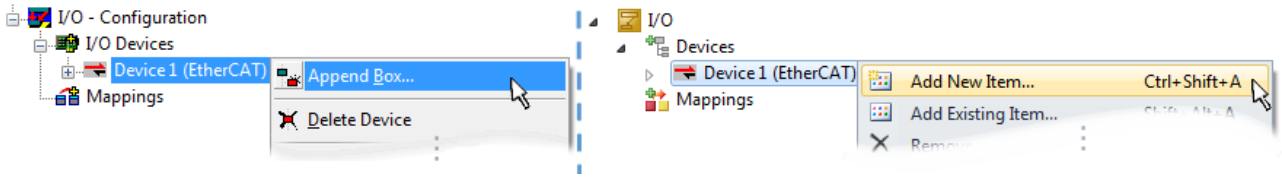


Fig. 94: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

The dialog for selecting a new device opens. Only devices for which ESI files are available are displayed.

Only devices are offered for selection that can be appended to the previously selected device. Therefore the physical layer available for this port is also displayed (Fig. “Selection dialog for new EtherCAT device”, A). In the case of cable-based Fast-Ethernet physical layer with PHY transfer, then also only cable-based devices are available, as shown in Fig. “Selection dialog for new EtherCAT device”. If the preceding device has several free ports (e.g. EK1122 or EK1100), the required port can be selected on the right-hand side (A).

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: EK couplers, EP boxes, devices with RJ45/M8/M12 connector

- “E-Bus”: LVDS “terminal bus”, “EJ-module”: EL/ES terminals, various modular modules

The search field facilitates finding specific devices (since TwinCAT 2.11 or TwinCAT 3).

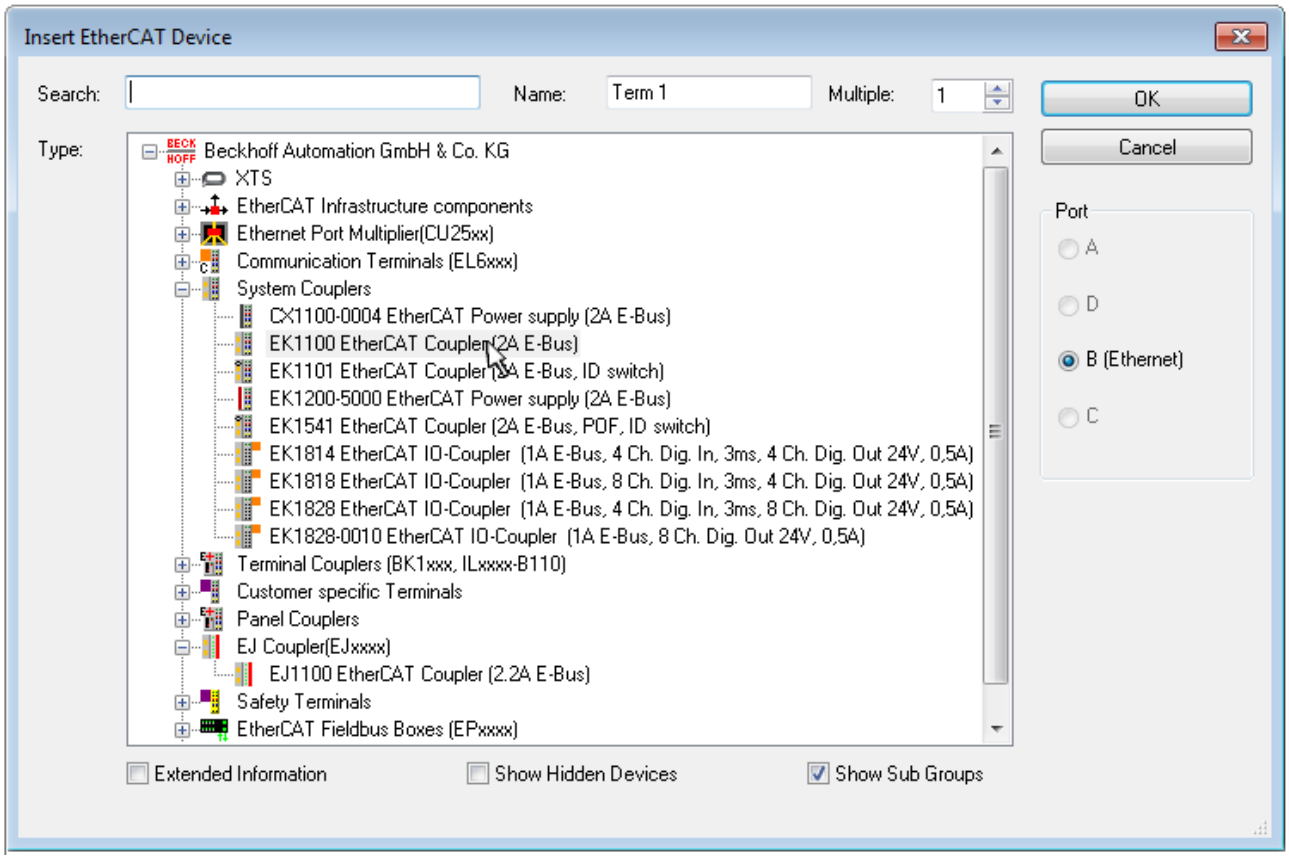


Fig. 95: Selection dialog for new EtherCAT device

By default only the name/device type is used as selection criterion. For selecting a specific revision of the device the revision can be displayed as “Extended Information”.

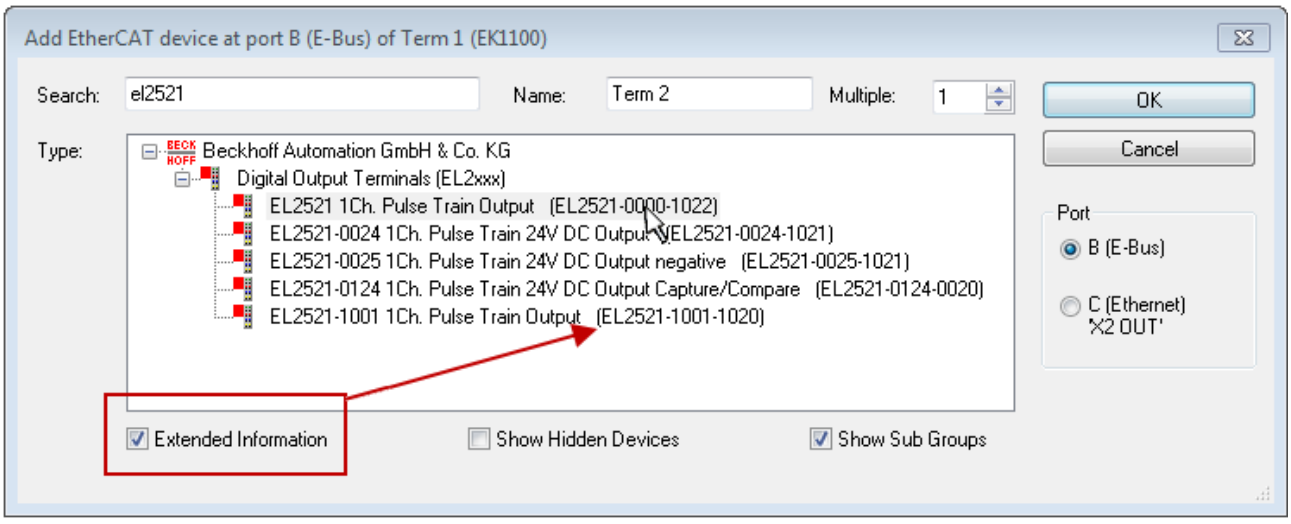


Fig. 96: Display of device revision

In many cases several device revisions were created for historic or functional reasons, e.g. through technological advancement. For simplification purposes (see Fig. “Selection dialog for new EtherCAT device”) only the last (i.e. highest) revision and therefore the latest state of production is displayed in the selection dialog for Beckhoff devices. To show all device revisions available in the system as ESI descriptions tick the “Show Hidden Devices” check box, see Fig. “Display of previous revisions”.

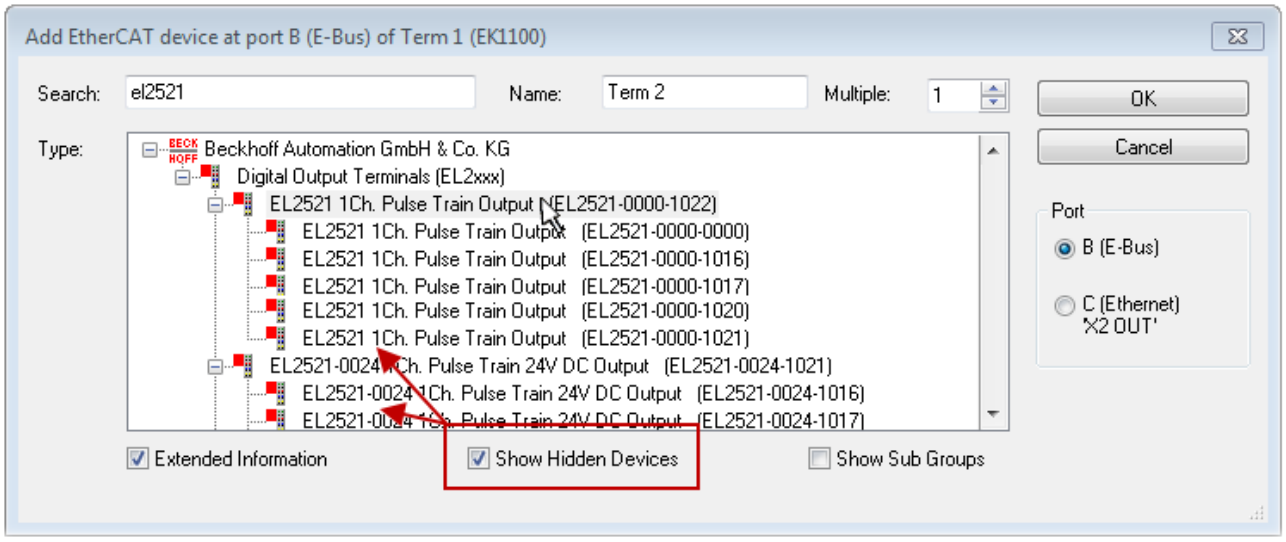


Fig. 97: Display of previous revisions

Device selection based on revision, compatibility

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (-**1019**, -**1020**) can be used in practice.

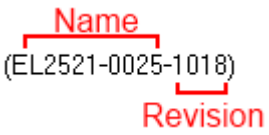


Fig. 98: Name/revision of the terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

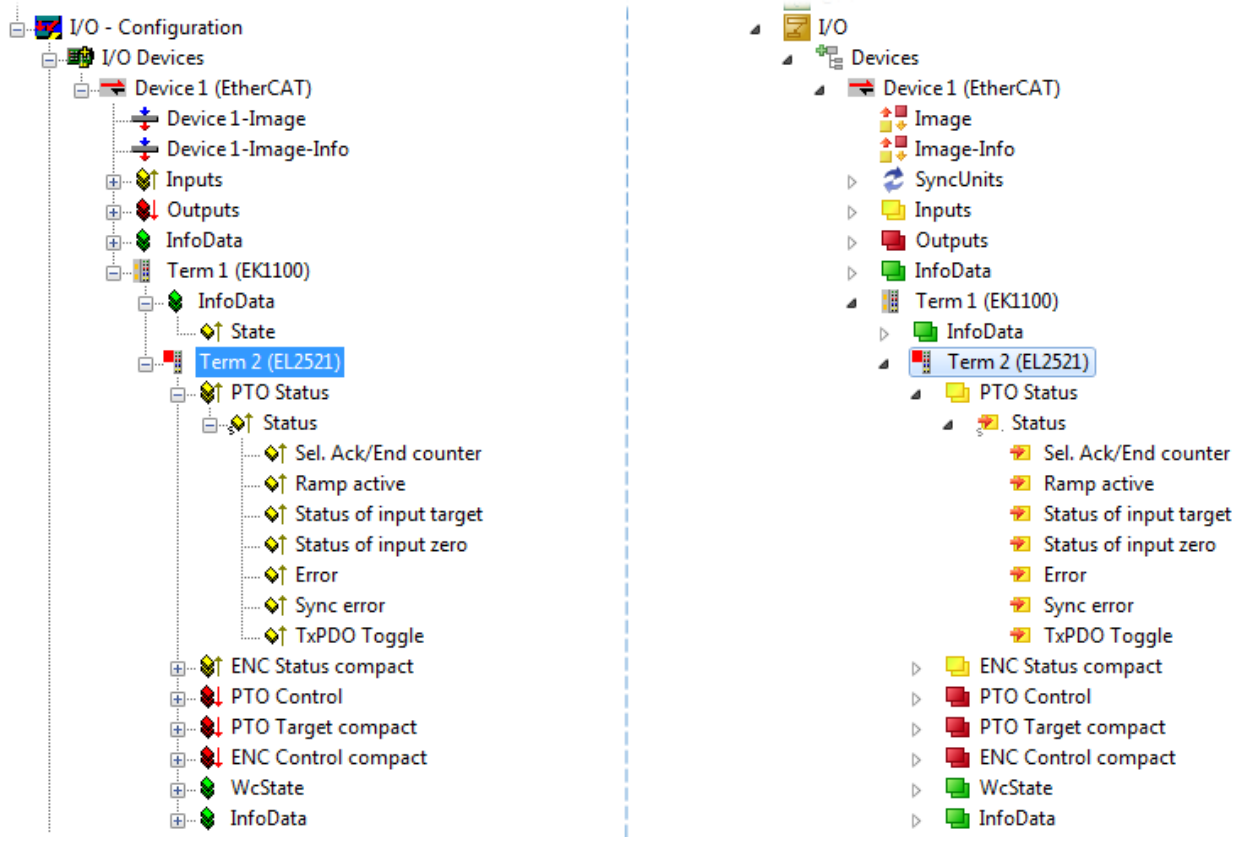


Fig. 99: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

The online device search can be used if the TwinCAT system is in CONFIG mode. This can be indicated by a symbol right below in the information bar:

- on TwinCAT 2 by a blue display “Config Mode” within the System Manager window: Config Mode .
- on TwinCAT 3 within the user interface of the development environment by a symbol .

TwinCAT can be set into this mode:

- TwinCAT 2: by selection of in the Menubar or by “Actions” → “Set/Reset TwinCAT to Config Mode…”
- TwinCAT 3: by selection of in the Menubar or by “TwinCAT” → “Restart TwinCAT (Config Mode)”

Online scanning in Config mode

The online search is not available in RUN mode (production operation). Note the differentiation between TwinCAT programming system and TwinCAT target system.

The TwinCAT 2 icon ( ) or TwinCAT 3 icon ( ) within the Windows-Taskbar always shows the TwinCAT mode of the local IPC. Compared to that, the System Manager window of TwinCAT 2 or the user interface of TwinCAT 3 indicates the state of the target system.

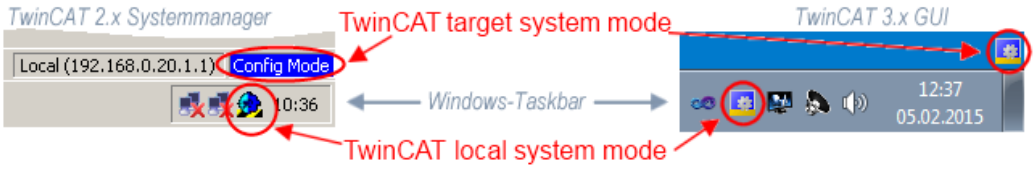


Fig. 100: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

Right-clicking on “I/O Devices” in the configuration tree opens the search dialog.

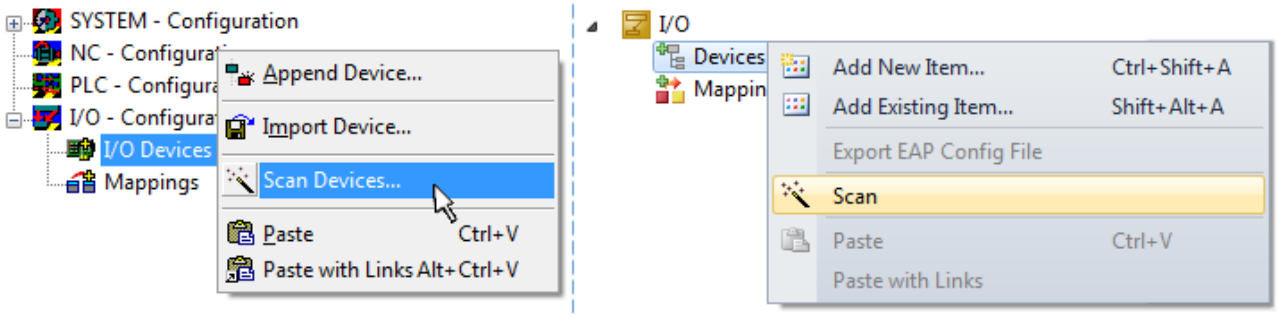


Fig. 101: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

This scan mode attempts to find not only EtherCAT devices (or Ethernet ports that are usable as such), but also NOVRAM, fieldbus cards, SMB etc. However, not all devices can be found automatically.

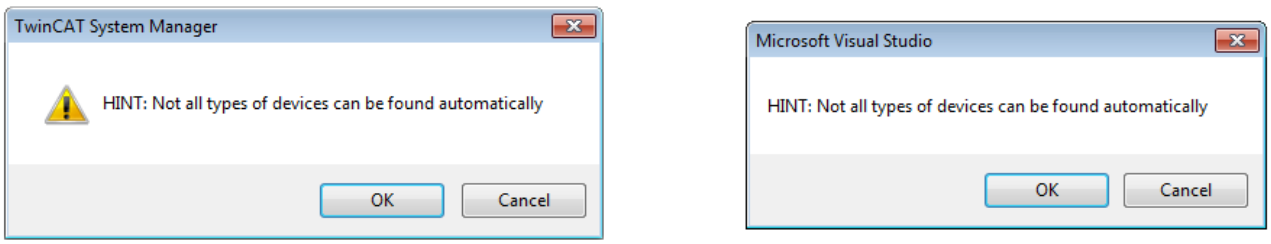


Fig. 102: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

Ethernet ports with installed TwinCAT real-time driver are shown as “RT Ethernet” devices. An EtherCAT frame is sent to these ports for testing purposes. If the scan agent detects from the response that an EtherCAT slave is connected, the port is immediately shown as an “EtherCAT Device” .

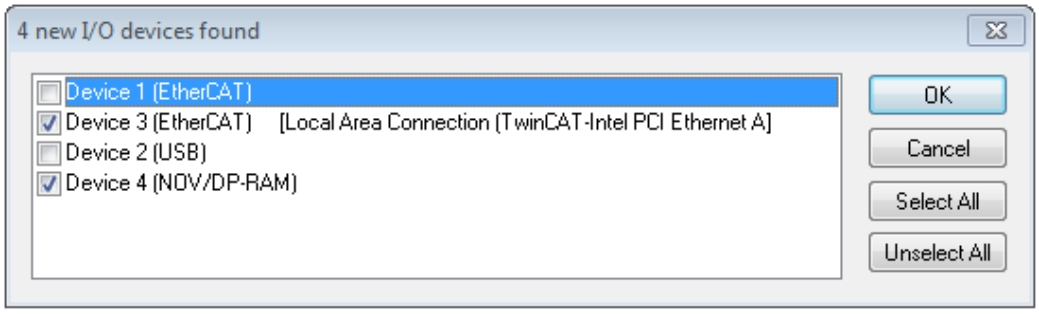


Fig. 103: Detected Ethernet devices

Via respective checkboxes devices can be selected (as illustrated in Fig. “Detected Ethernet devices” e.g. Device 3 and Device 4 were chosen). After confirmation with “OK” a device scan is suggested for all selected devices, see Fig.: “Scan query after automatic creation of an EtherCAT device”.

Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page.

Detecting/Scanning the EtherCAT devices

Online scan functionality

During a scan the master queries the identity information of the EtherCAT slaves from the slave EEPROM. The name and revision are used for determining the type. The respective devices are located in the stored ESI data and integrated in the configuration tree in the default state defined there.

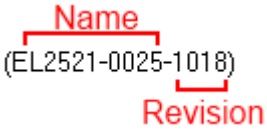


Fig. 104: Example default state

NOTICE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▶ 93] with the defined initial configuration.Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

Company A builds the prototype of a machine B, which is to be produced in series later on. To do this the prototype is built, a scan of the IO devices is performed in TwinCAT and the initial configuration “B.tsm” is created. The EL2521-0025 EtherCAT terminal with the revision 1018 is located somewhere. It is thus built into the TwinCAT configuration in this way:

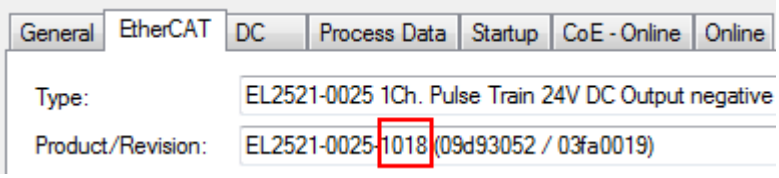


Fig. 105: Installing EthetCAT terminal with revision -1018

Likewise, during the prototype test phase, the functions and properties of this terminal are tested by the programmers/commissioning engineers and used if necessary, i.e. addressed from the PLC “B.pro” or the NC. (the same applies correspondingly to the TwinCAT 3 solution files).

The prototype development is now completed and series production of machine B starts, for which Beckhoff continues to supply the EL2521-0025-0018. If the commissioning engineers of the series machine production department always carry out a scan, a B configuration with the identical contents results again for each machine. Likewise, A might create spare parts stores worldwide for the coming series-produced machines with EL2521-0025-1018 terminals.

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and **a new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of “B.tsm” or even “B.pro” is therefore unnecessary. The series-produced machines can continue to be built with “B.tsm” and “B.pro”; it makes sense to perform a comparative scan [▶ 93] against the initial configuration “B.tsm” in order to check the built machine.

However, if the series machine production department now doesn’t use “B.tsm”, but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:

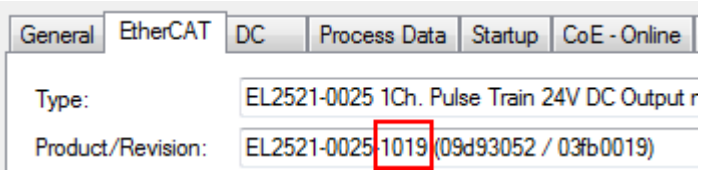


Fig. 106: Detection of EtherCAT terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since virtually a new configuration is created. According to the compatibility rule, however, this means that no EL2521-0025-**1018** should be built into this machine as a spare part (even if this nevertheless works in the vast majority of cases).

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration “B2.tsm” created in this way. Þ if series machine production is established, the scan should only be performed for informative purposes for comparison with a defined initial configuration. Changes are to be made with care!

If an EtherCAT device was created in the configuration (manually or through a scan), the I/O field can be scanned for devices/slaves.

Fig. 107: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

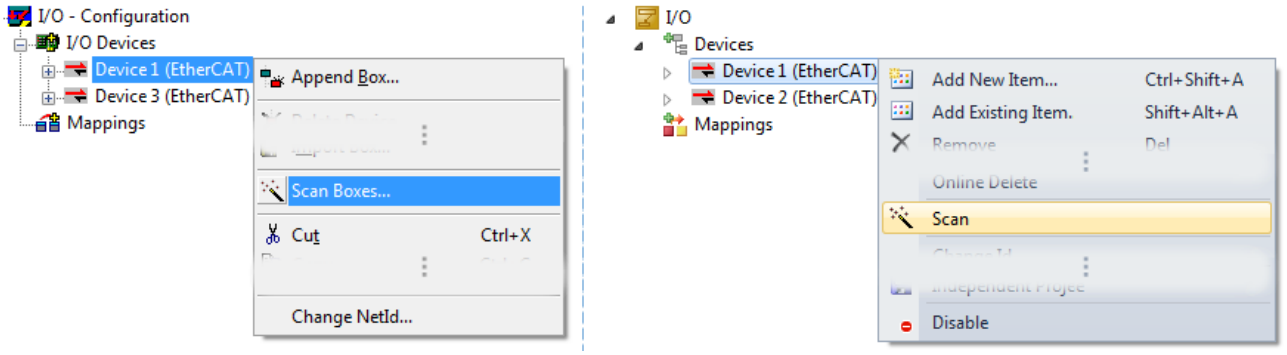


Fig. 108: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

In the System Manager (TwinCAT 2) or the User Interface (TwinCAT 3) the scan process can be monitored via the progress bar at the bottom in the status bar.

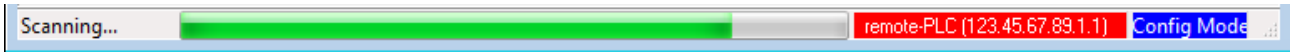


Fig. 109: *Scan progressexemplary by TwinCAT 2*

The configuration is established and can then be switched to online state (OPERATIONAL).

Fig. 110: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

In Config/FreeRun mode the System Manager display alternates between blue and red, and the EtherCAT device continues to operate with the idling cycle time of 4 ms (default setting), even without active task (NC, PLC).

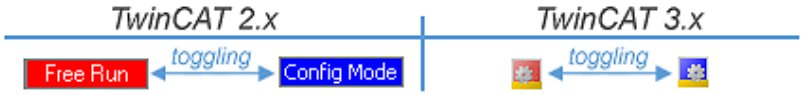


Fig. 111: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

Fig. 112: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

The EtherCAT system should then be in a functional cyclic state, as shown in Fig. *Online display example*.

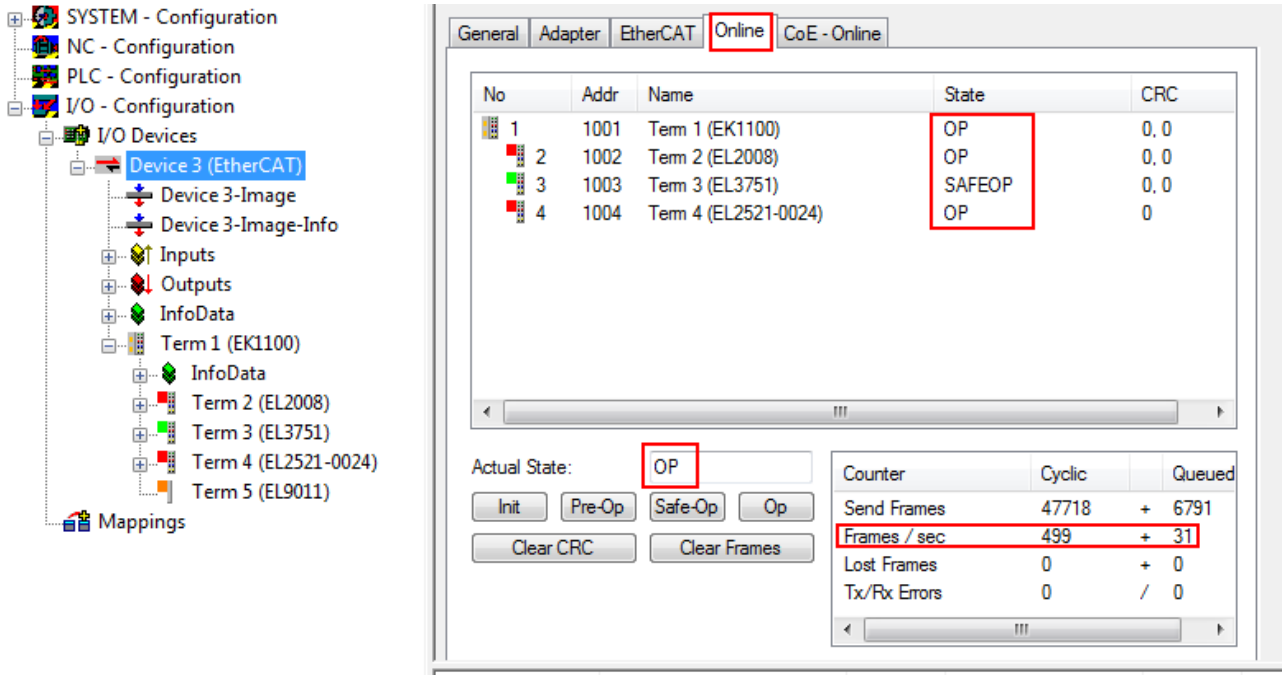


Fig. 113: Online display example

Please note:

- all slaves should be in OP state
- the EtherCAT master should be in “Actual State” OP
- “frames/sec” should match the cycle time taking into account the sent number of frames
- no excessive “LostFrames” or CRC errors should occur

The configuration is now complete. It can be modified as described under manual procedure.

Troubleshooting

Various effects may occur during scanning.

- An **unknown device** is detected, i.e. an EtherCAT slave for which no ESI XML description is available. In this case the System Manager offers to read any ESI that may be stored in the device. This case is described in the chapter “Notes regarding ESI device description”.
- **Device are not detected properly**
  Possible reasons include:
  - faulty data links, resulting in data loss during the scan
  - slave has invalid device description

  The connections and devices should be checked in a targeted manner, e.g. via the emergency scan. Then re-run the scan.

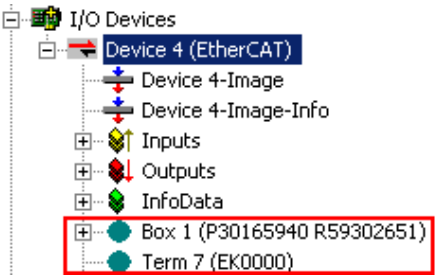


Fig. 114: Faulty identification

In the System Manager such devices may be set up as EK0000 or unknown devices. Operation is not possible or meaningful.

Scan over existing Configuration

NOTICE

Change of the configuration after comparison

With this scan (TwinCAT 2.11 or 3.1) only the device properties vendor (manufacturer), device name and revision are compared at present! A “ChangeTo” or “Copy” should only be carried out with care, taking into consideration the Beckhoff IO compatibility rule (see above). The device configuration is then replaced by the revision found; this can affect the supported process data and functions.

If a scan is initiated for an existing configuration, the actual I/O environment may match the configuration exactly or it may differ. This enables the configuration to be compared.

Fig. 115: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

If differences are detected, they are shown in the correction dialog, so that the user can modify the configuration as required.

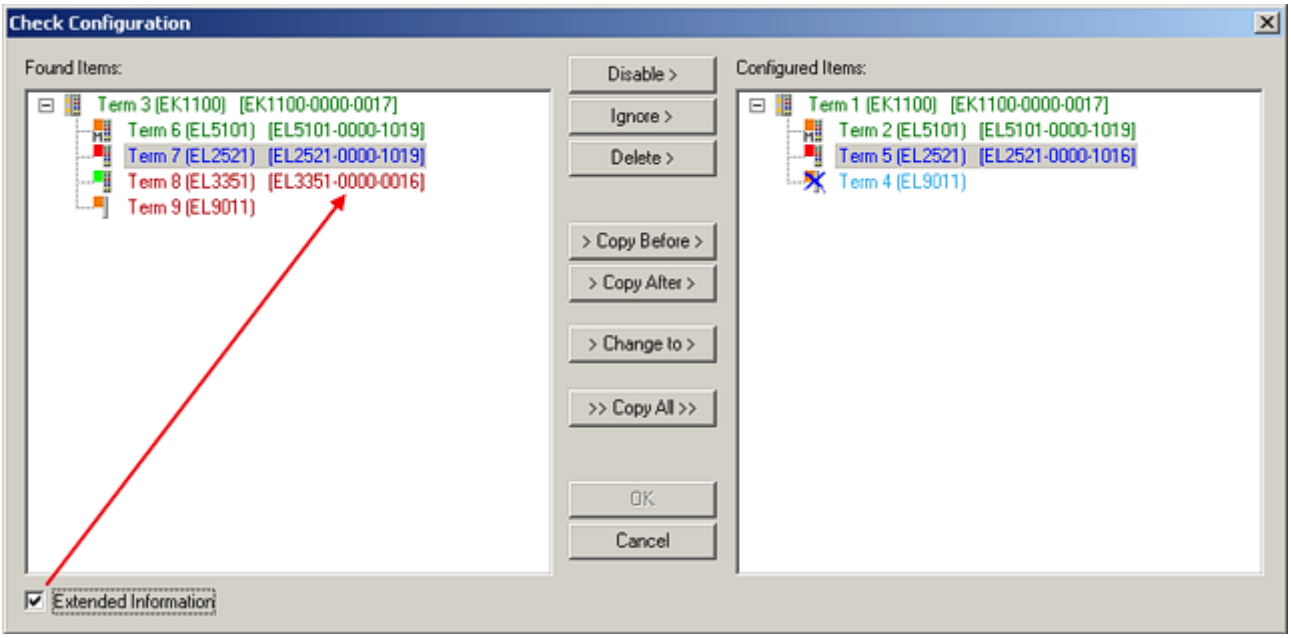


Fig. 116: Correction dialog

It is advisable to tick the “Extended Information” check box to reveal differences in the revision.

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored (“Ignore” button)
red	• This EtherCAT slave is not present on the other side. • It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

Device selection based on revision, compatibility

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (-**1019**, -**1020**) can be used in practice.

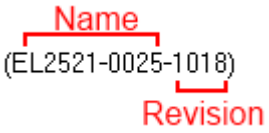


Fig. 117: Name/revision of the terminal

If current ESI descriptions are available in the TwinCAT system, the last revision offered in the selection dialog matches the Beckhoff state of production. It is recommended to use the last device revision when creating a new configuration, if current Beckhoff devices are used in the real application. Older revisions should only be used if older devices from stock are to be used in the application.

In this case the process image of the device is shown in the configuration tree and can be parameterized as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

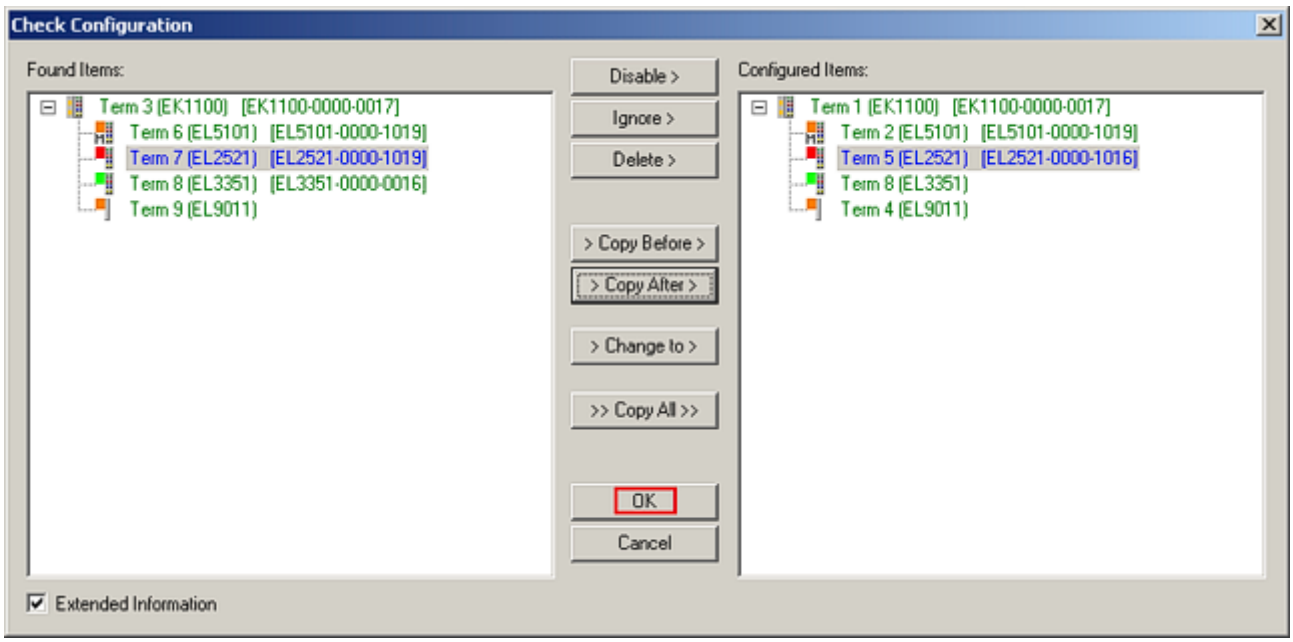


Fig. 118: Correction dialog with modifications

Once all modifications have been saved or accepted, click “OK” to transfer them to the real *.tsm configuration.

Change to Compatible Type

TwinCAT offers a function *Change to Compatible Type…* for the exchange of a device whilst retaining the links in the task.

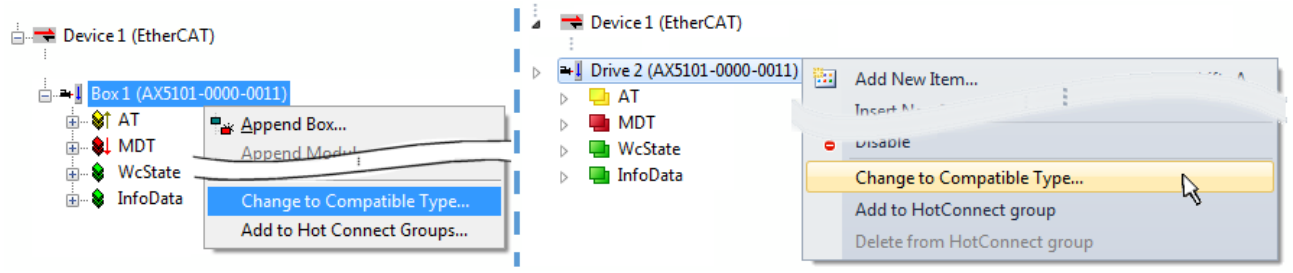


Fig. 119: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

The TwinCAT System Manager offers a function for the exchange of a device: Change to Alternative Type

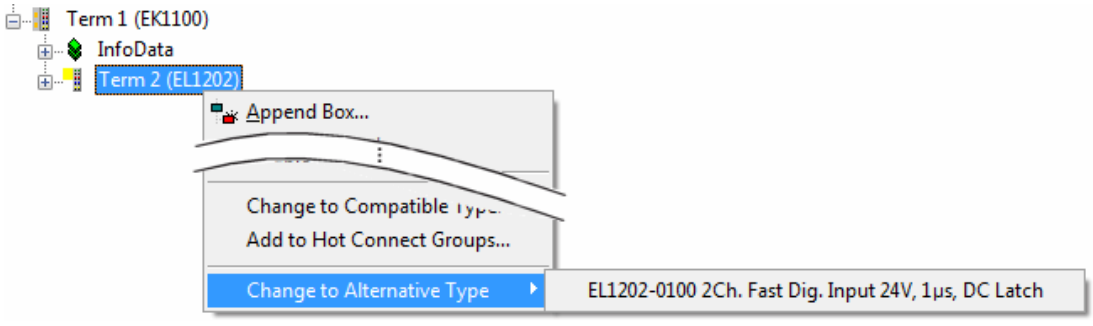


Fig. 120: TwinCAT 2 Dialog Change to Alternative Type

If called, the System Manager searches in the procured device ESI (in this example: EL1202-0000) for details of compatible devices contained there. The configuration is changed and the ESI-EEPROM is overwritten at the same time – therefore this process is possible only in the online state (ConfigMode).

5.2.7 EtherCAT subscriber configuration

In the left-hand window of the TwinCAT 2 System Manager or the Solution Explorer of the TwinCAT 3 Development Environment respectively, click on the element of the terminal within the tree you wish to configure (in the example: EL3751 Terminal 3).

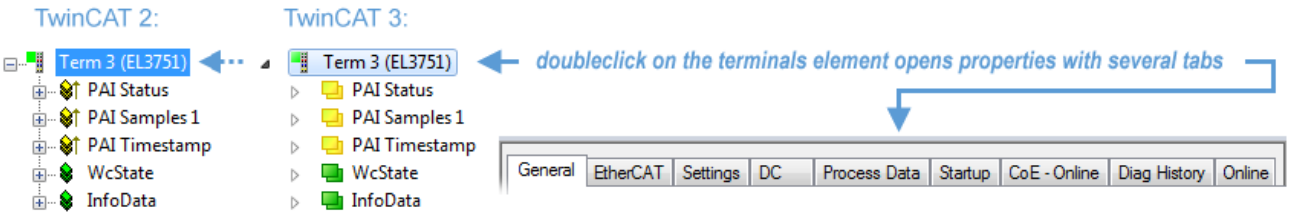


Fig. 121: Branch element as terminal EL3751

In the right-hand window of the TwinCAT System Manager (TwinCAT 2) or the Development Environment (TwinCAT 3), various tabs are now available for configuring the terminal. And yet the dimension of complexity of a subscriber determines which tabs are provided. Thus as illustrated in the example above the terminal EL3751 provides many setup options and also a respective number of tabs are available. On the contrary by the terminal EL1004 for example the tabs “General”, “EtherCAT”, “Process Data” and “Online“ are available only. Several terminals, as for instance the EL6695 provide special functions by a tab with its own terminal name, so “EL6695” in this case. A specific tab “Settings” by terminals with a wide range of setup options will be provided also (e.g. EL3751).

“General” tab

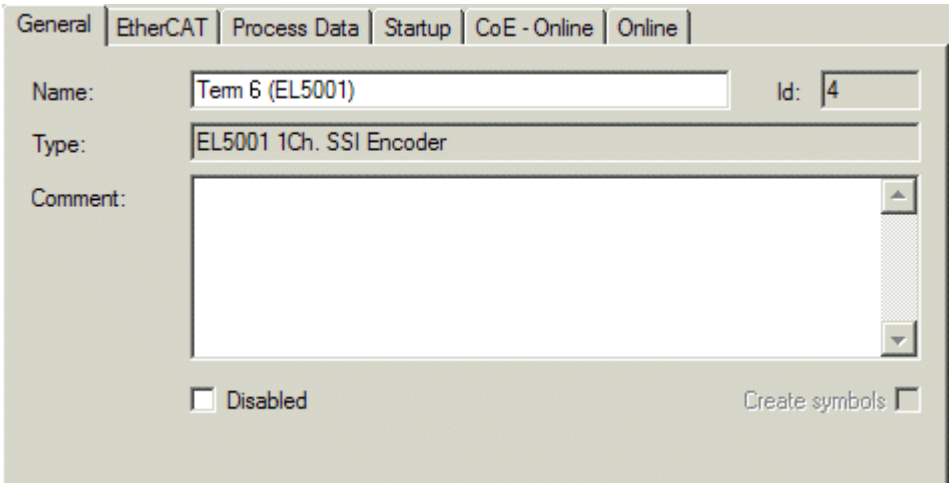


Fig. 122: “General” tab

Name	Name of the EtherCAT device
Id	Number of the EtherCAT device
Type	EtherCAT device type
Comment	Here you can add a comment (e.g. regarding the system).
Disabled	Here you can deactivate the EtherCAT device.
Create symbols	Access to this EtherCAT slave via ADS is only available if this control box is activated.

“EtherCAT” tab

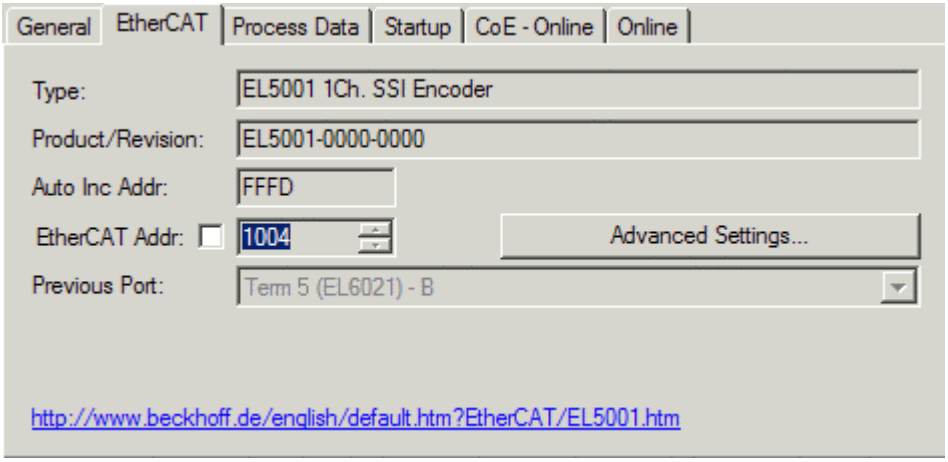


Fig. 123: “EtherCAT” tab

Type	EtherCAT device type
Product/Revision	Product and revision number of the EtherCAT device
Auto Inc Addr.	Auto increment address of the EtherCAT device. The auto increment address can be used for addressing each EtherCAT device in the communication ring through its physical position. Auto increment addressing is used during the start-up phase when the EtherCAT master allocates addresses to the EtherCAT devices. With auto increment addressing the first EtherCAT slave in the ring has the address 0000_{hex}. For each further slave the address is decremented by 1 ($FFFF_{hex}$, $FFFE_{hex}$ etc.).
EtherCAT Addr.	Fixed address of an EtherCAT slave. This address is allocated by the EtherCAT master during the start-up phase. Tick the control box to the left of the input field in order to modify the default value.
Previous Port	Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this device is to be connected can be selected.
Advanced Settings	This button opens the dialogs for advanced settings.

The link at the bottom of the tab points to the product page for this EtherCAT device on the web.

“Process Data” tab

Indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (**P**rocess **D**ata **O**bjects, PDOs). The user can select a PDO via PDO assignment and modify the content of the individual PDO via this dialog, if the EtherCAT slave supports this function.

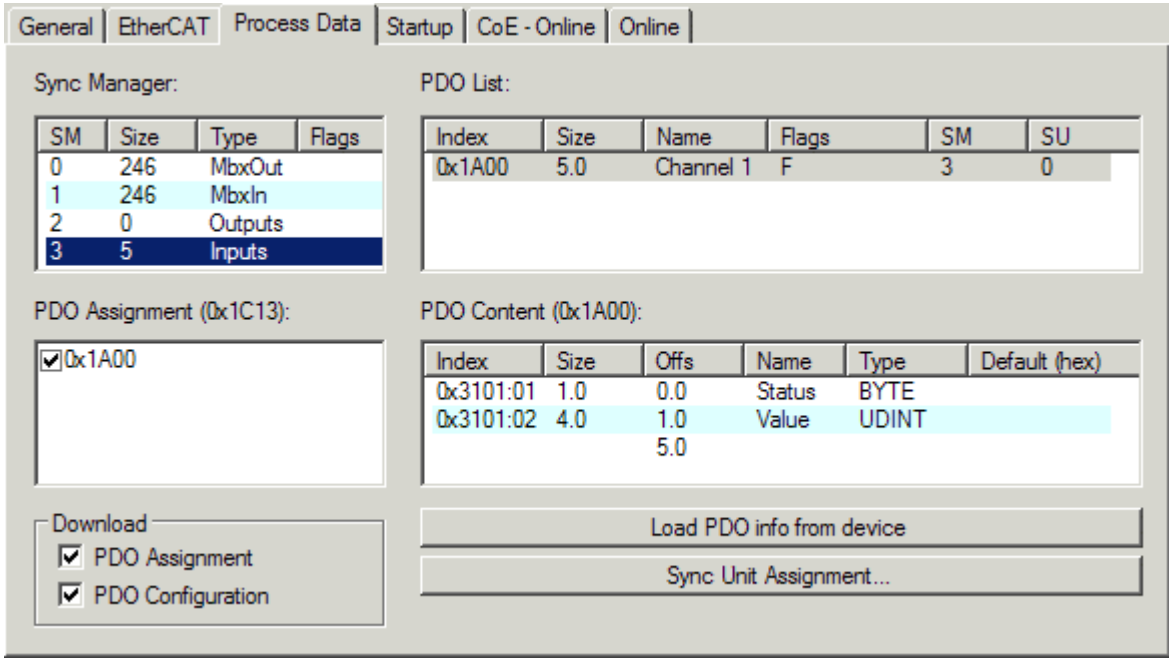


Fig. 124: “Process Data” tab

The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which process data (size in bits/bytes, source location, transmission type) it wants to transfer to or from this slave. Incorrect configuration can prevent successful start-up of the slave.

For Beckhoff EtherCAT EL, ES, EM, EJ and EP slaves the following applies in general:

- The input/output process data supported by the device are defined by the manufacturer in the ESI/XML description. The TwinCAT EtherCAT Master uses the ESI description to configure the slave correctly.
- The process data can be modified in the System Manager. See the device documentation.
  Examples of modifications include: mask out a channel, displaying additional cyclic information, 16-bit display instead of 8-bit data size, etc.
- In so-called “intelligent” EtherCAT devices the process data information is also stored in the CoE directory. Any changes in the CoE directory that lead to different PDO settings prevent successful startup of the slave. It is not advisable to deviate from the designated process data, because the device firmware (if available) is adapted to these PDO combinations.

If the device documentation allows modification of process data, proceed as follows (see Figure *Configuring the process data*).

- A: select the device to configure
- B: in the “Process Data” tab select Input or Output under SyncManager (C)
- D: the PDOs can be selected or deselected
- H: the new process data are visible as linkable variables in the System Manager
  The new process data are active once the configuration has been activated and TwinCAT has been restarted (or the EtherCAT master has been restarted)
- E: if a slave supports this, Input and Output PDO can be modified simultaneously by selecting a so-called PDO record (“predefined PDO settings”).

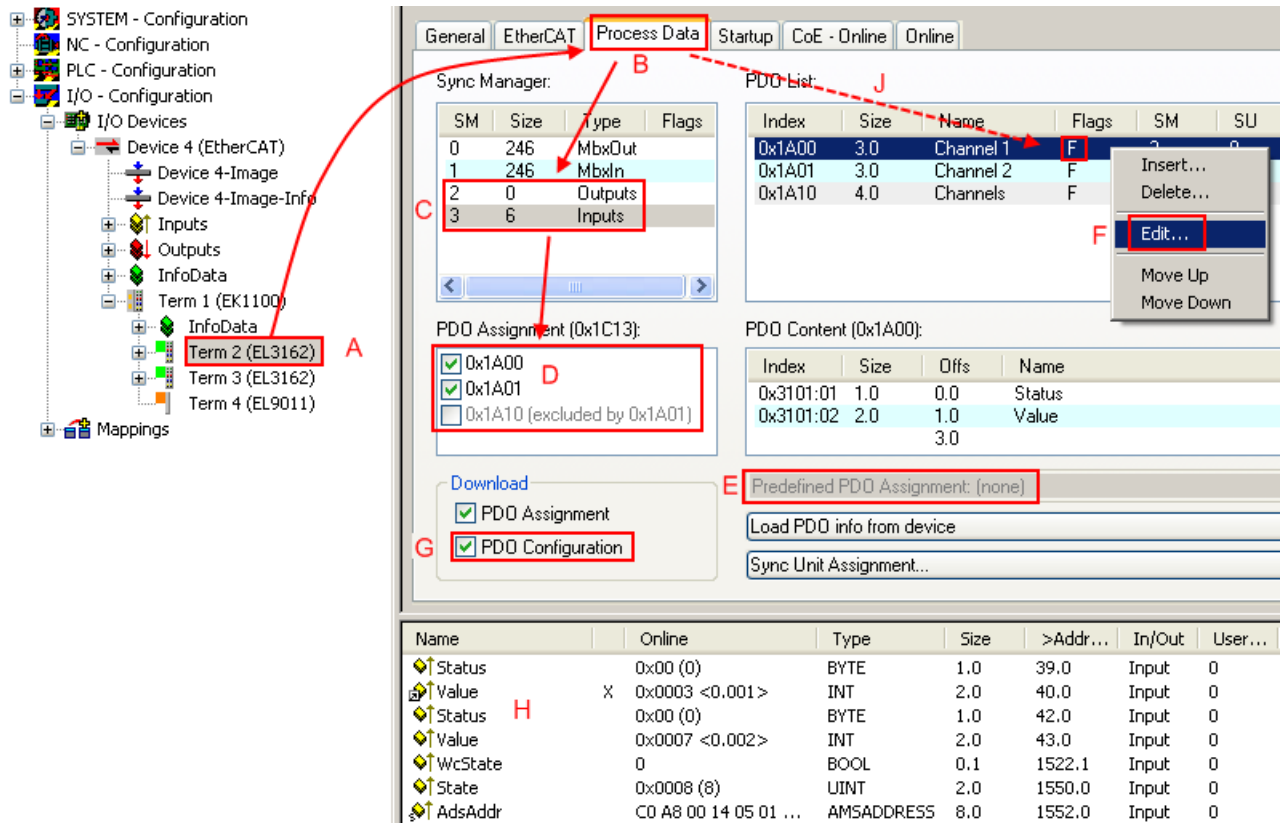


Fig. 125: Configuring the process data

Manual modification of the process data

According to the ESI description, a PDO can be identified as “fixed” with the flag “F” in the PDO overview (Fig. *Configuring the process data*, J). The configuration of such PDOs cannot be changed, even if TwinCAT offers the associated dialog (“Edit”). In particular, CoE content cannot be displayed as cyclic process data. This generally also applies in cases where a device supports download of the PDO configuration, “G”. In case of incorrect configuration the EtherCAT slave usually refuses to start and change to OP state. The System Manager displays an “invalid SM cfg” logger message: This error message (“invalid SM IN cfg” or “invalid SM OUT cfg”) also indicates the reason for the failed start.

A detailed description [▶ 104] can be found at the end of this section.

“Startup” tab

The *Startup* tab is displayed if the EtherCAT slave has a mailbox and supports the *CANopen over EtherCAT* (CoE) or *Servo drive over EtherCAT* protocol. This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.

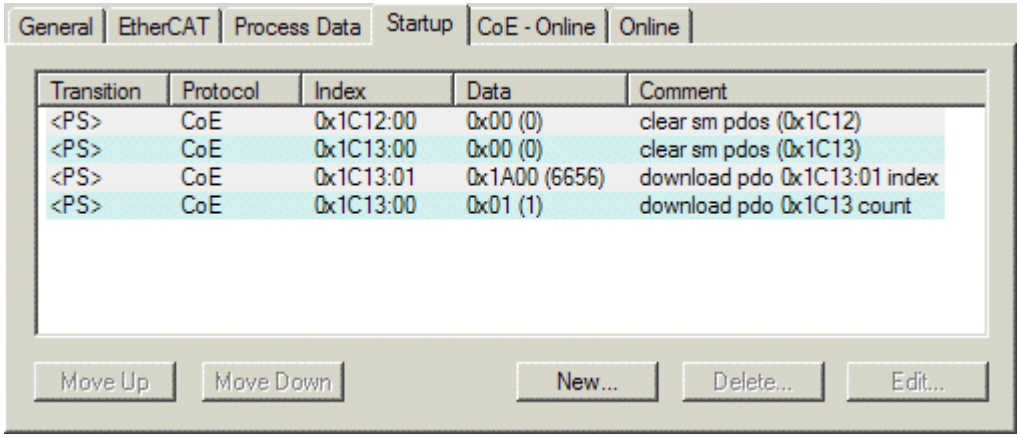


Fig. 126: “Startup” tab

Column	Description
Transition	Transition to which the request is sent. This can either be • the transition from pre-operational to safe-operational (PS), or • the transition from safe-operational to operational (SO). If the transition is enclosed in “<>” (e.g. <PS>), the mailbox request is fixed and cannot be modified or deleted by the user.
Protocol	Type of mailbox protocol
Index	Index of the object
Data	Date on which this object is to be downloaded.
Comment	Description of the request to be sent to the mailbox

Move Up	This button moves the selected request up by one position in the list.
Move Down	This button moves the selected request down by one position in the list.
New	This button adds a new mailbox download request to be sent during startup.
Delete	This button deletes the selected entry.
Edit	This button edits an existing request.

“CoE - Online” tab

The additional *CoE - Online* tab is displayed if the EtherCAT slave supports the *CANopen over EtherCAT* (CoE) protocol. This dialog lists the content of the object list of the slave (SDO upload) and enables the user to modify the content of an object from this list. Details for the objects of the individual EtherCAT devices can be found in the device-specific object descriptions.

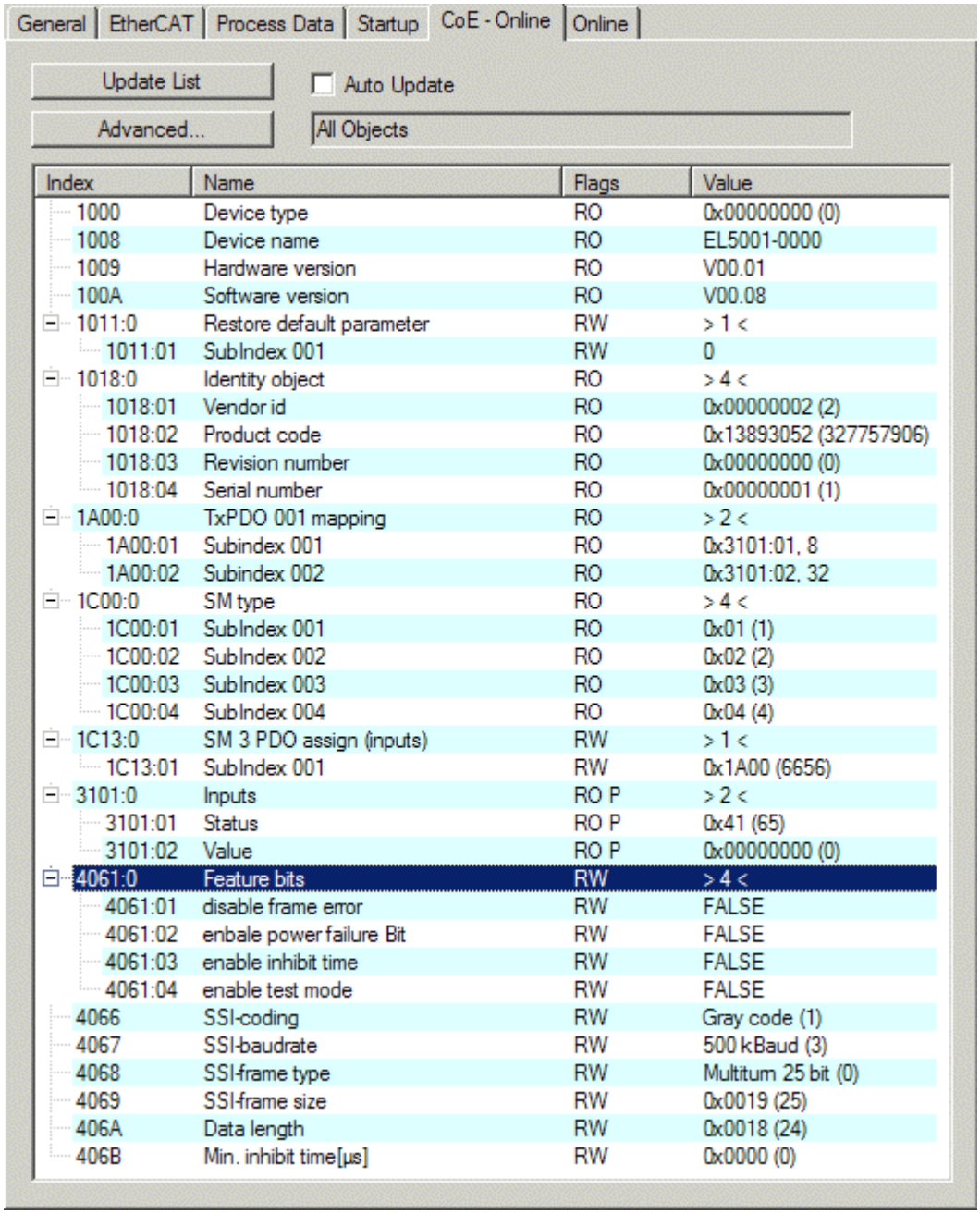


Fig. 127: “CoE - Online” tab

Object list display

Column	Description	
Index	Index and sub-index of the object	
Name	Name of the object	
Flags	RW	The object can be read, and data can be written to the object (read/write)
	RO	The object can be read, but no data can be written to the object (read only)
	P	An additional P identifies the object as a process data object.
Value	Value of the object	

Update List The *Update list* button updates all objects in the displayed list

Auto Update If this check box is selected, the content of the objects is updated automatically.

Advanced The *Advanced* button opens the *Advanced Settings* dialog. Here you can specify which objects are displayed in the list.

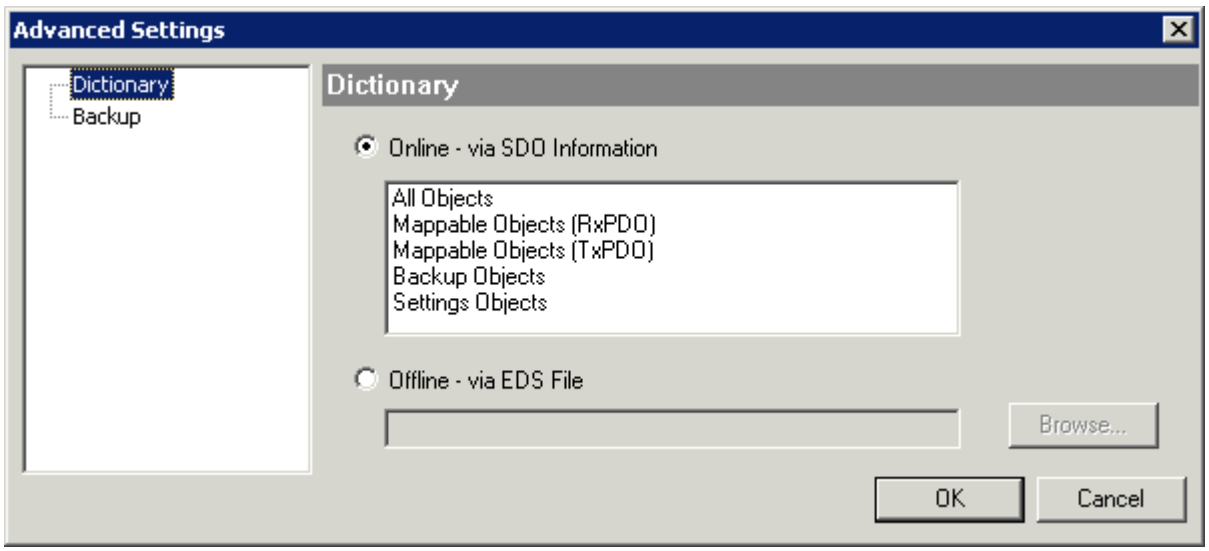


Fig. 128: Dialog “Advanced settings”

Online - via SDO Information	If this option button is selected, the list of the objects included in the object list of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.
Offline - via EDS File	If this option button is selected, the list of the objects included in the object list is read from an EDS file provided by the user.

“Online” tab

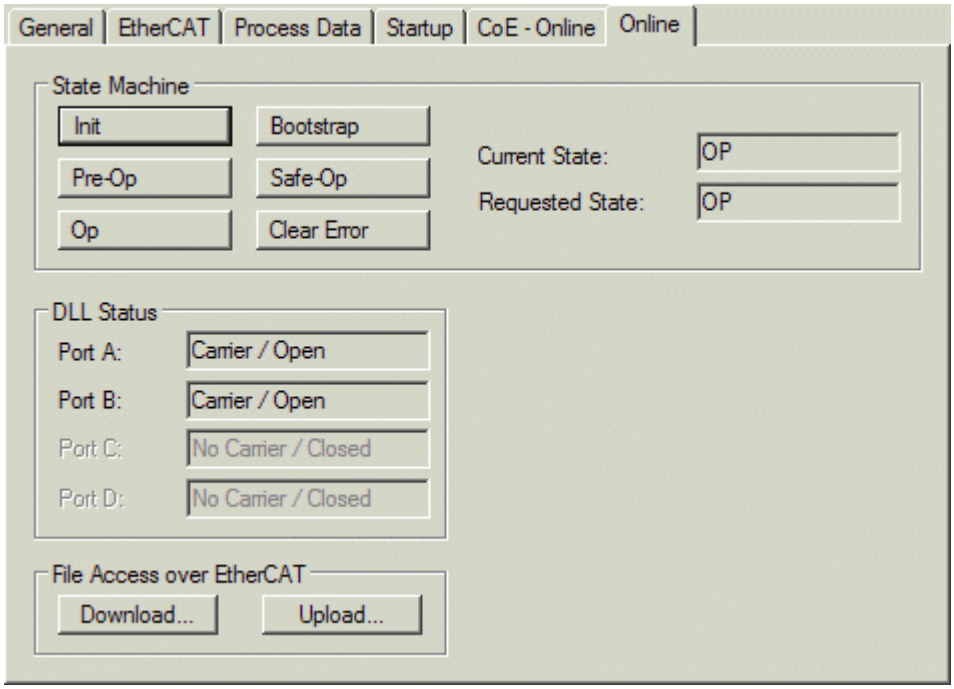


Fig. 129: “Online” tab

State Machine

Init	This button attempts to set the EtherCAT device to the *Init* state.
Pre-Op	This button attempts to set the EtherCAT device to the *pre-operational* state.
Op	This button attempts to set the EtherCAT device to the *operational* state.
Bootstrap	This button attempts to set the EtherCAT device to the *Bootstrap* state.
Safe-Op	This button attempts to set the EtherCAT device to the *safe-operational* state.
Clear Error	This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag. Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. When the *Clear Error* button is pressed the error flag is cleared, and the current state is displayed as PREOP again.
Current State	Indicates the current state of the EtherCAT device.
Requested State	Indicates the state requested for the EtherCAT device.

DLL Status

Indicates the DLL status (data link layer status) of the individual ports of the EtherCAT slave. The DLL status can have four different states:

Status	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal is available at the port, but the port is closed.

File Access over EtherCAT

Download	With this button a file can be written to the EtherCAT device.
Upload	With this button a file can be read from the EtherCAT device.

“DC” tab (Distributed Clocks)

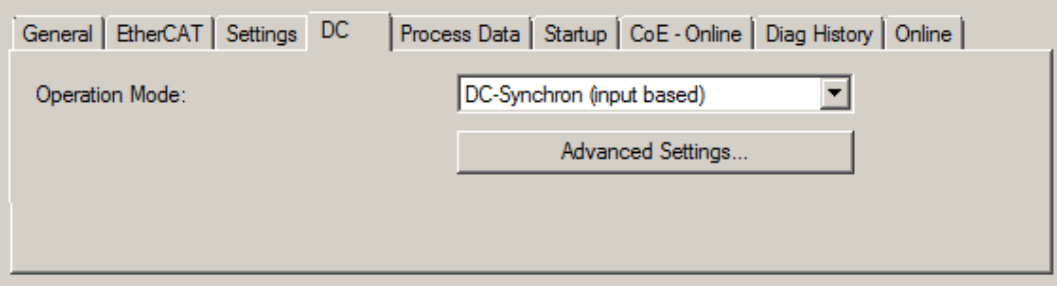


Fig. 130: “DC” tab (Distributed Clocks)

Operation Mode	Options (optional): • FreeRun • SM-Synchron • DC-Synchron (Input based) • DC-Synchron
Advanced Settings…	Advanced settings for readjustment of the real time determinant TwinCAT-clock

Detailed information to Distributed Clocks is specified on http://infosys.beckhoff.com:

Fieldbus Components → EtherCAT Terminals → EtherCAT System documentation → EtherCAT basics → Distributed Clocks

5.2.7.1 Detailed description of Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).
If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).
SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

If an input is selected, the corresponding PDO assignment is displayed in the *PDO Assignment* list below.

PDO Assignment

PDO assignment of the selected Sync Manager. All PDOs defined for this Sync Manager type are listed here:

- If the output Sync Manager (outputs) is selected in the Sync Manager list, all RxPDOs are displayed.
- If the input Sync Manager (inputs) is selected in the Sync Manager list, all TxPDOs are displayed.

The selected entries are the PDOs involved in the process data transfer. In the tree diagram of the System Manager these PDOs are displayed as variables of the EtherCAT device. The name of the variable is identical to the *Name* parameter of the PDO, as displayed in the PDO list. If an entry in the PDO assignment list is deactivated (not selected and greyed out), this indicates that the input is excluded from the PDO assignment. In order to be able to select a greyed out PDO, the currently selected PDO has to be deselected first.

Activation of PDO assignment

✓ If you have changed the PDO assignment, in order to activate the new PDO assignment,

a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see Online tab [▶ 102]),

b) and the System Manager has to reload the EtherCAT slaves

( button for TwinCAT 2 or  button for TwinCAT 3)

PDO list

List of all PDOs supported by this EtherCAT device. The content of the selected PDOs is displayed in the *PDO Content* list. The PDO configuration can be modified by double-clicking on an entry.

Column	Description	
Index	PDO index.	
Size	Size of the PDO in bytes.	
Name	Name of the PDO. If this PDO is assigned to a Sync Manager, it appears as a variable of the slave with this parameter as the name.	
Flags	F	Fixed content: The content of this PDO is fixed and cannot be changed by the System Manager.
	M	Mandatory PDO. This PDO is mandatory and must therefore be assigned to a Sync Manager! Consequently, this PDO cannot be deleted from the *PDO Assignment* list
SM	Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic.	
SU	Sync unit to which this PDO is assigned.	

PDO Content

Indicates the content of the PDO. If flag F (fixed content) of the PDO is not set the content can be modified.

Download

If the device is intelligent and has a mailbox, the configuration of the PDO and the PDO assignments can be downloaded to the device. This is an optional feature that is not supported by all EtherCAT slaves.

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the Startup [▶ 99] tab.

PDO Configuration

If this check box is selected, the configuration of the respective PDOs (as shown in the PDO list and the PDO Content display) is downloaded to the EtherCAT slave.

5.3 General Notes - EtherCAT Slave Application

This summary briefly deals with a number of aspects of EtherCAT slave operation under TwinCAT. More detailed information on this may be found in the corresponding sections of, for instance, the EtherCAT System Documentation.

Diagnosis in real-time: WorkingCounter, EtherCAT State and Status

Generally speaking an EtherCAT slave provides a variety of diagnostic information that can be used by the controlling task.

This diagnostic information relates to differing levels of communication. It therefore has a variety of sources, and is also updated at various times.

Any application that relies on I/O data from a fieldbus being correct and up to date must make diagnostic access to the corresponding underlying layers.

EtherCAT and the TwinCAT System Manager offer comprehensive diagnostic elements of this kind. Those diagnostic elements that are helpful for the controlling task for diagnosis that is accurate for the current cycle when in operation (not during commissioning) are discussed below.

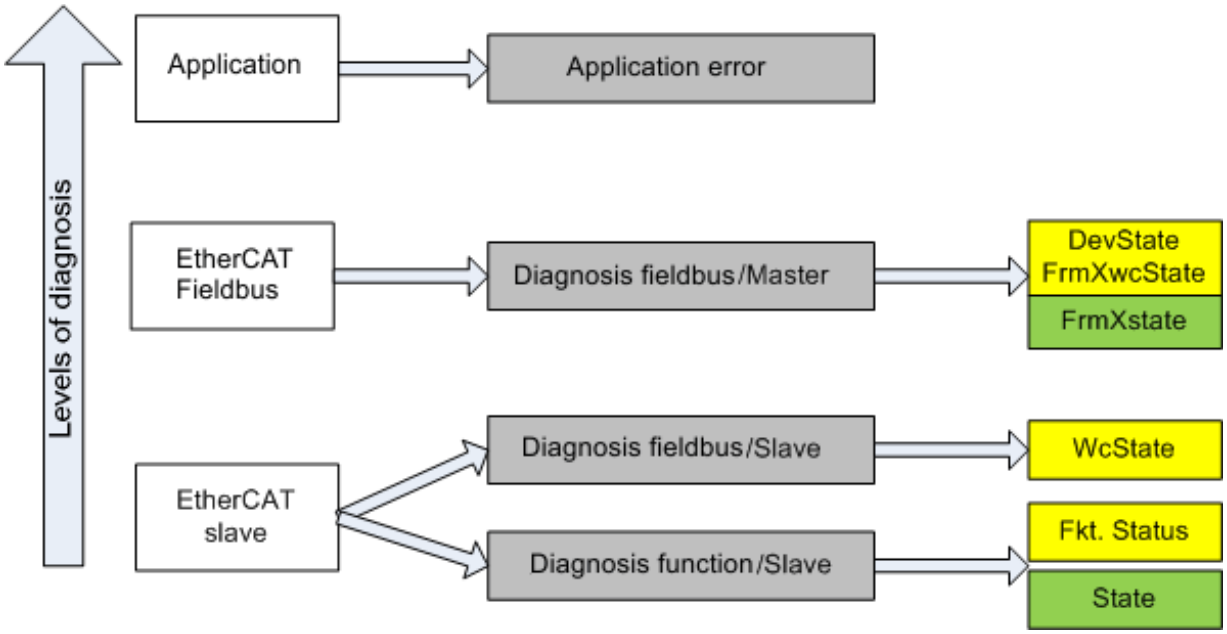


Fig. 131: Selection of the diagnostic information of an EtherCAT slave

In general, an EtherCAT slave offers

- communication diagnosis typical for a slave (diagnosis of successful participation in the process data exchange, and correct operation mode)
  This diagnosis is the same for all slaves.

  and
- channel-typical function diagnosis (device-dependent),
  see corresponding device documentation

The coloring in Fig. *Selection of diagnostic information for an EtherCAT slave* also corresponds to the variable colors in the System Manager, see Fig. *Basic EtherCAT slave diagnosis in the PLC*.

Color	Meaning
yellow	Input variables from the slave to the EtherCAT master, updated in every cycle
red	Output variables from the slave to the EtherCAT master, updated in every cycle
green	Information variables for the EtherCAT master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore useful to read such variables through ADS.

Fig. *Basic EtherCAT slave diagnosis in the PLC* shows an example implementation of basic EtherCAT slave diagnosis. A Beckhoff EL3102 (2-channel analog input terminal) is used here, as it offers both the communication diagnosis typical of a slave and the functional diagnosis that is specific to a channel. Structures are created as input variables in the PLC, each corresponding to the process image.

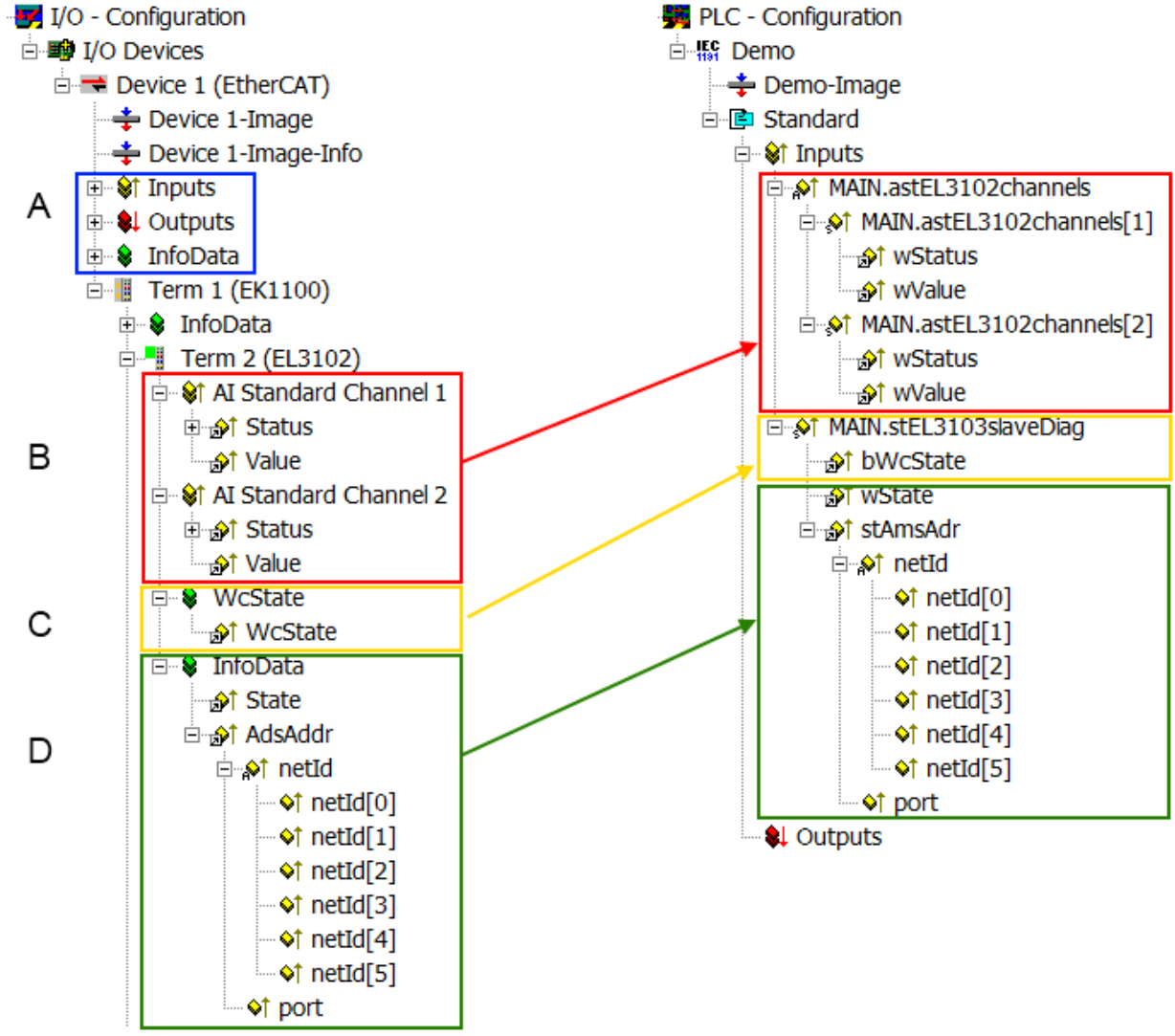


Fig. 132: Basic EtherCAT slave diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: • CoE in the Master for communication with/through the Slaves • Functions from *TcEtherCAT.lib* • Perform an OnlineScan
B	In the example chosen (EL3102) the EL3102 comprises two analogue input channels that transmit a single function status for the most recent cycle.	Status • the bit significations may be found in the device documentation • other devices may supply more information, or none that is typical of a slave	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the function status must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
C	For every EtherCAT Slave that has cyclic process data, the Master displays, using what is known as a WorkingCounter, whether the slave is participating successfully and without error in the cyclic exchange of process data. This important, elementary information is therefore provided for the most recent cycle in the System Manager 1. at the EtherCAT Slave, and, with identical contents 2. as a collective variable at the EtherCAT Master (see Point A) for linking.	WcState (Working Counter) 0: valid real-time communication in the last cycle 1: invalid real-time communication This may possibly have effects on the process data of other Slaves that are located in the same SyncUnit	In order for the higher-level PLC task (or corresponding control applications) to be able to rely on correct data, the communication status of the EtherCAT Slave must be evaluated there. Such information is therefore provided with the process data for the most recent cycle.
D	Diagnostic information of the EtherCAT Master which, while it is represented at the slave for linking, is actually determined by the Master for the Slave concerned and represented there. This information cannot be characterized as real-time, because it • is only rarely/never changed, except when the system starts up • is itself determined acyclically (e.g. EtherCAT Status)	State current Status (INIT..OP) of the Slave. The Slave must be in OP (=8) when operating normally. *AdsAddr* The ADS address is useful for communicating from the PLC/task via ADS with the EtherCAT Slave, e.g. for reading/writing to the CoE. The AMS-NetID of a slave corresponds to the AMS-NetID of the EtherCAT Master; communication with the individual Slave is possible via the *port* (= EtherCAT address).	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore possible to read such variables through ADS.

NOTICE

Diagnostic information

It is strongly recommended that the diagnostic information made available is evaluated so that the application can react accordingly.

CoE parameter directory

The CoE parameter directory (CanOpen-over-EtherCAT) is used to manage the set values for the slave concerned. Changes may, in some circumstances, have to be made here when commissioning a relatively comprehensive EtherCAT slave. It can be accessed via the TwinCAT System Manager, see Fig. *EL3102, CoE directory*:

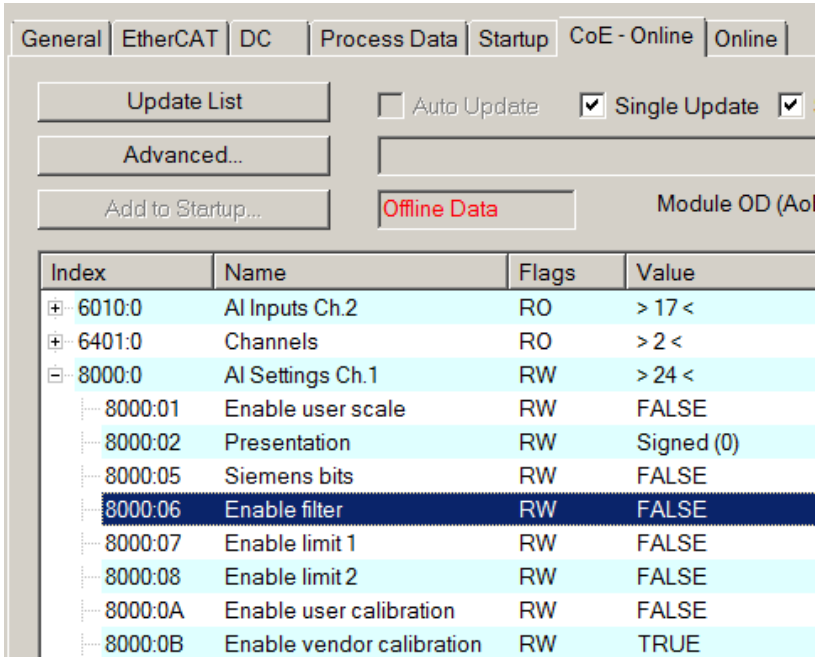


Fig. 133: EL3102, CoE directory

EtherCAT system documentation

The comprehensive description in the EtherCAT system documentation (EtherCAT Basics --> CoE Interface) must be observed!

A few brief extracts:

- Whether changes in the online directory are saved locally in the slave depends on the device. EL terminals (except the EL66xx) are able to save in this way.
- The user must manage the changes to the StartUp list.

Commissioning aid in the TwinCAT System Manager

Commissioning interfaces are being introduced as part of an ongoing process for EL/EP EtherCAT devices. These are available in the TwinCAT System Manager from TwinCAT 2.11R2 and above. They are integrated into the System Manager through appropriately extended ESI configuration files.

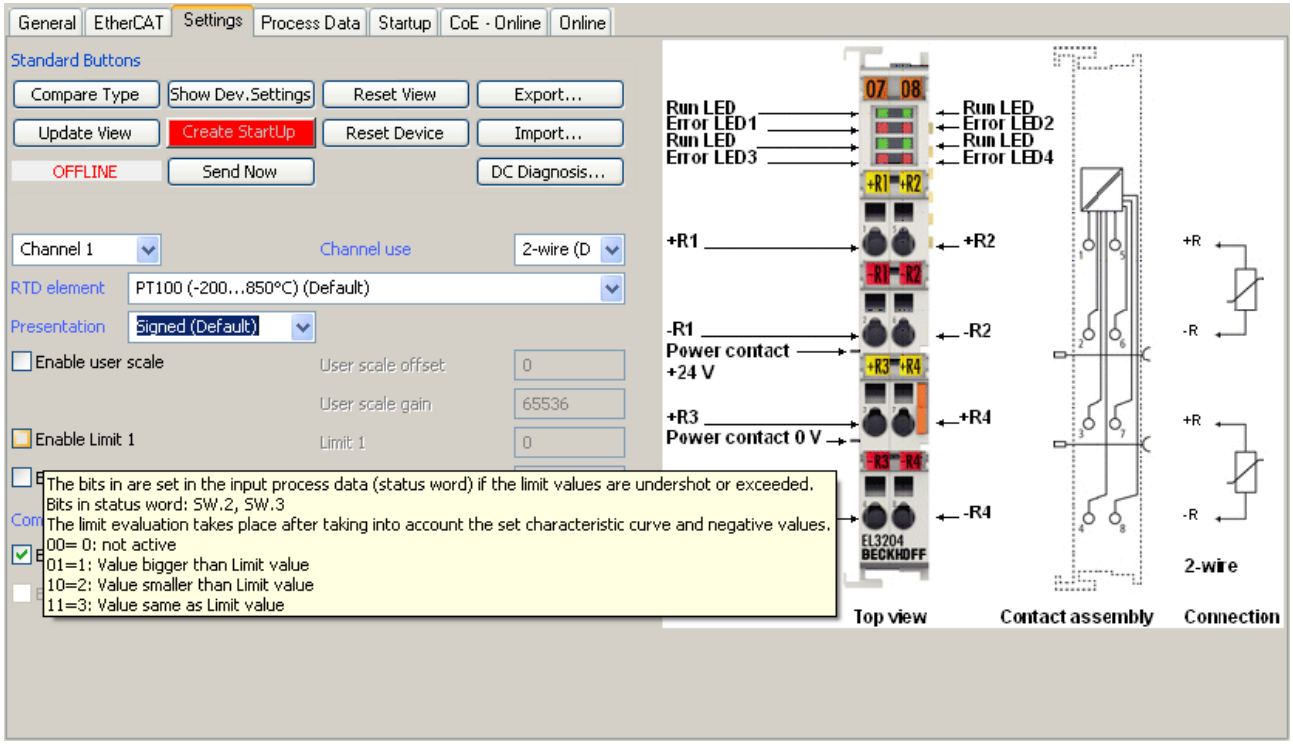


Fig. 134: Example of commissioning aid for an EL3204

This commissioning process simultaneously manages

- CoE parameter directory
- DC/FreeRun mode
- the available process data records (PDO)

The “Process Data”, “DC”, “Startup” and “CoE-Online” tabs previously required for this are still displayed, but it is recommended not to change the automatically generated settings using the commissioning help if this is used.

The commissioning tool does not cover all possible applications of an EL/EP device. If the setting options are not sufficient, the user can make DC, PDO and CoE settings manually as before.

EtherCAT State: automatic default behavior of the TwinCAT System Manager and manual control

After the operating power is switched on, an EtherCAT slave must go through the following states

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT master directs these states in accordance with the initialization routines that are defined for commissioning the device by the ES/XML and user settings (Distributed Clocks (DC), PDO, CoE). See also chapter “Basics of communication, EtherCAT State Machine”. Depending on the configuration effort and overall configuration, booting can take up to a few seconds.

The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

The target state intended by the user and automatically generated by TwinCAT at start-up can be set in the System Manager. As soon as TwinCAT is set to RUN, the TwinCAT EtherCAT master switches to the target states.

Standard setting

The advanced settings of the EtherCAT Master are set as standard:

- EtherCAT Master: OP
- Slaves: OP
  This setting applies equally to all Slaves.

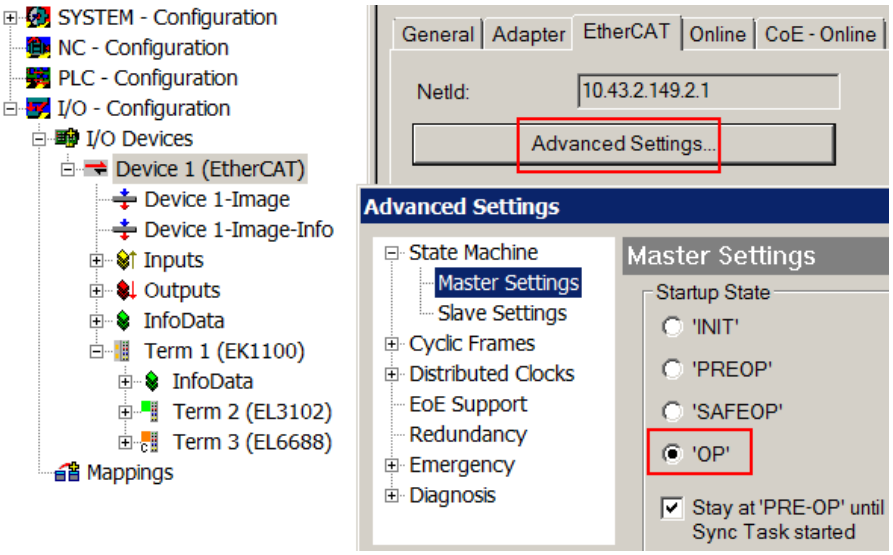


Fig. 135: Default behaviour of the System Manager

In addition, the target state of any particular Slave can be set in the “Advanced Settings” dialogue; the standard setting is again OP.

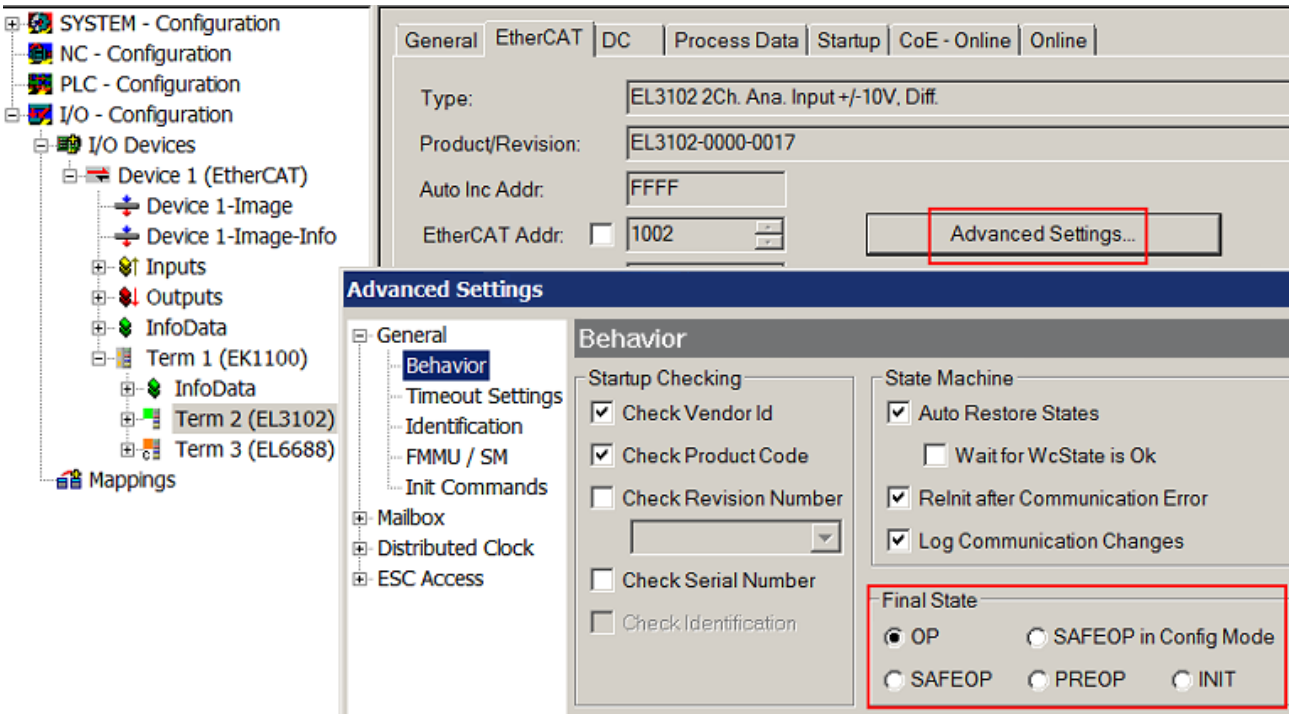


Fig. 136: Default target state in the Slave

Manual Control

There are particular reasons why it may be appropriate to control the states from the application/task/PLC. For instance:

- for diagnostic reasons
- to induce a controlled restart of axes
- because a change in the times involved in starting is desirable

In that case it is appropriate in the PLC application to use the PLC function blocks from the *TcEtherCAT.lib*, which is available as standard, and to work through the states in a controlled manner using, for instance, *FB_EcSetMasterState*.

It is then useful to put the settings in the EtherCAT Master to INIT for master and slave.

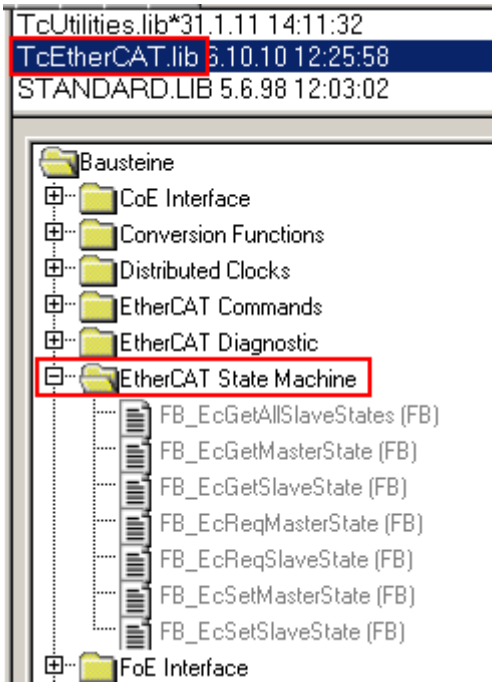
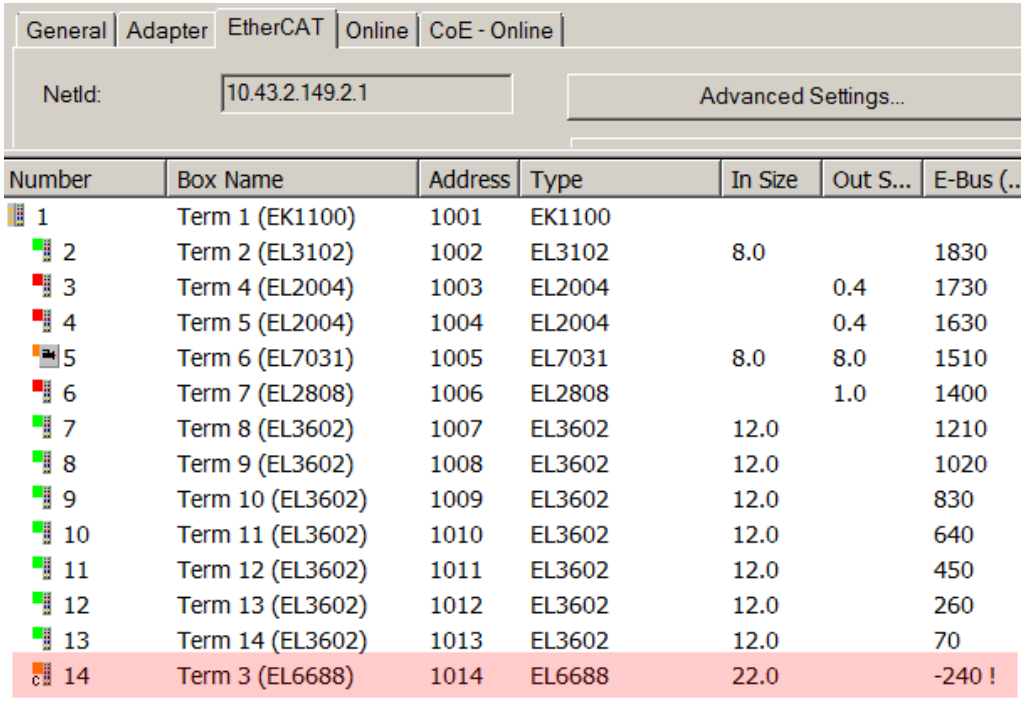


Fig. 137: PLC function blocks

Note E-bus power

EL/ES terminals are placed on the DIN rail at a coupler on the terminal segment. A bus coupler can supply the EL terminals connected to it with the E-bus system voltage of 5 V. As a rule, a coupler can be loaded with up to 2 A. For each EL terminal, information on how much current it requires from the E-bus supply is available online and in the catalog. If the connected terminals require more power than the coupler can supply, power supply terminals (e.g. EL9410) must be installed at the appropriate positions in the terminal segment.

The precalculated theoretical maximum E-bus current is displayed as a column value in the TwinCAT System Manager. An undershoot is indicated by a negative total and an exclamation mark; a power supply terminal must be placed before such a point.



Number	Box Name	Address	Type	In Size	Out S...	E-Bus (..
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL3102)	1002	EL3102	8.0		1830
3	Term 4 (EL2004)	1003	EL2004		0.4	1730
4	Term 5 (EL2004)	1004	EL2004		0.4	1630
5	Term 6 (EL7031)	1005	EL7031	8.0	8.0	1510
6	Term 7 (EL2808)	1006	EL2808		1.0	1400
7	Term 8 (EL3602)	1007	EL3602	12.0		1210
8	Term 9 (EL3602)	1008	EL3602	12.0		1020
9	Term 10 (EL3602)	1009	EL3602	12.0		830
10	Term 11 (EL3602)	1010	EL3602	12.0		640
11	Term 12 (EL3602)	1011	EL3602	12.0		450
12	Term 13 (EL3602)	1012	EL3602	12.0		260
13	Term 14 (EL3602)	1013	EL3602	12.0		70
14	Term 3 (EL6688)	1014	EL6688	22.0		-240 !

Fig. 138: Impermissible exceeding of E-bus current

From TwinCAT 2.11 and above, a warning message "E-Bus Power of Terminal..." is output in the logger window when such a configuration is activated:

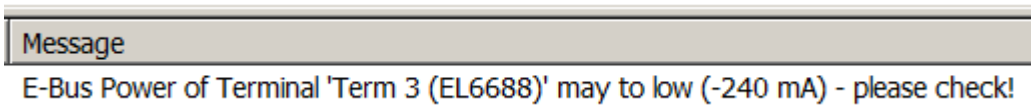


Fig. 139: Warning message for exceeding E-bus current

NOTICE

Malfunction possible!

The E-bus supply of all EtherCAT Terminals of a terminal block must be provided from the same ground potential!

5.4 Process data and operation modes

5.4.1 Parameterization

An ELX318x is parameterized via two dialog boxes/tabs in the TwinCAT System Manager, the Process Data tab (A) for the communication-specific settings and the CoE directory (B) for the settings in the slave. In addition, there are the settings for HART and FDT (see chapters 6 and 7).

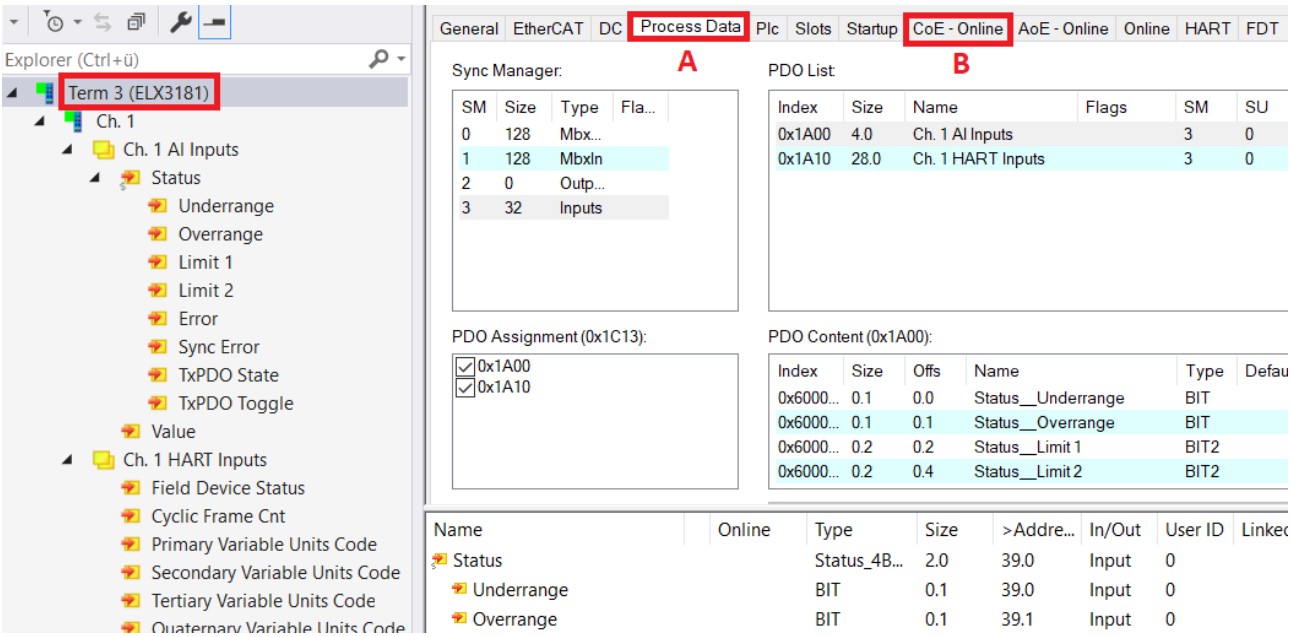


Fig. 140: ELX3181 - parameterization

- Changes to the process data-specific settings are generally only effective after a restart of the EtherCAT master:
  Restart TwinCAT in RUN or CONFIG mode; RELOAD in CONFIG mode
- Changes to the online CoE directory
  - are in general immediately effective
  - are in general stored non-volatile *only* in the terminal/in the slave and should therefore be entered in the CoE StartUp list. This list is processed at each EtherCAT start and the settings are loaded into the slave.

5.4.2 Settings and operating modes

5.4.2.1 Presentation, index 0x80nD

The output of the measured value depends on the settings in Index 80nD:0 AI Advanced Settings Ch.n.

With Index 80nD:11 (Input Type) the input signal can be selected:

- 4...20 mA
- 4...20 mA NAMUR

With Index 80nD:12 (Scaler) the Range can be switched:

- Extended Range (measurement beyond the measurement value range)
- Legacy Range (measurement value range = minimum/maximum value)

The output values depending on these settings are shown in the following table.

Scaler	Extended Range		Legacy Range	Value	
Input Type	4…20 mA	4…20 mA Na-mur	4…20 mA	dez	hex
	21.179 mA	-	20 mA	32767	0x7FFF
	20 mA	20 mA	-	30518	0x7736
	4 mA	4 mA	4 mA	0	0x0000
	0 mA	-	-	-7629	0xE233

5.4.2.2 Undershoot and overshoot of the measuring range (under-range, over-range), index 0x60n0:01, 0x60n0:02

Undershoot and overshoot of the measuring range (under-range, over-range), index 0x60n0:01 [▶ 131], 0x60n0:02 [▶ 131]

Chapter Data stream and correction calculation [▶ 125] contains a clear description of the correction calculation between the raw values and the output values if the limit ranges are exceeded.

5.4.2.3 Limit 1 and limit 2, Swap Limit Bits

Limit 1 and limit 2, index 0x80n0:13, index 0x80n0:14

If the value exceeds or falls below these values, which can be entered in the indices 0x80n0:13 and 0x80n0:14, then the bits in the indices 0x60n0:03 [▶ 131] and 0x60n0:05 [▶ 131]are set accordingly (see example below). The indices 0x80n0:07 or 0x80n0:08 serve to activate the limit value monitoring.

Output limit n (2-bit):

- 0: not active
- 1: Value is smaller than the limit value
- 2: Value is larger than the limit value
- 3: Value is equal to the limit value

Linking in the PLC with 2-bit values

The limit information consists of 2 bits. Limitn can be linked to the PLC or a task in the System Manager.

- PLC:
  IEC61131-PLC contains no 2-bit data type that can be linked with this process data directly. For transferring the limit information, define an input byte (e.g. see Fig. *Input byte definition*) and link the limit to the *VariableSizeMismatch* dialog, as described in chapter Note about the 1-byte status of earlier EtherCAT Terminals.

```
VAR
    byLimit1 AT %I*:BYTE;
END_VAR
```

Fig. 141: Input byte definition

- Additional task
  2-bit variables can be created in the System Manager.

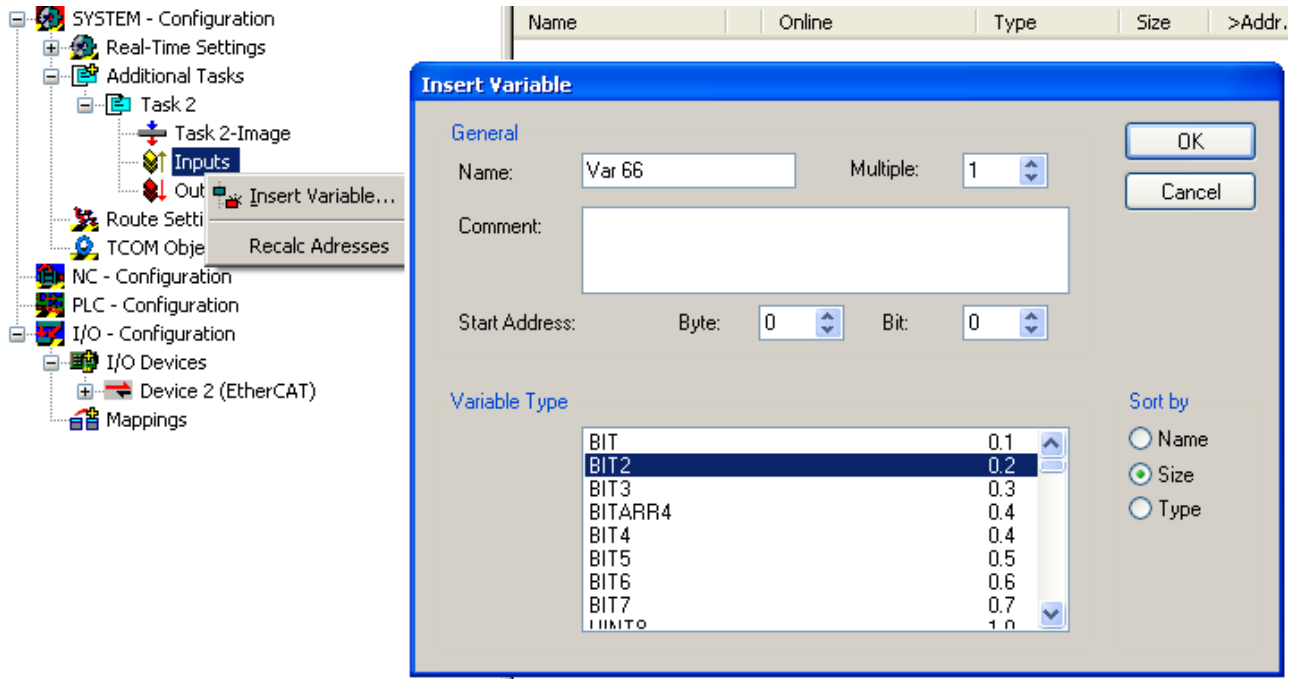


Fig. 142: Linking of 2-bit variable to additional task

Swap Limit index 0x80n0:0E

The limit function can be inverted by *SwapLimitBits* in index 0x80n0:0E.

Output Limit n (2-bit):

***SwapLimitBits* setting**	**Value**
FALSE (default setting)	• 0: not active • 1: value < limit value • 2: value > limit value • 3: Value is equal to the limit value
TRUE	• 0: not active • 1: value > limit value • 2: value < limit value • 3: Value is equal to the limit value

5.4.2.4 Operating modes

The ELX31xx/EPX31xx support three different operation modes:

- Freerun (filter on, timer interrupt)
- Synchronous (filter off, SyncManager interrupt) and
- DC (DC-Sync-Interrupt)

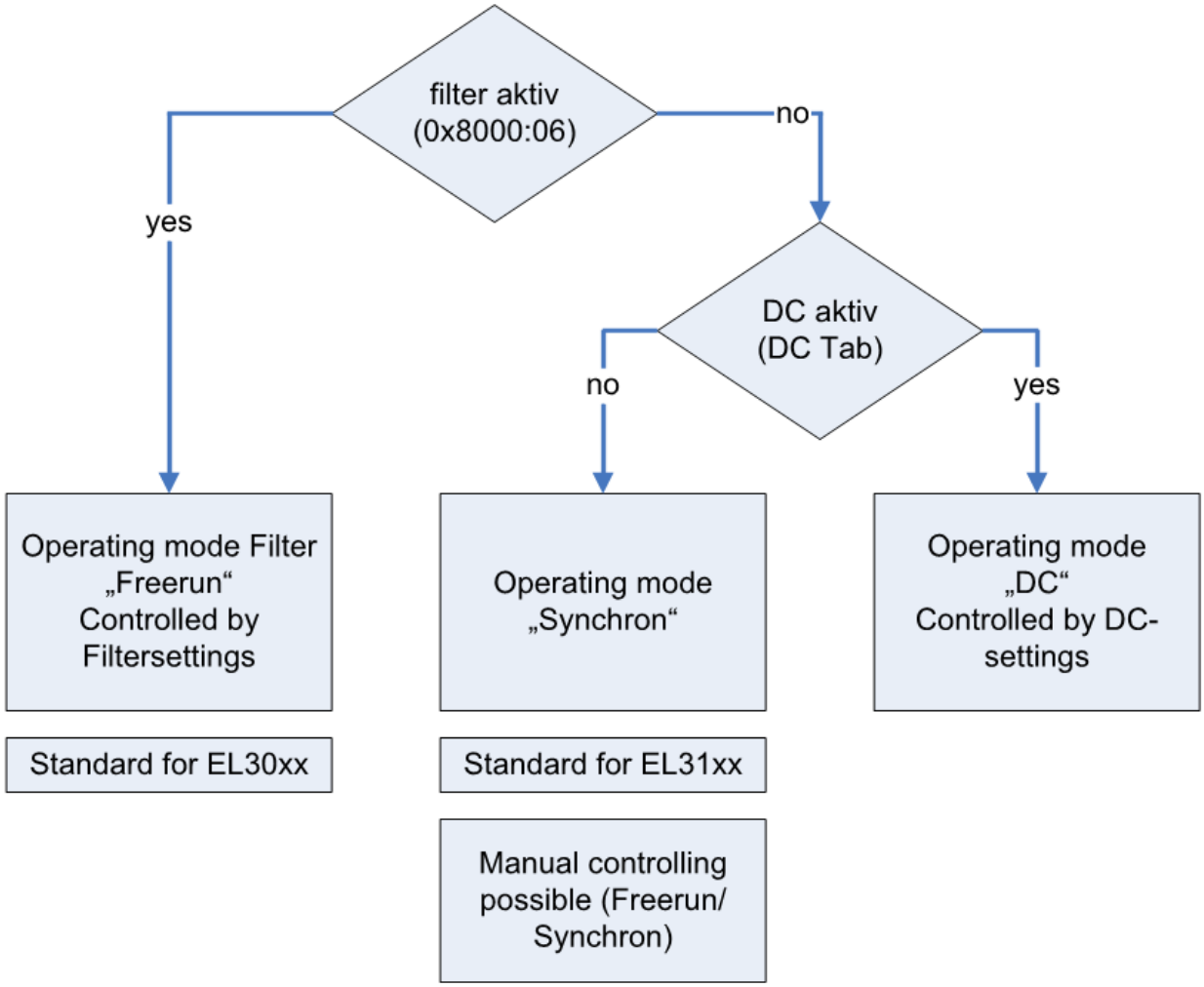


Fig. 143: Relationship of operation modes

The device switches between the Freerun (filter on) and Synchron modes by enabling/disabling the filter via the index. The terminal remains in OP mode during this process. The changeover may result in longer sampling times and step changes in the process data until the filters have assumed a steady state.

DC mode can only be used when the filters are switched off. Likewise, it is not possible to switch the filters on in DC mode. The DC mode is parameterized via the DC tab in the TwinCAT System Manager.

The operation modes of the ELX31xx/EPX31xx

Mode	1 (default)	2	3	4	5	6
Filter (Index: 0x8000:06)	On (default.: 50 Hz FIR)	Off				
Distributed Clocks mode	Off					On
Synchronization mode (Index: 0x1C33:01**, Bit 0+1)**	0: FreeRun (default)	0: FreeRun (default)		1: Frame Triggered (SM3 inputs)		3: DC-mode
FastOp-Mode "CoE" (Index: 0x1C33:01**, Bit 15)**	Off (default)	Off (default)	On	Off (default)	On	Off (default) (FastOp mode not possible in DC)
StartUp entry index 0x1C33:01	0x0000	0x0000	0x8000	0x0001	0x8001	
Update rate	depends on filter setting, automatically set internally in device see following values			EtherCAT cycle time, if value does not fall below the lower setting-dependent limit. See following values for typical limit. Operation with a faster EtherCAT cycle is possible, but in that case the ELX31xx / EPX31xx no longer supplies new data in each cycle.		EtherCAT cycle time, if value does not fall below the lower setting-dependent limit.
typ. data update time ELX31xx/EPX31xx	< 1 s					
Note	Once filtering is enabled, regardless of other settings "FreeRun" = On and "FastOp-Mode" = Off is set in the ELX31xx / EPX31xx.					The notes on the min. EtherCAT cycle time in DC mode must be observed

Combinations of filters, FastOp mode and Synchronization mode

Other combination options of filter, FastOp mode and Synchronization mode are expressly not recommended.

Synchron mode

In synchronous operation process data are generated frame-triggered, so that a new value is available with each PLC cycle. In the ELX31xx/EPX31xx synchron mode is used automatically (filter off, no DC). Minimum cycle time is 1 ms for standard IPCs.

DC operation

In DC mode the analog sampling is triggered by DC-interrupt. As a result, the temporal jitter between two frames is no longer important and the sampling point is the same across the entire system.

The ELX31xx/EPX31xx are to be operated in DC-Input-Based mode.

The Input-Based mode shifts the sync-interrupt in such a way that the process data are ready for collection shortly before the current process data cycle.
If the frame jitter is too large, it is possible that data may be collected twice or there may be interruptions in the transmission. In that case the jitter must be reduced by measures in the TwinCAT system or a slower cycle time must be chosen.

5.4.2.5 Filter operation (FIR and IIR), index 0x80n0:06, 0x80n0:15

Filter operation (FIR and IIR), index 0x80n00:06, 0x80n00:15

The ELX31xx/EPX31xx are equipped with a digital filter which, depending on the setting, can assume the characteristics of a finite impulse response filter (*F*inite *I*mpulse *R*esponse filter, *FIR filter*) or an infinite impulse response filter (*I*nfinite *I*mpulse *R*esponse filter, *IIR filter)*. The filter is disabled on delivery. Please observe the following note regarding activation with index 0x8000:06 [▶ 128].

Activation of the filter with index 0x8000:06 and setting of the filter characteristics via index 0x8000:15

The filter frequencies are set for all channels of the ELX31xx/EPX31xx centrally via index 0x8000:15 [▶ 128] (channel 1). The corresponding indices 0x80n00:15 of the other channels have no parameterization function.

FIR filter

Parameterized via the index 0x8000:15 [▶ 128].

The filter works as a notch filter and determines the conversion time of the device. The higher the filter frequency, the faster the conversion time. A 50 Hz and a 60 Hz filter are available.
Notch filter means that the filter has zeros (notches) in the frequency response at the filter frequency and multiples thereof, i.e. it attenuates the amplitude at these frequencies.

The FIR filter operates as a non-recursive filter.

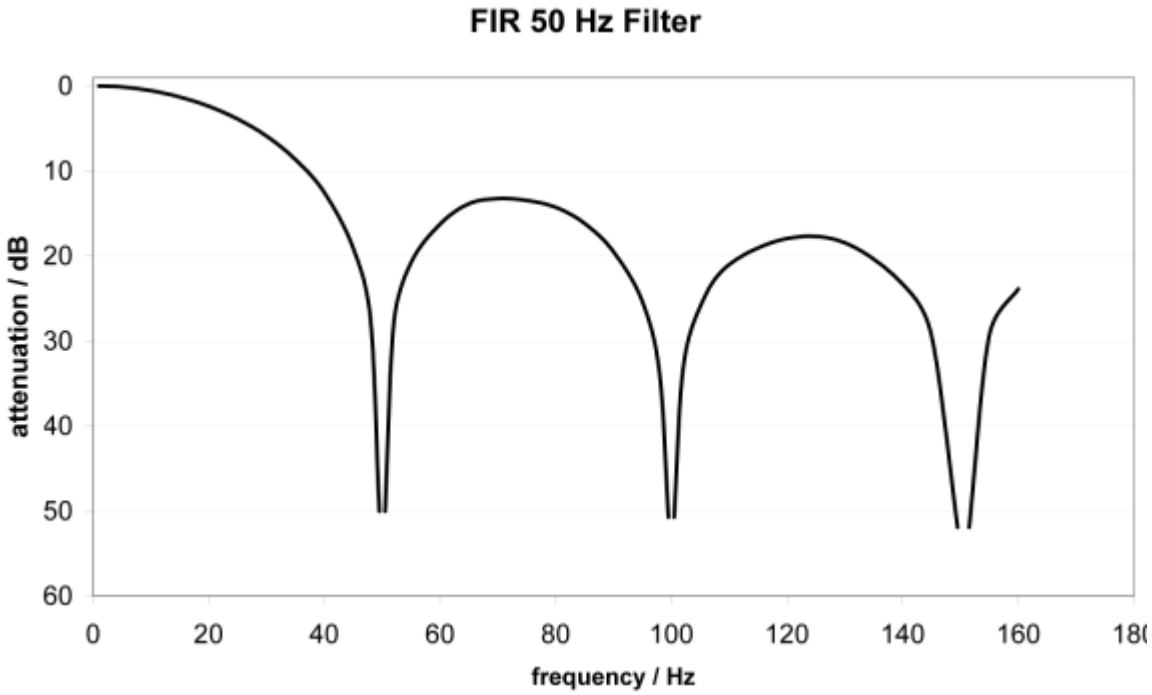


Fig. 144: Attenuation curve notch.filter at 50 Hz

Filter data FIR - Filter (one to four channel devices)			
Filter	**Attenuation**	**Cut-off frequency (-3 dB)**	**Conversion time**
50 Hz FIR	> 50 dB	22 Hz	625 µs
60 Hz FIR	> 45 dB	26 Hz	521 µs

IIR filter

The filter with IIR characteristic is a time-discrete, linear, time-invariant 1st order low-pass filter (-20dB/decade), which can be set in 8 levels, i.e. cut-off frequencies (level 1 = weak recursive filter, up to level 8 = strong recursive filter)
The IIR can be understood to be a sliding average value calculation after a low-pass filter.
By means of the synchronization mode *FreeRun*, the IIR filter works with an internal cycle time of 1 ms.

IIR filter	ELX31xx/EPX31xx, sampling time 1 ms
	Cut-off frequency (-3 dB)
IIR 1	168 Hz
IIR 2	88 Hz
IIR 3	43 Hz
IIR 4	21 Hz
IIR 5	10.5 Hz
IIR 6	5.2 Hz
IIR 7	2.5 Hz
IIR 8	1.2 Hz

5.4.3 Process data

Overview
• Interpretation of value & status variable [▶ 120]
• Status word [▶ 120]
• Changeover of process data sets [▶ 121]
• Note about TwinCAT 2.10 [▶ 122]
• Password protection for user calibration [▶ 122]

The ELX318x offers three different process data for transmission per analog channel: the analog value Value (16 bit), the status information (16 bit) and cyclic process data via HART Cmd3 (28 byte). The transmission of individual status information as well as individual channels can be deactivated in the Slots tab, these changes are effective after activation and EtherCAT restart or reload.

Two process data types are available for selection for the ELX318x in the *AI Channel* slot:

- Standard: Standard setting, value (16 bit) and status information (8 or 16 bit) are transmitted per channel.
- Compact: only the value (16 bit) is transmitted per channel

A process data type is available for selection for the ELX318x in the *HART Cmd3* slot:

- HART Cmd3: additional process values and their units (28 bytes in total) are transmitted via the HART protocol for each channel (see chapter 6).

Interpretation of value and status variable

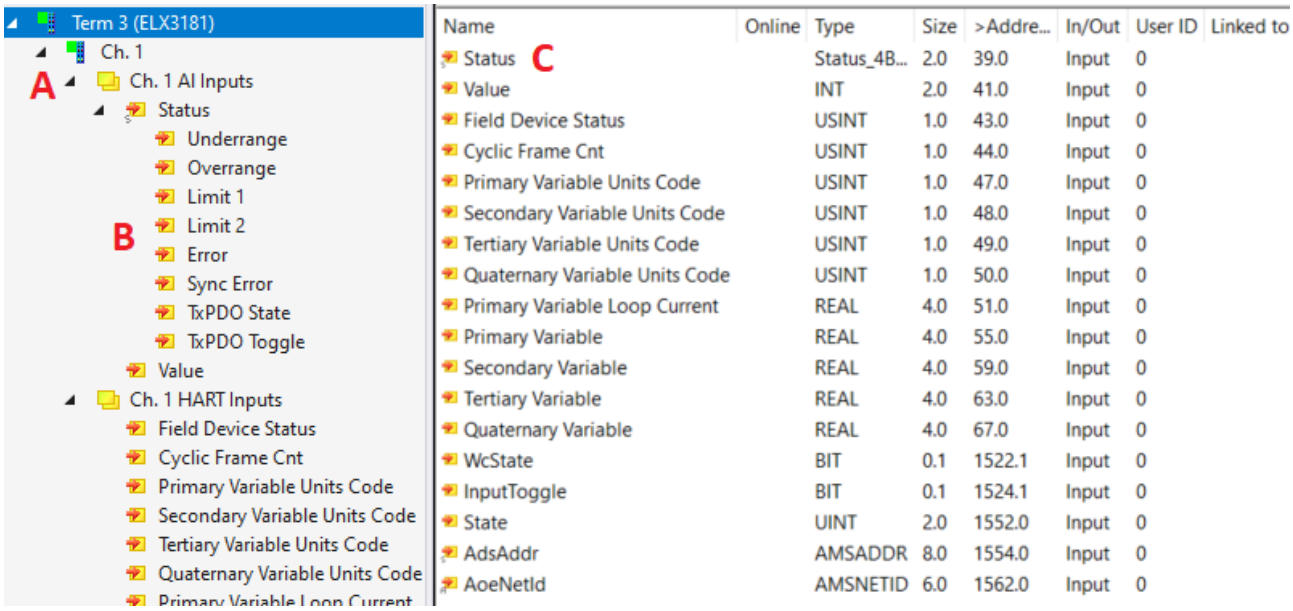


Fig. 145: ELX3181 - Default process data

The plain text display of the bit meanings of the status word is particularly helpful not only in commissioning but also for linking to the PLC program.
By right-clicking on the status variable in the configuration tree (A), the structure can be opened for linking (B).

In order to be able to read the bit meanings in plain text in the online display (C), the button

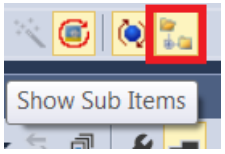


Fig. 146: Show sub items

can be used to show all subvariables, including the structure content of the status word, see Fig. *ELX3181 - Display of the subvariables*

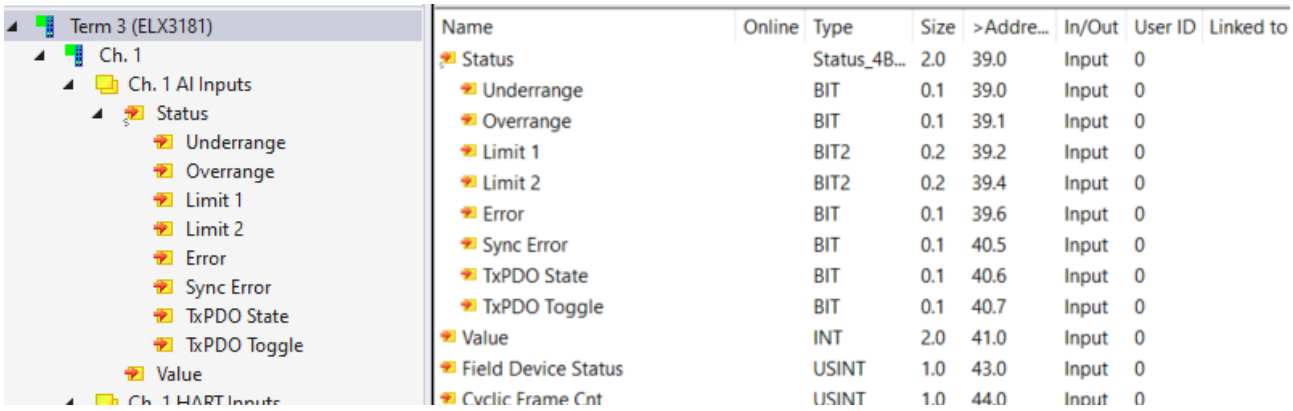


Fig. 147: ELX3181 - Display of the subvariables

Control and status word

Status word

The status word (SW) is located in the input process image, and is transmitted from the device to the controller.

Bit	SW.15	SW.14	SW.13	SW.12	SW.11	SW.10	SW.9	SW.8
Name	TxPDO Toggle	TxPDO State	Sync error	-	-	-	-	-

Bit	SW.7	SW.6	SW.5	SW.4	SW.3	SW.2	SW.1	SW.0
Name	-	ERROR	Limit 2		Limit 1		Overrange	Underrange

Legend

Bit	Name	Description	
SW.15	TxPDO Toggle	1_{bin}	Toggles with each new analog process value
SW.14	TxPDO State	1_{bin}	TRUE in the case of an internal error
SW.13	Sync error	1_{bin}	TRUE (DC mode): a synchronization error occurred in the expired cycle.
SW.6	ERROR	1_{bin}	General error bit, is set together with overrange and underrange
SW.5	Limit 2	1_{bin}	See Limit
SW.4		1_{bin}	
SW.3	Limit 1	1_{bin}	See Limit
SW.2		1_{bin}	
SW.1	Overrange	1_{bin}	Analog input signal is above the upper permissible. Threshold for this device
SW.0	Underrange	1_{bin}	Analog input signal lies below the upper permissible threshold for this device

Control word

The ELX31xx/EPX31xx do not have a control word

Changeover of process data sets

The process data to be transmitted (PDO, ProcessDataObjects) can be selected by the user

- completely for all channels via the selection dialog “Predefined PDO Assignment” (all TwinCAT versions)
- selective for individual PDOs, taking into account the excluded elements.

Selection dialog *Predefined PDO Assignment*

The ELX31xx/EPX31xx do not have any Predefined PDO Assignments.

Selective PDO selection

All TwinCAT versions support the selective selection of individual PDOs, as defined in the XML description. Exclusion criteria prevent irregular combinations.

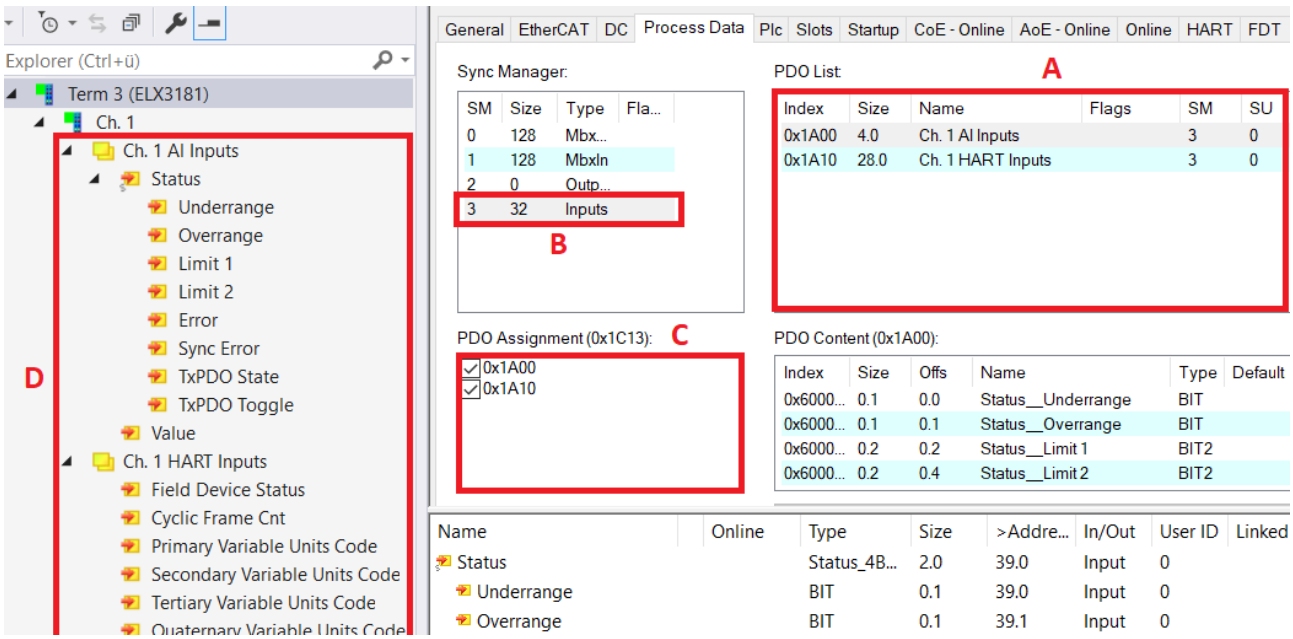


Fig. 148: Selective PDO selection

By selecting the necessary Input SyncManager in (B), the PDO assignment under (C) can be carried out manually. The process data can then be linked in the TwinCAT tree (D).

Note about TwinCAT 2.10

The structured representation of ELX31xx/EPX31xx as shown in Fig. *Selective PDO selection taking the ELX3152 as an example* is due to a corresponding interpretation of the variable names. This function does not exist in TwinCAT 2.10 yet, which is why only element-wise linking is possible there.

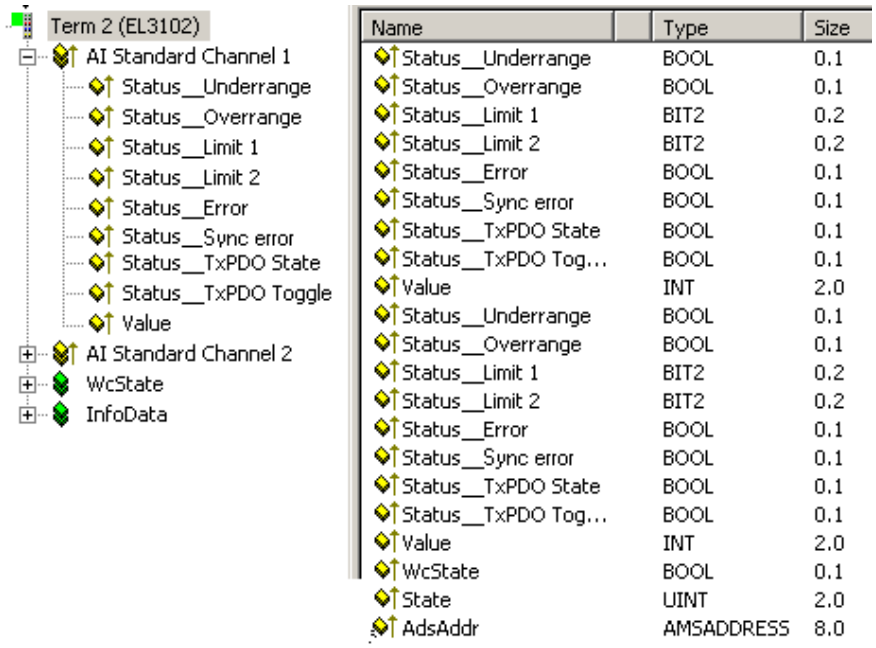


Fig. 149: Element-orientated process image under TwinCAT 2.10

Password protection for user data

Some user data are protected against unwanted or inadvertent writing by an additional password to be entered in CoE 0xF009:

- CoE write accesses by the user, PLC or startup entries in *Single* or *CompleteAccess* mode
- Overwrite the values by *RestoreDefaultParameter* Access to 0x80n0 (or 0x80nD, if available)

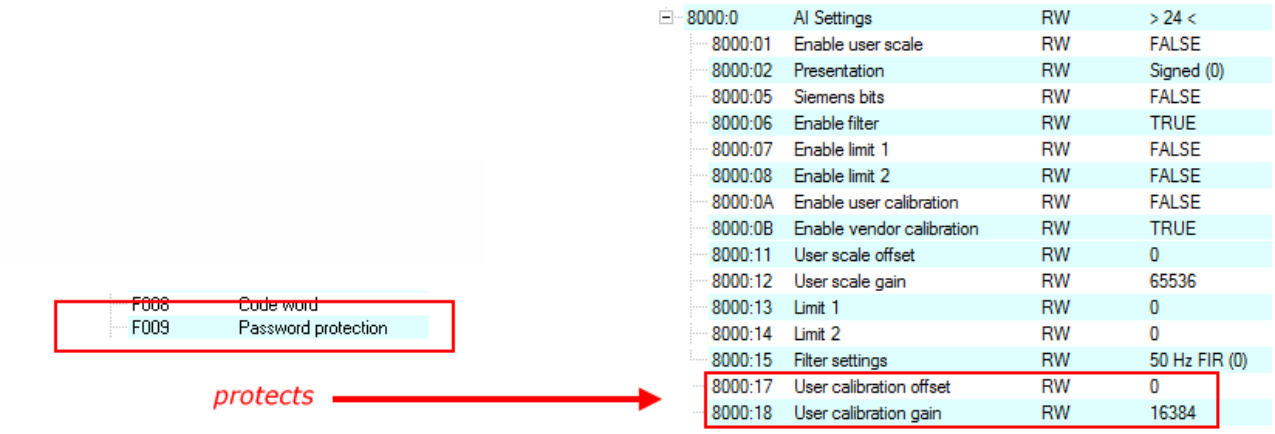


Fig. 150: Password protection for the 0x8000:17 and 0x8000:18 entries (example)

The password protection applies to the following user data

Devices	Protectable CoE object
ELX3152, ELX3158, ELX3181, ELX3184, EPX3158, EPX3184	0x80n00:17 (User Calibration Offset) 0x80n00:18 (User Calibration Gain) 0x80nD:17 (Low Range Error) 0x80nD:18 (High Range Error)

Use of CoE 0xF009

- Entering 0x12345678 enables the password protection → Object shows "1" (enabled)
  Protected objects can now no longer be changed, no error message occurs during a write access!
- Entering 0x11223344 disables password protection → Object displays "0" (disabled)

Code word Index 0xF008

This CoE object has only a vendor-specific function and is not intended for the user.

5.4.4 Data stream and measurement ranges

Data stream

The flow chart below (Fig. *Data stream of the ELX31xx/EPX31xx*) shows the data stream of the ELX31xx/EPX31xx (processing of the raw data).

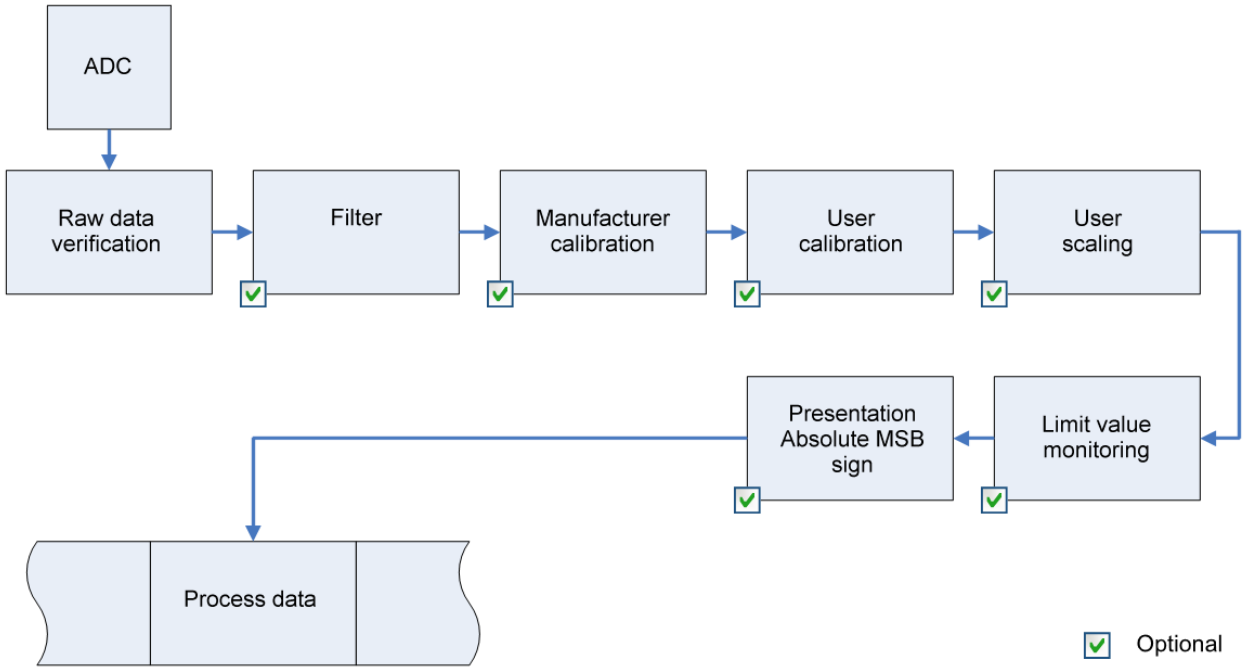


Fig. 151: Data stream of the ELX31xx/EPX31xx

Calculation of process data

"Calibration"

The concept "calibration", which has historical roots at Beckhoff, is used here even if it has nothing to do with the deviation statements of a calibration certificate. Actually, this is a description of the vendor or customer calibration data/adjustment data used by the device during operation in order to maintain the assured measuring accuracy.

The ELX31xx/EPX31xx permanently record measured values and place the raw values of their A/D converter into the ADC raw value object 0x80nE:01. After each recording of the analog signal, the correction calculation takes place with the vendor and user calibration data as well as the user scaling, if these are activated (see fig. Data stream of the ELX31xx/EPX31xx).

Calculation	Designation
$X_F = f(X_{ADC})$	Output value after the filter
$Y_H = (X_F - B_H) \times A_H \times 2^{-14}$	Measured value after vendor calibration,
$Y_A = (Y_H - B_A) \times A_A \times 2^{-14}$	Measured value after vendor and user calibration
$Y_S = Y_A \times A_S \times 2^{-16} + B_S$	Measured value following user scaling

Name	Name	Index
X_{ADC}	Output value of the A/D converter	0x80nE:01
X_F	Output value after the filter	-
B_H	Vendor calibration offset (not changeable)	0x80nF:01
A_H	Vendor calibration gain (not changeable)	0x80nF:02
B_A	User calibration offset (can be activated via index 0x80n0:0A)	0x80n00:17
A_A	User calibration gain (can be activated via index 0x80n0:0A)	0x80n00:18
B_S	User scaling offset (can be activated via index 0x80n00:01)	0x80n00:11
A_S	User scaling gain (can be activated via index 0x80n00:01)	0x80n00:12
Y_S	Process data for controller	-

Measurement result

The accuracy of the result may be reduced if the measured value is smaller than 32767 / 4 due to one or more multiplications.

Measurement ranges

The diagrams at the bottom show the output values of the measurement ranges and the behavior if the limit ranges are exceeded.

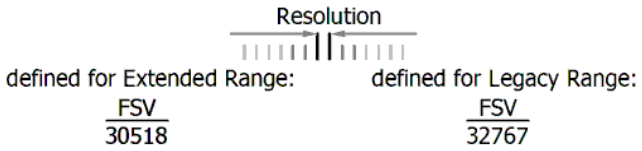
ELX318x scaler: extended range / legacy range

The ELX318x offers the scales Extended Range and Legacy Range (Scaler, AI Advanced settings Object 0x80nD:12 [▶ 128]).

Extended Range:
This scaling type allows to exceed or fall below the actual measuring range by approx. 7%. The technically usable range is thus -107% to +107% of the respective measuring range end value. Example:
Measuring range end value = 0...20 mA, then the technical measuring range is approx. 0...21.4 mA.

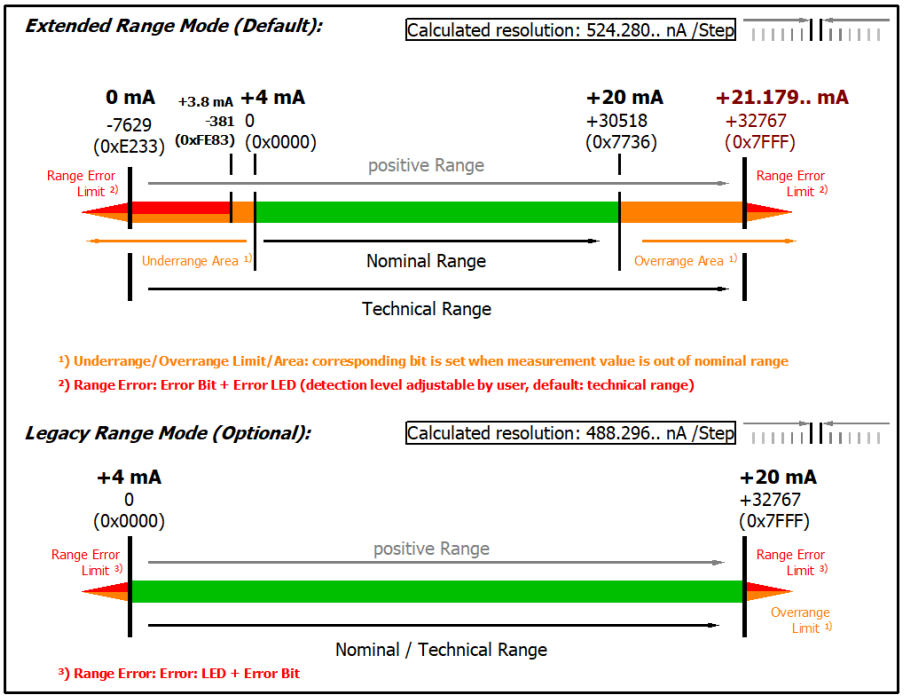
Legacy Range:
The Legacy Range, on the other hand, reflects the conventional range from -100% to +100%.
+100% corresponds to +32767
-100% corresponds to -32768.

For the Extended Range 100% of a 16 bit PDO value ±30518 (0x7736) have been defined. Therefore resulting is the meaning of a bit given by the (user selected measurement range) respective full scale value (FSV) as follows:



Diagrams of all measurement ranges are shown as follows:

Measurement range 4...20 mA (current loop):



Technical note: The detection level for underrange and range error of 0 value area is located at 3.8 mA (-1% of the FSV full scale vale). This has been configured to prevent a misleading setting of the error bit.

Fig. 152: ELX3181 - measurement range 4…20 mA

Measurement range 4...20 mA/ NAMUR NE43 (current loop):

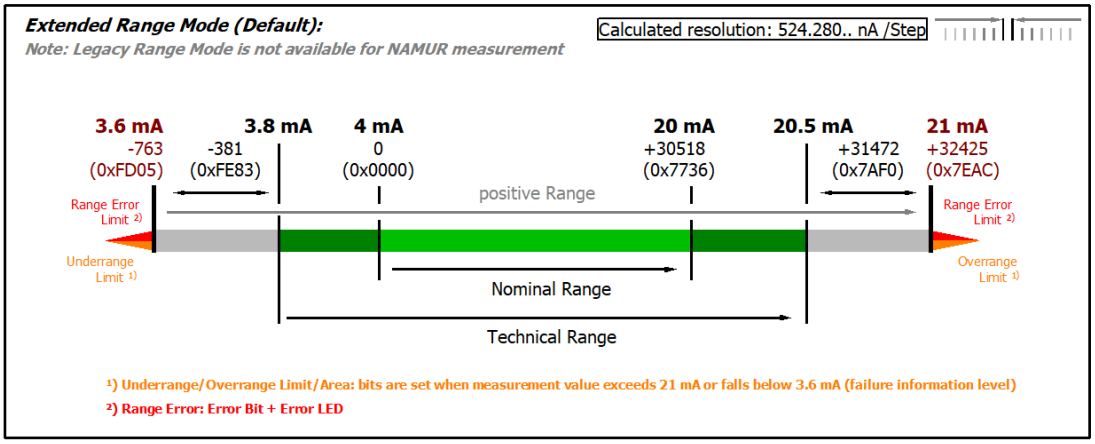


Fig. 153: ELX3181 - measurement range 4…20 mA (NAMUR NE43)

Calibration

Vendor calibration, index 0x80n0:0B

The vendor calibration is enabled via index 0x80n0:0B. Parameterization takes place via the indices

- 0x80nF:01 Vendor calibration: Offset
- 0x80nF:02 Vendor calibration: Gain

User calibration, index 0x80n0:0A

The user calibration is enabled via index 0x80n0:0A. Parameterization takes place via the indices

- 0x80n0:17 User calibration: Offset
- 0x80n0:18 User calibration: Gain

User scaling, index 0x80n00:01

The user scaling is enabled via index 0x80n00:01. Parameterization takes place via the indices

- 0x80n00:11 User scaling: offset
- 0x80n00:12 User scaling: gain

Vendor calibration

The vendor reserves the right to calibrate the ELX315x/EPX31xx. Therefore, the vendor calibration cannot be changed.

5.5 CoE object description and parameterization

EtherCAT XML Device Description

The presentation 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.

Overview

The CoE overview contains objects for different intended applications:

Object overview
• Restore object [▶ 127]
• Configuration data [▶ 128]
• Information and diagnostic data [▶ 130]
• Input data [▶ 131]
• Command objects [▶ 131]
• Output data [▶ 131]
• Standard objects [▶ 132]

5.5.1 Restore object

Index 1011 Restore default parameters

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

5.5.2 Configuration data

Index 80n0 AI settings (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	AI Settings	Maximum subindex	UINT8	RO	0x18 (24_{dec})
80n0:01	Enable user scale	User scale is active.	BOOLEAN	RW	0x00 (0_{dec})
80n0:06	Enable filter	Enable filter, which makes PLC-cycle-synchronous data exchange unnecessary	BOOLEAN	RW	0x00 (0_{dec})
80n0:07	Enable limit 1	Limit 1 enabled	BOOLEAN	RW	0x00 (0_{dec})
80n0:08	Enable limit 2	Limit 2 enabled	BOOLEAN	RW	0x00 (0_{dec})
80n0:0A	Enable user calibration	Enabling of the user calibration	BOOLEAN	RW	0x00 (0_{dec})
80n0:0B	Enable vendor calibration	Enabling of the vendor calibration	BOOLEAN	RW	0x01 (1_{dec})
80n0:0E	Swap limit bits	Changing of the Limit Bits	BOOLEAN	RW	0x00 (0_{dec})
80n0:11	User scale offset	User scaling offset	INT16	RW	0x0000 (0_{dec})
80n0:12	User scale gain	User scaling gain. The gain is represented in fixed-point format, with the factor 2^{-16}. The value 1 corresponds to 65536_{dec} (0x00010000) and is limited to ±0x7FFFF	INT32	RW	0x00010000 (65536_{dec})
80n0:13	Limit 1	First limit value for setting the status bits	INT16	RW	0x0000 (0_{dec})
80n0:14	Limit 2	Second limit value for setting the status bits	INT16	RW	0x0000 (0_{dec})
80n0:15	Filter settings	This object determines the digital filter settings, if it is active via Enable filter (index 0x80n0:06 [▶ 128]). The possible settings are sequentially numbered. 0: 50 Hz FIR 1: 60 Hz FIR 2: IIR 1 3: IIR 2 4: IIR 3 5: IIR 4 6: IIR 5 7: IIR 6 8: IIR 7 9: IIR 8	UINT16	RW	0x0000 (0_{dec})
80n0:17	User calibration offset	User offset compensation	INT16	RW	0x0000 (0_{dec})
80n0:18	User calibration gain	User calibration gain	INT16	RW	0x4000 (16384_{dec})

The filter characteristics are set via index 0x8000:15 [▶ 128]

The filter frequencies are set for all channels of the ELX318x terminals centrally via index 0x8000:15 [▶ 128] (channel 1). All other corresponding indices 0x80n0:15 have no parameterization function!

Index 80nD AI Advanced settings (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning		Data type	Flags	Default
80nD:0	AI Advanced Settings	Maximum subindex		UINT8	RO	0x14 (20_{dez})
80nD:11	Input Type	Measurement mode, allowed values:		UINT16	RW	0x0002 (2_{dez})
		0x12	0..20 mA (not for ELX3158)			
		0x13	4..20 mA			
		0x14	4..20 mA (NAMUR),			
80nD:12	Scaler	Scaling range, allowed values:		UINT16	RW	0x0000 (0_{dez})
		0x00	Extended Range			
		0x03	Legacy Range (not for input type 4…20 mA NAMUR.)			
80nD:17	Low Range Error	Lower threshold for setting the error bit and error led		INT32	RW	Dependent on 80nD:11
80nD:18	High Range Error	Upper threshold for setting the error bit and error led		INT32	RW	Dependent on 80nD:11

Index 0x80n0 HART settings (n=8: channel 1 ... n=B: channel 4)

Index (hex)	Name	Meaning		Data type	Flags	Default
80n0:0	HART Settings	Maximum subindex		UINT8	RO	0x06 (6_{dez})
80n0:01	Preamble	Preamble length:		UINT8	RW	0x00 (0_{dez})
		Automatic allocation	0			
		Allowed values	5…20			
80n0:02	MasterMode	Allowed values:		UINT8	RW	0x00 (0_{dez})
		Primary	0			
		Secondary	1			
80n0:03	PollingAddress	Allowed values:		UINT8	RW	0x00 (0_{dez})
		0…63				
80n0:04	PollingTime	Unit in seconds, allowed values:		UINT8	RW	0x03 (3_{dez})
		1…64				
80n0:05	MaxRetry	Allowed values:		UINT8	RW	0x03 (3_{dez})
		3…10				
80n0:06	MasterCtrl	reserved		UINT8	RW	0x00 (0_{dez})

Index 80nE AI Internal data (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
80nE:0	AI internal data	Maximum subindex	UINT8	RO	0x01 (1_{dec})
80nE:01	ADC raw value	ADC raw value	UINT16	RO	-

Index 80nF AI Vendor data (for 0 ≤ n ≤ 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
80nF:0	AI vendor data	Maximum subindex	UINT8	RW	0x02 (2_{dec})
80nF:01	Calibration offset	Offset (vendor calibration)	INT16	RW	0x0000 (0_{dec})
80nF:02	Calibration gain	Gain (vendor calibration)	INT16	RW	0x4000 (16384_{dec})

5.5.3 Information and diagnostic data

Index 0x90n0 HART Info data (n=8: channel 1 ... n=B: channel 4)

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:0	HART Info data	Maximum subindex	UINT8	RO	0x11 (17_{dez})
90n0:01	ExtendetDeviceType	Expanded Device Type	UINT16	RO	0x0000 (0_{dez})
90n0:02	MinNrOfPreamblesReq	Minimum number of Preambles required for the request message from the Master to the Slave.	UINT8	RO	0x00 (0_{dez})
90n0:03	HartMajorRevNr	HART Protocol Major Revision Number implemented by this device.	UINT8	RO	0x00 (0_{dez})
90n0:04	DeviceRevLevel	Device Revision Level	UINT8	RO	0x00 (0_{dez})
90n0:05	SoftwareRevLevel	Software Revision Level for this device	UINT8	RO	0x00 (0_{dez})
90n0:06	HardwareRevLevel	Hardware Revision Level of the electronics in this particular device.	UINT8	RO	0x00 (0_{dez})
90n0:07	PhysicalSignalingCode	Allowed values: 0 Bell 202 Current 1 Bell 202 Voltage 2 RS-485 3 RS-232 4 Wireless 6 Special	UINT8	RO	0x00 (0_{dez})
90n0:08	Flags	HART Flags	UINT8	RO	0x00 (0_{dez})
90n0:09	DeviceId	Device ID	OCTET-STRING[3]	RO	{0}
90n0:0A	MinNrOfPreamblesResp	Minimum number of preambles to be sent with the response message from the slave to the master	UINT8	RO	0x00 (0_{dez})
90n0:0B	MaxNrOfDevVariables	Maximum Number of Device Variables.	UINT8	RO	0x00 (0_{dez})
90n0:0C	ConfigChangeCounter	Configuration Change Counter	UINT16	RO	0x0000 (0_{dez})
90n0:0D	ExtFieldDeviceStatus	Extended Field Device Status	UINT8	RO	0x00 (0_{dez})
90n0:0E	DeviceProfile	Device Profile	UINT8	RO	0x00 (0_{dez})
90n0:0F	ManuIdentCode	Manufacturer Identification Code	UINT16	RO	0x0000 (0_{dez})
90n0:10	PrivLabelDistCode	Private Label Distributor Code	UINT16	RO	0x0000 (0_{dez})
90n0:11	SlavePollingAdress	Current slave polling address	UINT8	RO	0x00 (0_{dez})

Index 0xA0n0 HART Diag data (n=8: channel 1 ... n=B: channel 4)

Index (hex)	Name	Meaning	Data type	Flags	Default
A0n0:0	HART Diag data	Maximum subindex	UINT8	RO	0x04 (4_{dez})
A0n0:01	RcvFrameError	Received Frame Error Counter	UINT8	RO	0x00 (0_{dez})
A0n0:02	RcvCheckSumError	Received CheckSum Error Counter	UINT8	RO	0x00 (0_{dez})
A0n0:03	RcvTimeOutError	Received Timeout Error Counter	UINT8	RO	0x00 (0_{dez})
A0n0:04	DataExchState	DataExchange State	UINT8	RO	0x00 (0_{dez})

Index 0xF900 Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
F900:0	Info data	Maximum subindex	UINT8	RO	0x01 (1_{dez})
F900:01	HART Version	Master HART Version	UINT16	RO	-

5.5.4 Input data

Index 60n0 AI Inputs (for 0 ≤ n ≤ 7)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	AI inputs	Maximum subindex	INT16	RO	0x11 (17_{dec})
60n00:01	Underrange	Value below measuring range.	BOOLEAN	RO	0x00 (0_{dec})
60n0:02	Overrange	Measuring range exceeded.	BOOLEAN	RO	0x00 (0_{dec})
60n0:03	Limit 1	Limit value monitoring Limit 1 0: not active 1: value is less than limit value 1 2: value is greater than limit value 1 3: value is equal to limit value 1	BIT2	RO	0x00 (0_{dec})
60n0:05	Limit 2	Limit value monitoring Limit 2 0: not active 1: value is less than limit value 2 2: value is greater than limit value 2 3: value is equal to limit value 2	BIT2	RO	0x00 (0_{dec})
60n0:07	Error	The error bit is set if the data is invalid (over-range, under-range)	BOOLEAN	RO	0x00 (0_{dec})
60n0:0E	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle. This means a SYNC signal was triggered in the I/O device, although no new process data were available (0=OK, 1=NOK).	BOOLEAN	RO	0x00 (0_{dec})
60n0:0F	TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).	BOOLEAN	RO	0x00 (0_{dec})
60n0:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0_{dec})
60n00:11	Value	Analog input date	INT16	RO	0x0000 (0_{dec})

5.5.5 Command objects

Index 0x60n0 HART Command 3 (n=8: channel 1 ... n=B: channel 4)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	HART Cmd3	Maximum subindex	UINT8	RO	0x0D (13_{dez})
60n0:01	Field Device Status	Represent the current state of the slave	UINT8	RO	0x00 (0_{dez})
60n0:02	Cyclic Frame Cnt	Cyclic Frame Counter	UINT8	RO	0x00 (0_{dez})
60n0:05	Primary Variable Units Code	Primary Variable Units Code (refer to HART 'Common Table Specification')	UINT8	RO	0x00 (0_{dez})
60n0:06	Secondary Variable Units Code	Secondary Variable Units Code (refer to HART 'Common Table Specification')	UINT8	RO	0x00 (0_{dez})
60n0:07	Tertiary Variable Units Code	Tertiary Variable Units Code (refer to HART 'Common Table Specification')	UINT8	RO	0x00 (0_{dez})
60n0:08	Quaternary Variable Units Code	Quaternary Variable Units Code (refer to HART 'Common Table Specification')	UINT8	RO	0x00 (0_{dez})
60n0:09	Primary Variable Loop Current	Primary Variable Loop Current (units of milli-amperes)	REAL32	RO	0x00000000 (0_{dez})
60n0:0A	Primary Variable	Primary Variable (vendor specific)	REAL32	RO	0x00000000 (0_{dez})
60n0:0B	Secondary Variable	Secondary Variable (vendor specific)	REAL32	RO	0x00000000 (0_{dez})
60n0:0C	Tertiary Variable	Tertiary Variable (vendor specific)	REAL32	RO	0x00000000 (0_{dez})
60n0:0D	Quaternary Variable	Quaternary Variable (vendor specific)	REAL32	RO	0x00000000 (0_{dez})

5.5.6 Output data

The EPX3184 has no output objects.

5.5.7 Standard objects

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	0x00001389 (5001_{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	ELX3181

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	-
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	-
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special device number, the High Word (bit 16-31) refers to the device description	UINT32	RO	-
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	-

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	-

Index 180n (AI) TxPDO-Par (for 0 ≤ n ≤ 7, depends on number of channels)

Index (hex)	Name	Meaning	Data type	Flags	Default	
180n:0	AI TxPDO-Par Standard	PDO Parameter TxPDO	UINT8	RO	0x06 (6_{dez})	
180n:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with this PDO	OCTET-STRING[2]	RO	n=	Wert:
					0	0x011A
					1	0x001A

Index 1A0n AI TxPDO-Map Standard (for n = 0; 2, 4, 6; p = 0, 1, 2, 3)

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0n:0	AI TxPDO-Map standard	PDO Mapping TxPDO	UINT8	RO	0x9 (9_{dec})
1A0n:01	SubIndex 001	1. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x01 (Underrange))	UINT32	RO	0x60p0:01, 1
1A0n:02	SubIndex 002	2. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x02 (Overrange))	UINT32	RO	0x60p0:02, 1
1A0n:03	SubIndex 003	3. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x03 (Limit 1))	UINT32	RO	0x60p0:03, 2
1A0n:04	SubIndex 004	4. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x05 (Limit 2))	UINT32	RO	0x60p0:05, 2
1A0n:05	SubIndex 005	5. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x07 (Error))	UINT32	RO	0x60p0:07, 1
1A0n:06	SubIndex 006	6. PDO Mapping entry (7 bit align)	UINT32	RO	0x0000:00, 7
1A0n:07	SubIndex 007	9. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x0F (TxPDO State))	UINT32	RO	0x6000:0F, 1
1A0n:08	SubIndex 008	10. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6000:10, 1
1A0n:09	SubIndex 009	11. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x11 (Value))	UINT32	RO	0x6000:11, 16

Index 1A0n AI TxPDO-Map Compact (for n = 1, 3, 5 ... F; p = 0, 1, 2 ... 7)

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0n:0	AI TxPDO-Map Compact	PDO Mapping TxPDO	UINT8	RO	0x01 (1_{dec})
1A0n:01	SubIndex 001	1. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x11 (Value))	UINT32	RO	0x60p00:11, 16

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	0x00 (0_{dec})

Index 1C13 TxPDO assign

For operation on masters other than TwinCAT it must be ensured that the channels are entered in the PDO assignment (“TxPDO assign”, object 0x1C13) successively.

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2_{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	0x1A02 (6658_{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: • Bit 0 = 0: Free Run • Bit 0 = 1: Synchronous with SM 3 event • Bit 0-1 = 11: DC with SYNC1 event • Bit 15 = 1: Fast mode	UINT16	RW	0x0001 (1_{dec})
1C33:02	Cycle time	Cycle time (in ns): • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC mode: SYNC0/SYNC1 Cycle Time	UINT32	RW	0x000F4240 (1000000_{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00002710 (10000dec)
1C33:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: Free Run is supported • Bit 1: Synchron with SM 3 Event is supported (no Outputs available) • Bit 2 = 1: DC mode (SYNC0) • Bit 3 = 1: DC mode (SYNC1) • Bit 4-5 = 01: Input Shift with local event (Outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no Outputs available) • Bit 12 = 1: Legacy Synchron • Bit 13 = 1: SM event • Bit 14 = 1: dynamic times (measure by writing 1C33:08 [▶ 134]) • Bit 15 = 1: Fast mode	UINT16	RO	0x440B (17419dez)
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000927C0 (6000000_{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	0x000900B0 (590000dec)
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 1C33:03 [▶ 134], 1C33:06 [▶ 134], and 1C33:09 [▶ 134] 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_{dez})
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})

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 spacing of the objects of the individual channels	UINT16	RO	0x0010 (16_{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0009 (9_{dec})

Index F008 Code word

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

Index F009 Password protection

Index (hex)	Name	Meaning	Data type	Flags	Default
F009:0	Password protection	Password protection user calibration	UINT32	RW	0x00000000 (0_{dec})

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Maximum subindex	UINT8	RO	0x0C (12_{dec})
F010:01	Subindex 001	Analog input module channel 1	UINT32	RO	0x0000012C (300_{dec})
…	…	…	…	...	…
F010:04	Subindex 004	Analog input module channel 4	UINT32	RO	0x0000012C (300_{dec})
F010:05	Subindex 005	rese reserved	UINT32	RO	0x00000000 (0_{dec})
…	…	…	…	...	…
F010:08	Subindex 008	reserved	UINT32	RO	0x00000000 (0_{dec})
F010:09	Subindex 009	HART module channel 1	UINT32	RO	0x00001888 (6280_{dec})
…	…	…	…	...	…
F010:0C	Subindex 012	HART module channel 4	UINT32	RO	0x00001888 (6280_{dec})

5.6 Error messages and diagnosis

Interference from equipment

When operating the EPX3184 analog EtherCAT box, high frequency superimposed signals from interfering devices (e.g. proportional valves, stepper motors or DC motor output stages) can be picked up by the terminal. In order to guarantee interference-free operation, we recommend the use of separate power supply units for the EtherCAT box modules and the interference-causing devices.

6 HART

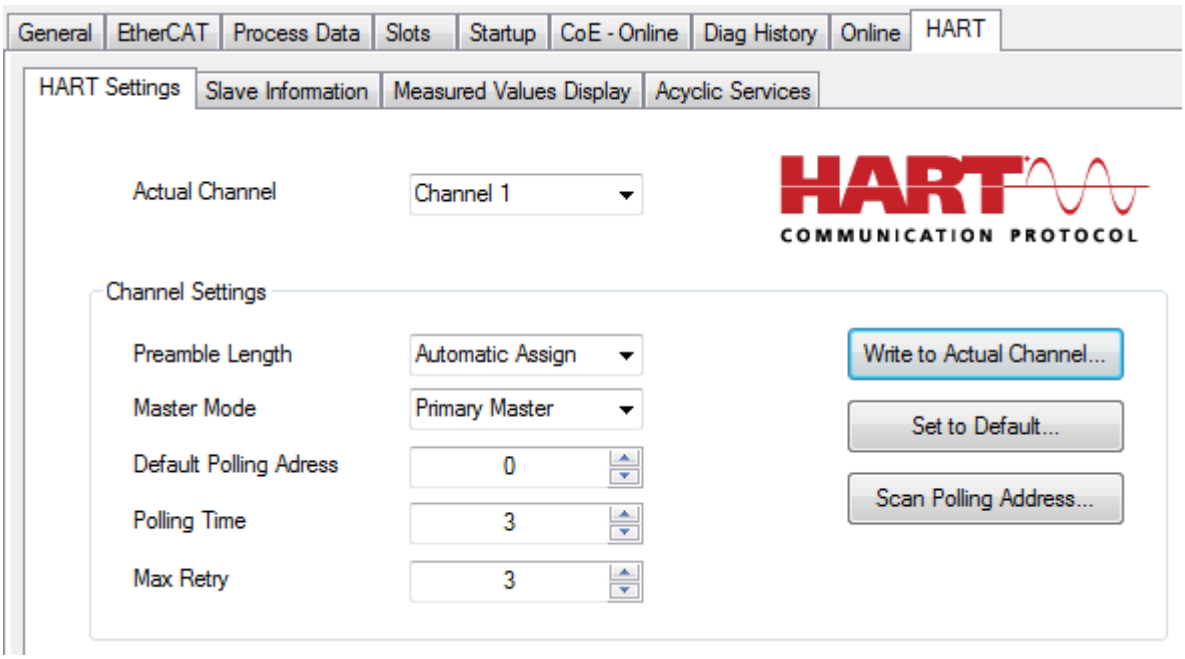
Note to HART Plug In

The HART Plug In is available since TwinCAT 3.1 Build 4022.

Please contact Beckhoff Support [▶ 148] for support in older TwinCAT Versions.

6.1 Setting

Use the "HART Settings" dialog to set the master properties.



After the corresponding channel selection, the following parameters can be changed.

- **PreambleLength** Length of the master preamble. If this is set to "Automatic Assign", the master automatically communicates with the minimum supported length of the connected HART slave.
- **MasterMode** Selection of primary or secondary master mode.
- **DefaultPollingAddress** A slave must be polled via Cmd0 in order to be able to form a unique address via the obtained information. Sets the address through which Cmd0 is polled (possible range: 0..63).
- **PollingTime** If a cyclic HART communication is enabled, this value specifies the cycle time.
- **MaxRetrys** If a HART request was not answered correctly, this indicates the maximum number of repeat cycles.

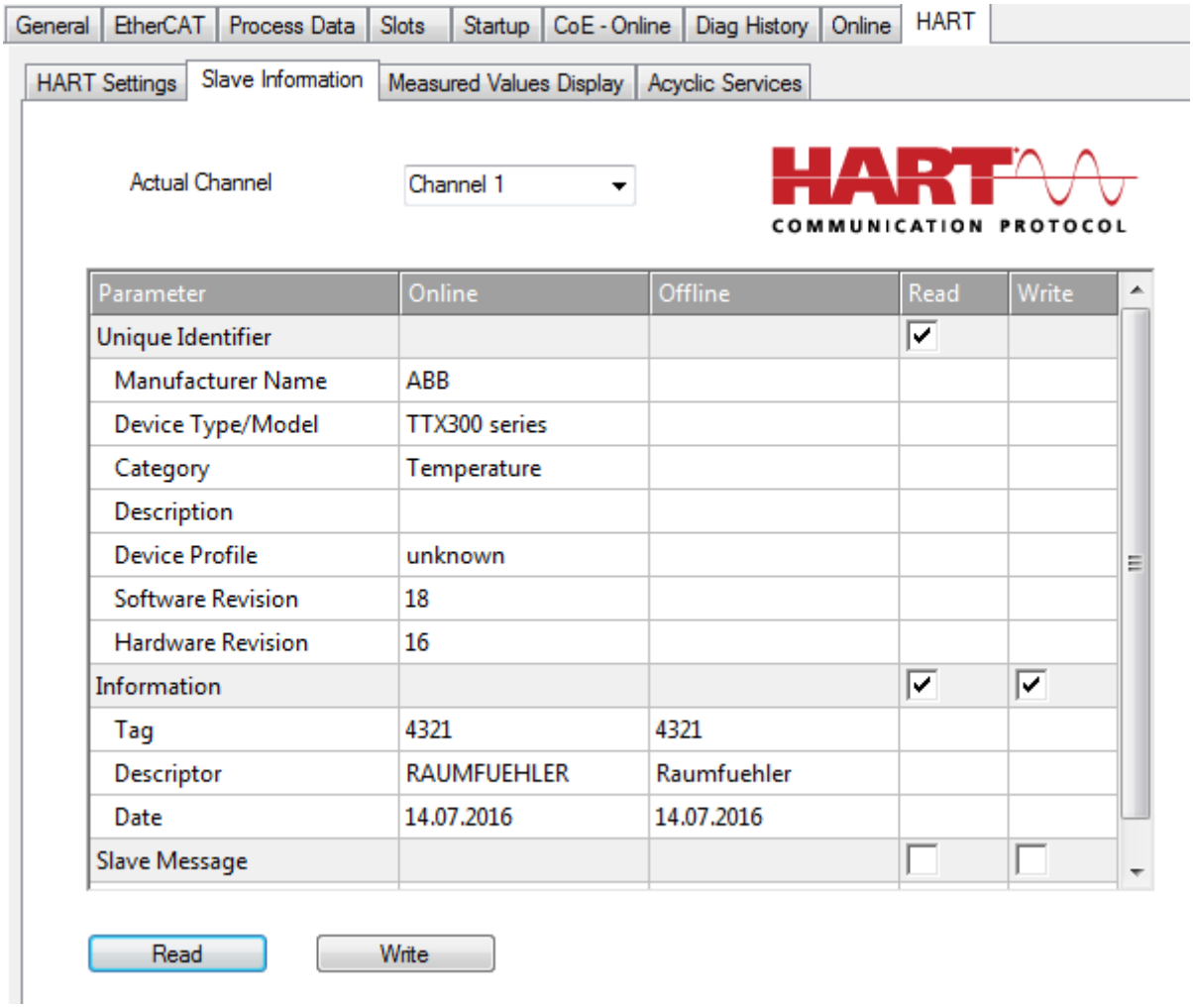
The settings have to be confirmed with the button "Write to Actual Channel..." before they become active and are saved, if required.

Use "Set to Default..." to switch back to the default values.

The button "Scan Polling Address..." enables the connected HART device to be scanned and therefore the polling address that was used to be found. The scanning range is 0 to 63. The scan is aborted once a device has responded. The address that was found can be included in the project.

6.2 Slave Information

The "Slave Information" dialog can be used to read and write general information from and to the connected slave.



The checkboxes can be used to select the data to be read and written.

The following rules apply for the data to be written:

- **Tag** 8 Bytes Packed ASCII
- **Descriptor** 16 Bytes Packed ASCII
- **Date** Format xx.yy.zzzz
- **Message** 32 Bytes Packed ASCII

Lower-case letters for packed ASCII

Lower-case letters are not allowed for packed ASCII. If lower-case letters are used, they are automatically converted to capital letters.

6.3 Measured values

Command 3 has a special function. It can be used to read the instantaneous current value via HART, plus up to 4 process values including their units. The number of returned process values depends on the HART slave used.

A graphical display of the values is available via the *Measured Values Display* tab in TwinCAT.

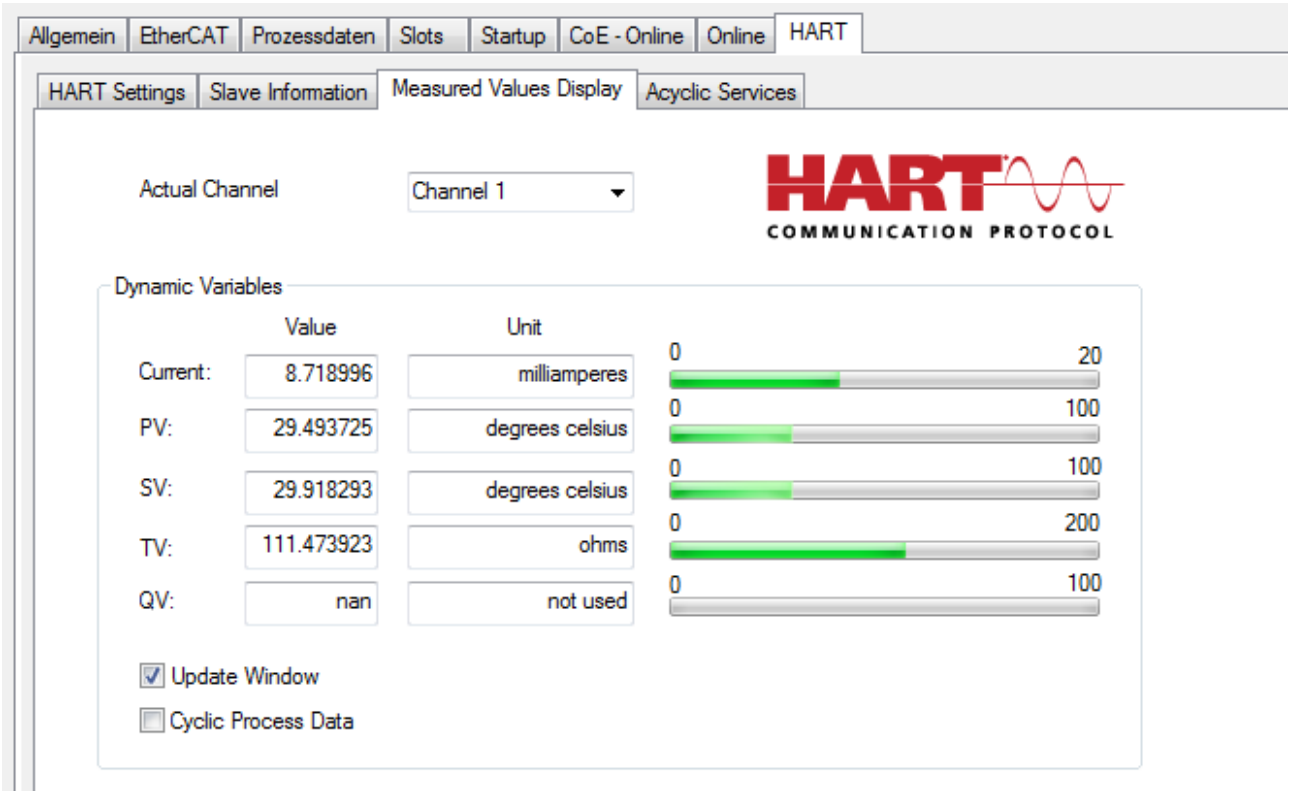


Fig. 154: Display of the measured values read by HART

The values are automatically updated every three seconds. This value is independent of the set polling time.

Cyclic process data

Furthermore, HART Cmd 3 can be mapped cyclically (see CoE object index 0x6080 [▶ 131] for channel 1 or index 0x6090 for channel 2). A HART-specific set of additional process data, which can be read by the control system, can be created by ticking the *Cyclic Process Data* checkbox.

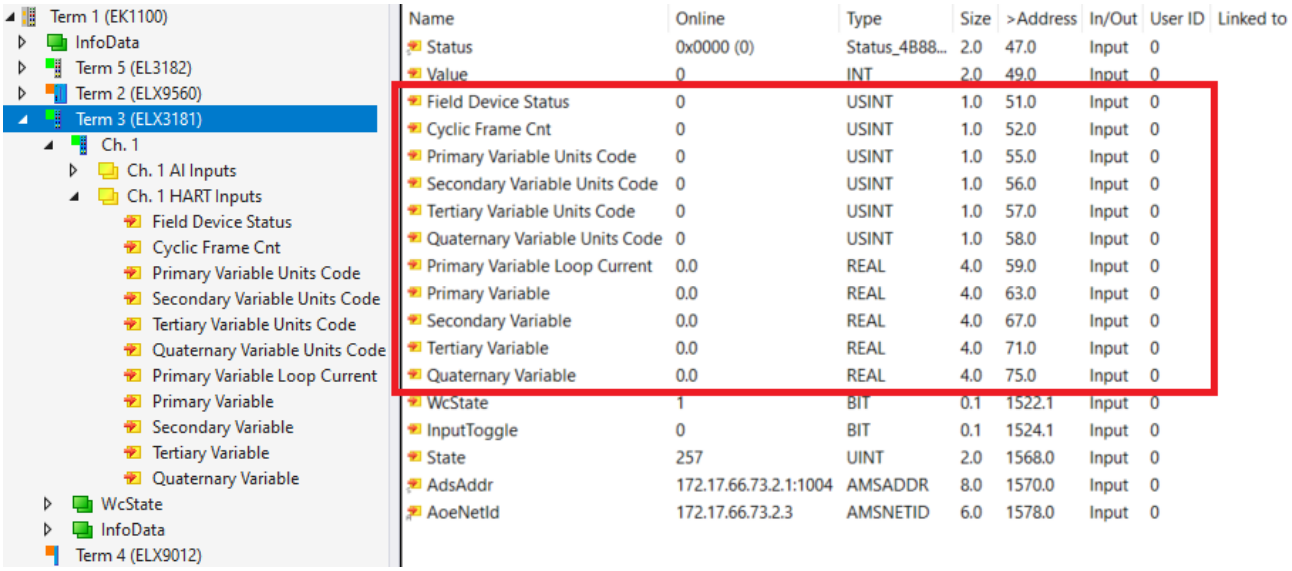
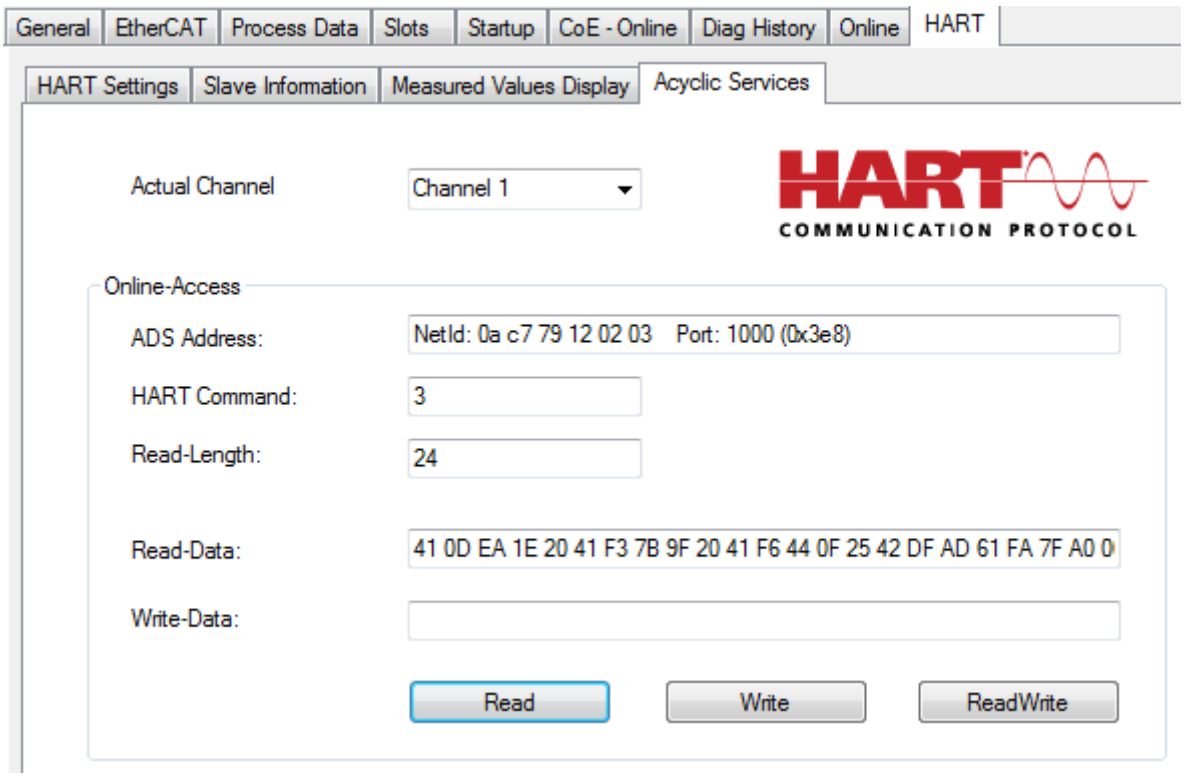


Fig. 155: Additional PDO (Cmd 3) of the HART communication

Once the configuration has been activated, it is updated cyclically with the polling time used as time base. The acyclic service from this dialog takes values from the cyclic buffer every three seconds.

6.4 Acyclic services

Simple reading or writing of individual HART commands can be initiated via the "Acyclic Services" dialog.



Reading is initiated by entering the HART command. If the slave supports the feature, the read length and the data are returned.

For a write call the command also has to be entered, plus the user data to be written (without header and CRC). The structure of the data to be used is described in the corresponding HART documentation (e.g. "Universal Command Specification").

If a HART command has user data in both directions (i.e. for reading and writing), the ReadWrite function can be used.

The commands can be issued directly via ADS. IdxGrp = 0xF302 and IdxOffs = Cmd should be used for this purpose.

7 Field Device Tool (FDT)

Note regarding the FDT plug-in

The FDT plug-in is available from TwinCAT 3.1 *build 4022*.

For older TwinCAT versions please contact Beckhoff support. [▶ 148]

A further possible use of a DTM for comprehensive sensor/ actuator communication is provided by an (external) FDT application. This is largely independent of the TwinCAT system (except for the physical layer). It is described in chapter "Using an external FDT application" [▶ 143].

7.1 Application via TwinCAT [FDT]

Once the terminal/ box has been selected in the "Solution Explorer" (available from TwinCAT 3.1, previously: System Manager in TwinCAT 2.11) the usual tabs are available; in addition [FDT]:

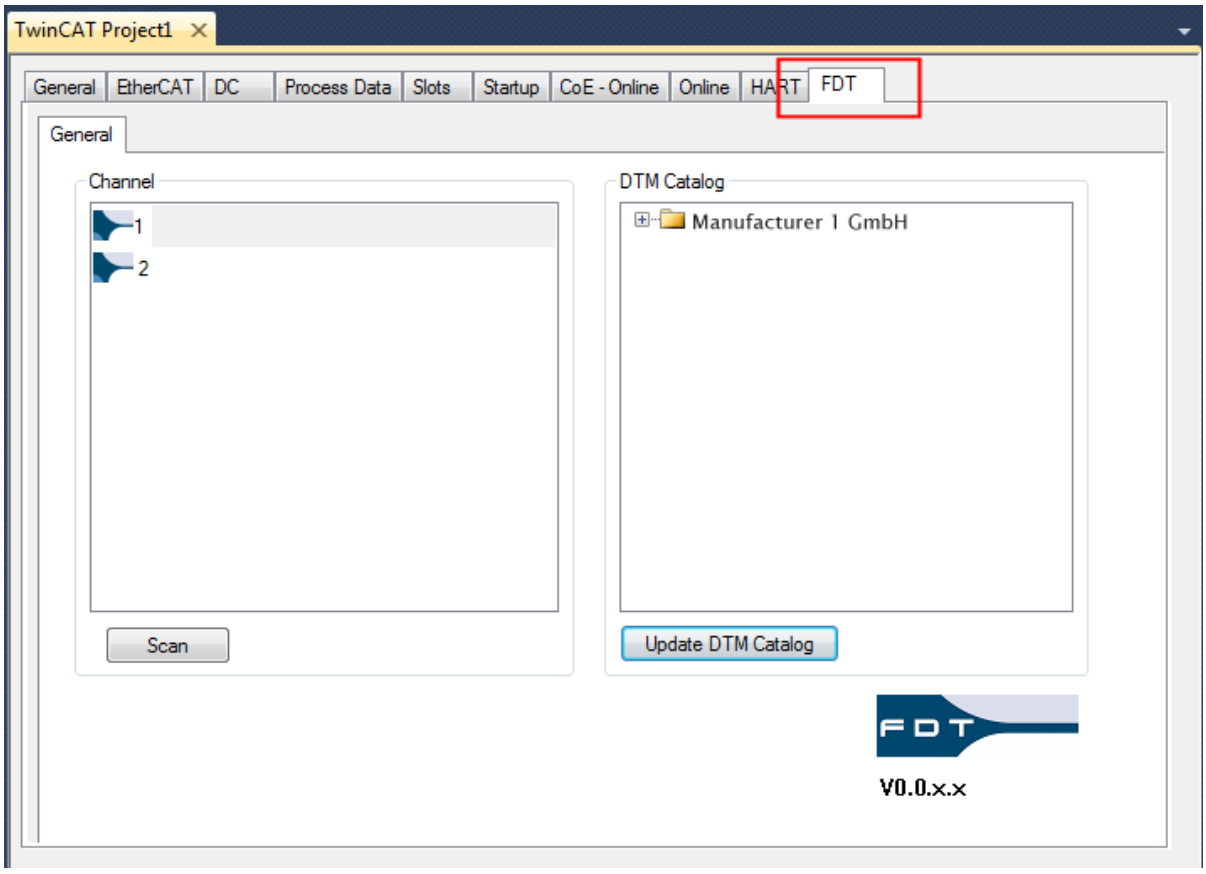


Fig. 156: FDT tab based on the example of an EL3182 terminal

If device drivers (in some cases available from the respective device manufacturer) were installed on the PC, the option "Update DTM Catalog" can be used to call the respective installation in the form of a selection list.

The DTM catalog appears in the window on the right. It should show one or several device drivers.

Use drag & drop to select the device driver from the DTM catalog that matches the sensor or actuator connected to the respective channel and pull it onto the symbol:

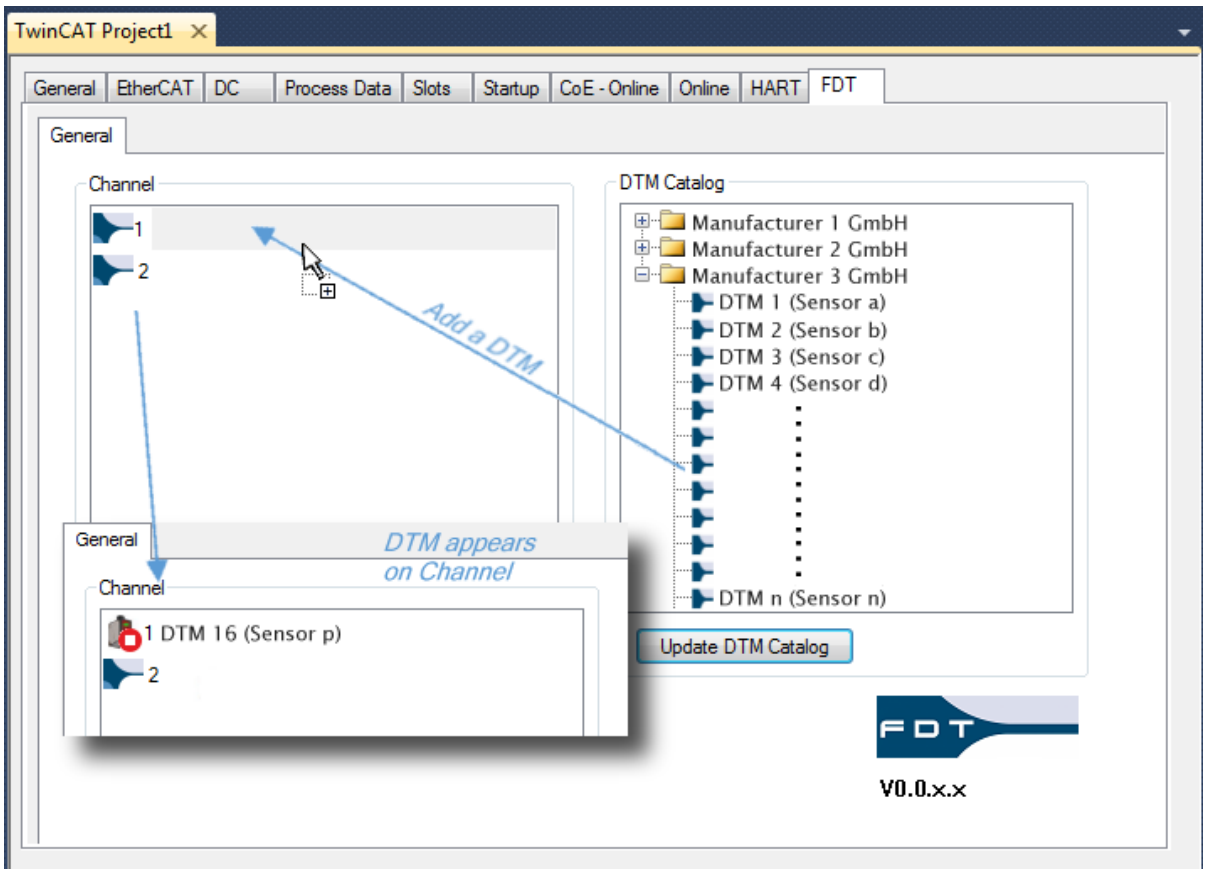


Fig. 157: Adding a sensor/actor DTM device driver to the corresponding channel of a terminal/ box

Double-click to open a further tab within [FDT] showing sensor-/actuator-specific configuration windows:

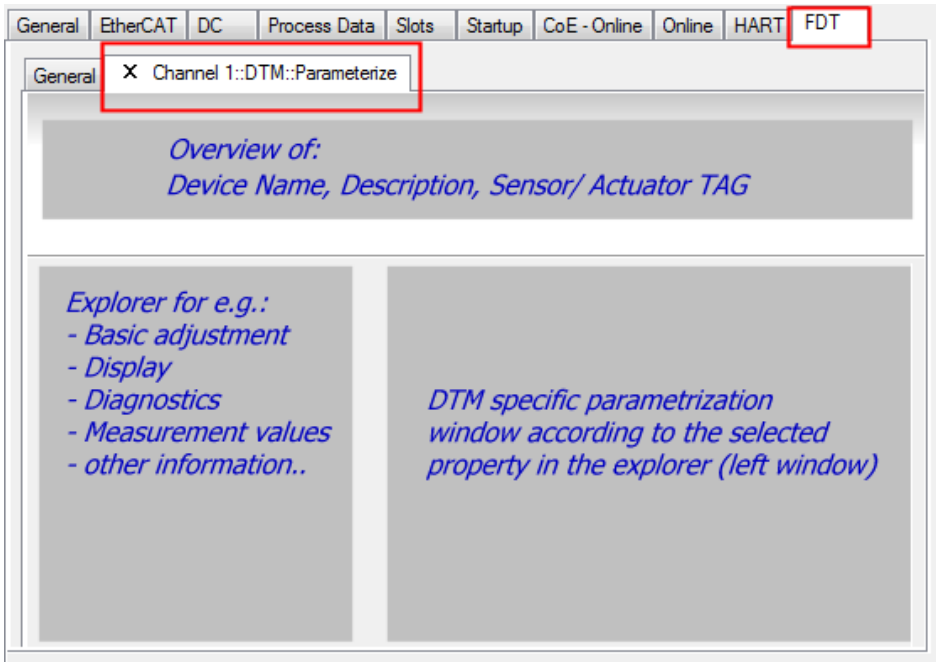


Fig. 158: General structure of a DTM configuration window within the field device tool (FDT)

Right-click on “DTM 16 (sensor p)” to open a context menu for the device used in the example:

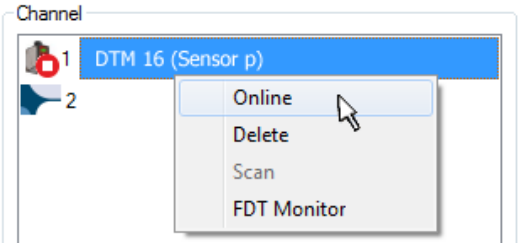


Fig. 159: Context menu of a DTM sensor/ actuator “offline”

Use “**FDT Monitor**” to open a page for log entries. This can be used to ascertain whether the DTM state machine has started up correctly, for example (debugging and service purposes).

Use “**Delete**” to remove the assignment of the DTM to the channel; “**Scan**” is described below [▶ 143].

Select “**Online**” to establish a (bus) connection with the device, which opens up additional - manufacturer-specific - functions via the context menu:

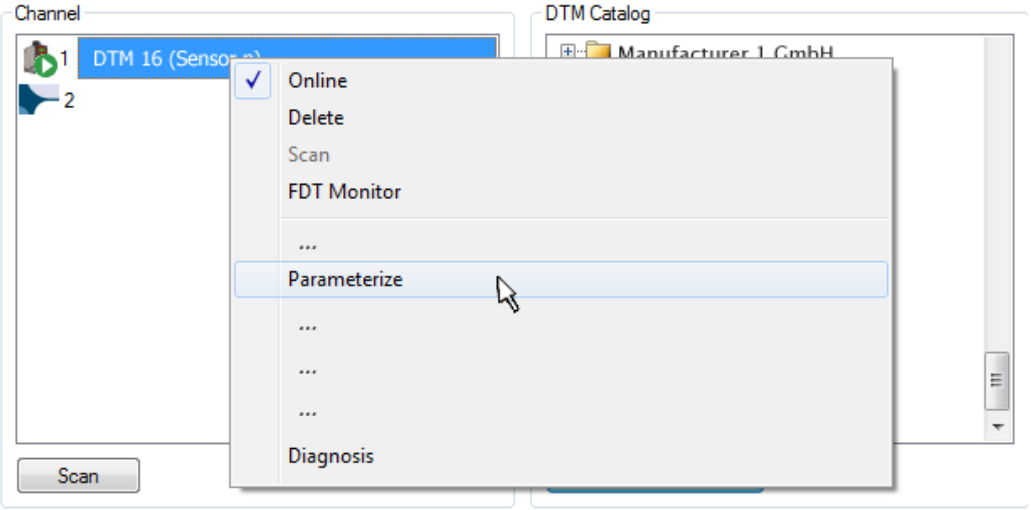


Fig. 160: Context menu of a DTM sensor/ actuator “offline”

The functions provided via the plug-in (TwinCAT) are shown at the top; the manufacturer-specific functions are shown below:

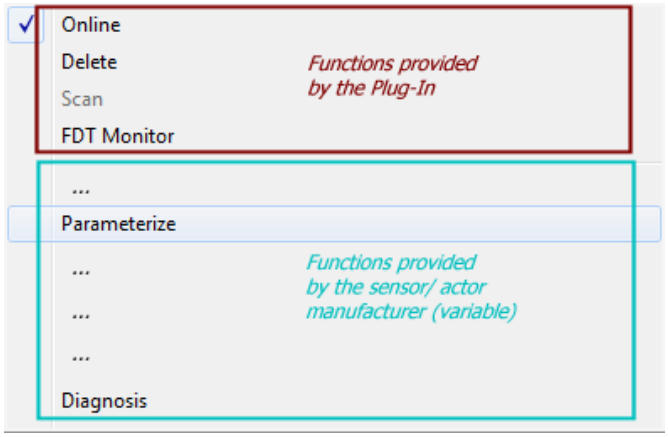


Fig. 161: Structuring of the “Online” context menu of a DTM sensor/ actuator via the FDT plug-in

Double-click on the symbol “DTM 16 (Sensor p)” as a shortcut to call a parameterization function (“Parameterize” or similar). The DTM may be in “offline” or “online” state.

Accessing a sensor/ actuator by scanning the DTM

In a configuration that is ready for operation, the DTM device can simply be added via “Scan” (button below the “Channel” window or selection in the context menu of a channel). The selection options are limited to the manufacturer-specific device drivers, which may simplify the selection of a type-conforming DTM.

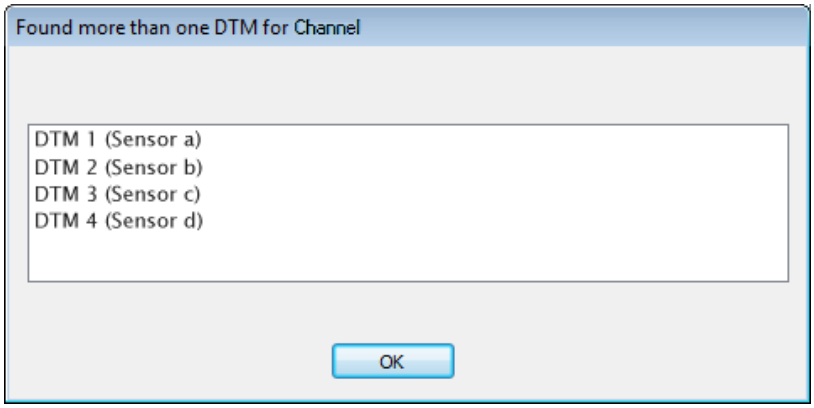


Fig. 162: Result after a DTM "scan" via the FDT plug-in in TwinCAT

7.2 Using an external FDT application

A separate FDT frame application can be used to integrate the Beckhoff ComDTM via the device catalog. A dialog with a structure that is similar to TwinCAT can then be used for establishing a bus connection (see also Startup: entering a target system).

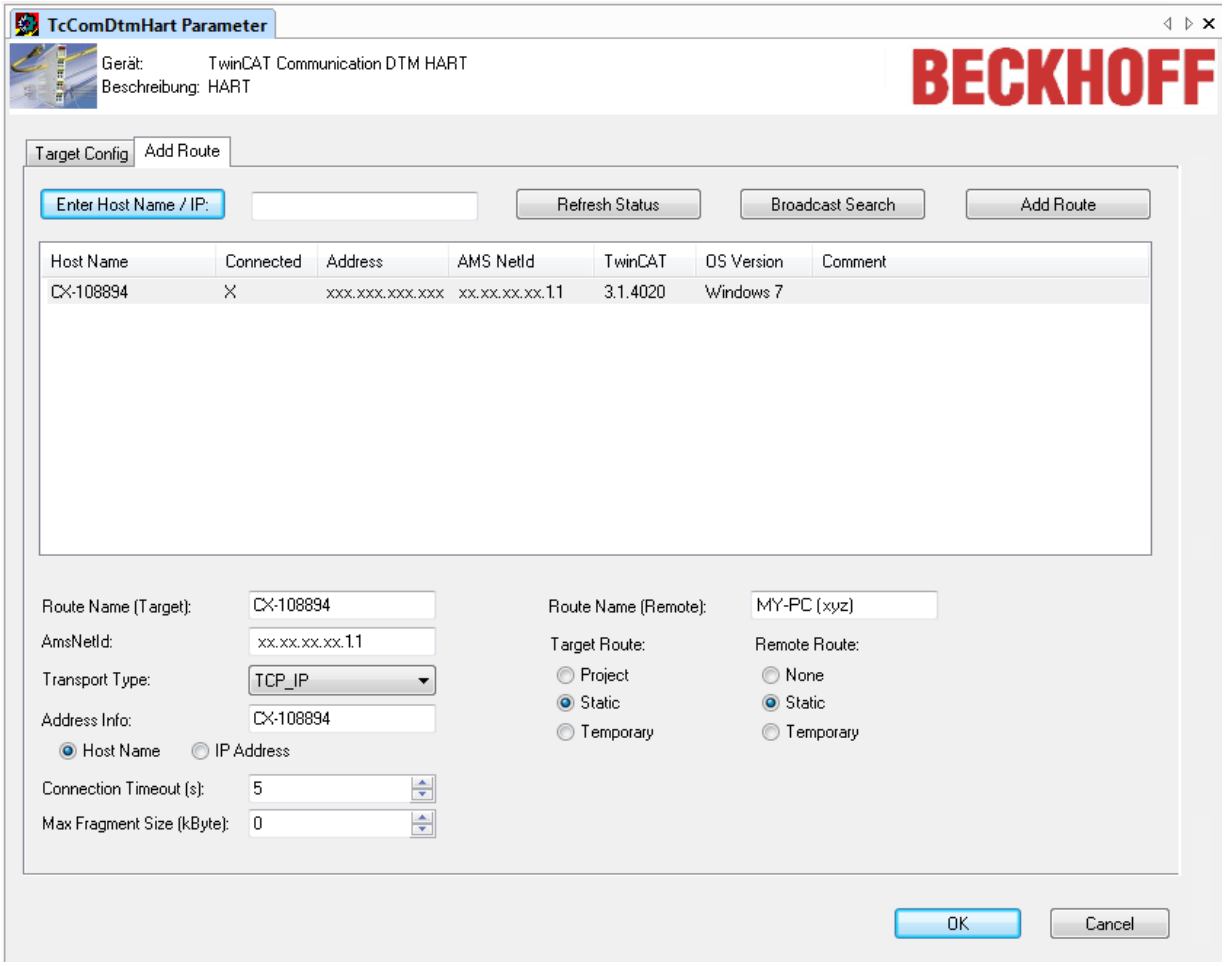


Fig. 163: AddRoute dialog in the Beckhoff DTM of the FDT application

The PLC controlling the HART terminal can therefore linked to the “Beckhoff ComDTM” either via a search or by entering the known IP address or host name, which subsequently enables access to the sensor-/actuator-specific device driver (DTM). After successful connection, all EtherCAT master systems that exist on the target system (including a local computer, for example) are automatically scanned for HART devices. Any channels that are found are then displayed accordingly.

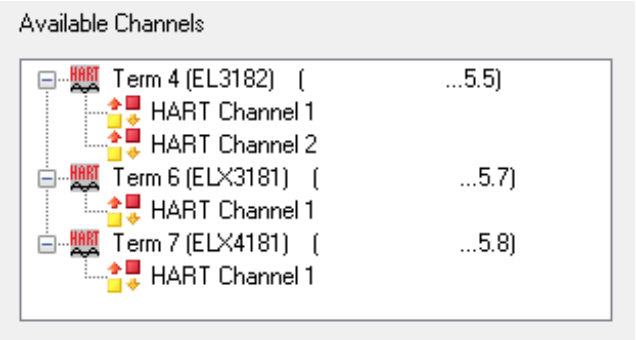


Fig. 164: Available Channels

One possible form of this application shows:

- On the left a Project Explorer with the currently configured devices
- In the center the menus for the individual objects / DTMs
- On the right the device catalog showing the DTMs available for selection

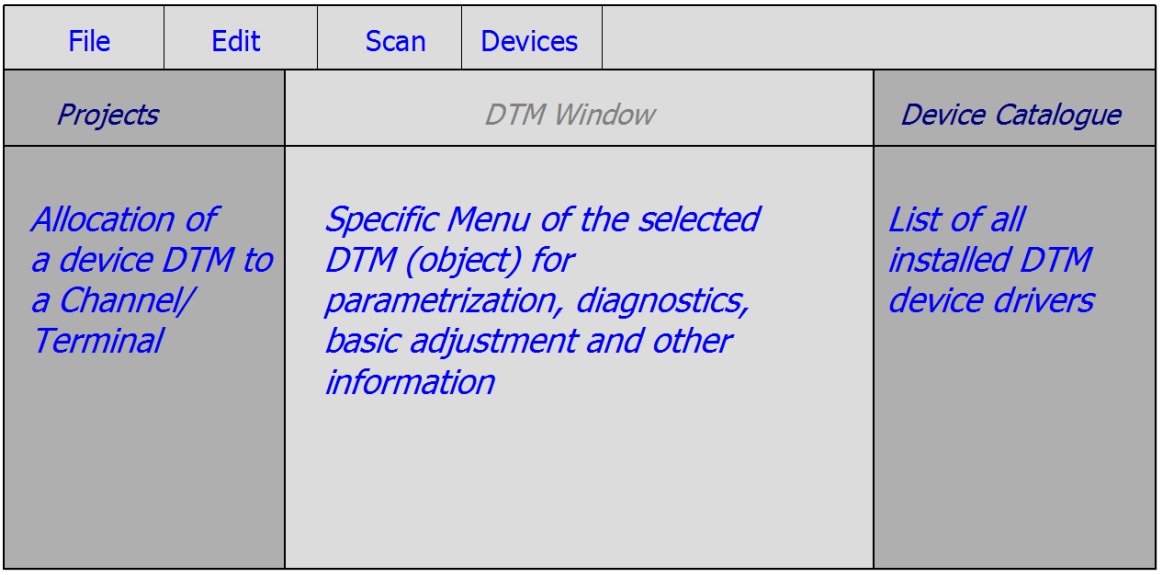


Fig. 165: Example FDT application

The device DTMs can be added in the Project Explorer under the channels (for the assignment channel → DTM). To this end an additional scan function id available, in order to limit the selection of suitable device drivers for the already connected sensors / actuators to the device-specific variants (see also: Accessing a sensor/ actuator by scanning the DTM [▶ 143]).

The following diagram shows a summary of the whole procedure:

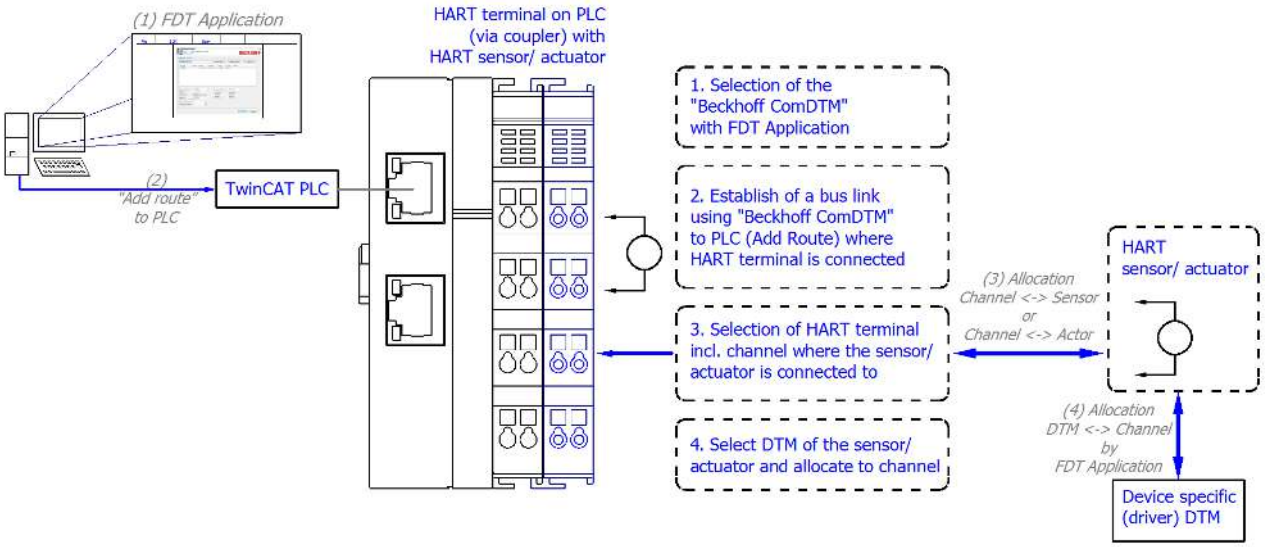


Fig. 166: Procedure with the FDT application for HART DTM access via the Beckhoff_ComDTM

8 Appendix

8.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

8.2 UL notice

Application

The modules are intended for use with Beckhoff's UL Listed EtherCAT System only.

Examination

For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).

For devices with Ethernet connectors

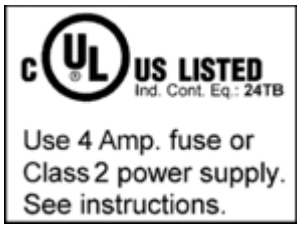
Not for connection to telecommunication circuits.

Basic principles

Two UL certificates are met in the Beckhoff EtherCAT product range, depending upon the components:

1. UL certification according to UL508. Devices with this kind of certification are marked by this sign:

2. UL certification according to UL508 with limited power consumption. The current consumed by the device is limited to a max. possible current consumption of 4 A. Devices with this kind of certification are marked by this sign:



Almost all current EtherCAT products (as at 2010/05) are UL certified without restrictions.

Application

If *restricted* certified devices are used, the current consumption at 24 V_{DC} must be limited accordingly by supplying

- from an isolated source protected by a fuse of max. 4 A (according to UL248) or
- from a voltage supply complying with *NEC class 2*.
  An *NEC class 2* voltage source must not be connected in series or parallel with another *NEC class 2* voltage source!

These requirements apply to the supply of all EtherCAT bus couplers, power supply terminals, bus terminals and their power contacts.

8.3 FM notice

Special notice regarding ANSI/ISA Ex

⚠ WARNING

Observe the permissible range of application!

The I/O modules of the ELX series may only be used in potentially explosive areas of Class I, Division 2, Group A, B, C, D or in non-explosive areas!

⚠ WARNING

Consider the *Control Drawing ELX* documentation!

When installing the I/O modules of the ELX series, be sure to read the *Control Drawing ELX* documentation, which is available in the download area of your ELX terminal on https://www.beckhoff.com/ELXxxxx!

8.4 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

Table of figures

More Information:
www.beckhoff.com/ELXxxxx

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