BECKHOFF New Automation Technology

Documentation | EN

ELM2xxx

Digital Output Terminals

2023-02-13 | Version: 1.3

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

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

Copyright

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

⚠ DANGER

Serious risk of injury!

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

⚠ WARNING

Risk of injury!

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

⚠ CAUTION

Personal injuries!

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

NOTE

Damage to environment/equipment or data loss

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

Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation Issue Status

Version	Comment
1.3	• Subchapter “Commissioning on EtherCAT Master“/ “General notes for setting the watchdog“ and “CoE Interface“ updated • Subchapter “Mounting and Wiring“/ “Accessories“/ “Shield connection ZS9100-0002“ updated • Changes within chapter Commissioning: ◦ Subchapter “Metrology and EtherCAT terminals - basic concepts“ deleted ◦ Subchapter “Continuative documentation for I/O components with analog in and outputs“ added

1.4 Version identification of EtherCAT devices

1.4.1 General notes on marking

Designation

A Beckhoff EtherCAT device has a 14-digit designation, made up of

- family key
- type
- version
- revision

Example	Family	Type	Version	Revision
EL3314-0000-0016	EL terminal (12 mm, non-pluggable connection level)	3314 (4-channel thermocouple terminal)	0000 (basic type)	0016
ES3602-0010-0017	ES terminal (12 mm, pluggable connection level)	3602 (2-channel voltage measurement)	0010 (high-precision version)	0017
CU2008-0000-0000	CU device	2008 (8-port fast ethernet switch)	0000 (basic type)	0000

Notes

- The elements mentioned above result in the **technical designation**. EL3314-0000-0016 is used in the example below.
- EL3314-0000 is the order identifier, in the case of “-0000” usually abbreviated to EL3314. “-0016” is the EtherCAT revision.
- The **order identifier** is made up of
  - family key (EL, EP, CU, ES, KL, CX, etc.)
  - type (3314)
  - version (-0000)
- The **revision** -0016 shows the technical progress, such as the extension of features with regard to the EtherCAT communication, and is managed by Beckhoff.
  In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation.
  Associated and synonymous with each revision there is usually a description (ESI, EtherCAT Slave Information) in the form of an XML file, which is available for download from the Beckhoff web site.
  From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *“EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)”*.
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

1.4.2 Version identification of EL terminals

The serial number/ data code for Beckhoff IO devices is usually the 8-digit number printed on the device or on a sticker. The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

Structure of the serial number: **KK YY FF HH**	Example with serial number 12 06 3A 02:
KK - week of production (CW, calendar week)	12 - production week 12
YY - year of production	06 - production year 2006
FF - firmware version	3A - firmware version 3A
HH - hardware version	02 - hardware version 02

Fig. 1: EL2872 with revision 0022 and serial number 01200815

1.4.3 Version identification of ELM terminals

The serial number/ data code for Beckhoff IO devices is usually the 8-digit number printed on the device or on a sticker. The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

Structure of the serial number: **KK YY FF HH**	Example with serial number 12 06 3A 02:
KK - week of production (CW, calendar week)	12 - production week 12
YY - year of production	06 - production year 2006
FF - firmware version	3A - firmware version 3A
HH - hardware version	02 - hardware version 02

Fig. 2: ELM3002-0000 with BTN 0000wwww and unique serial number 09200506

1.4.4 Beckhoff Identification Code (BIC)

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

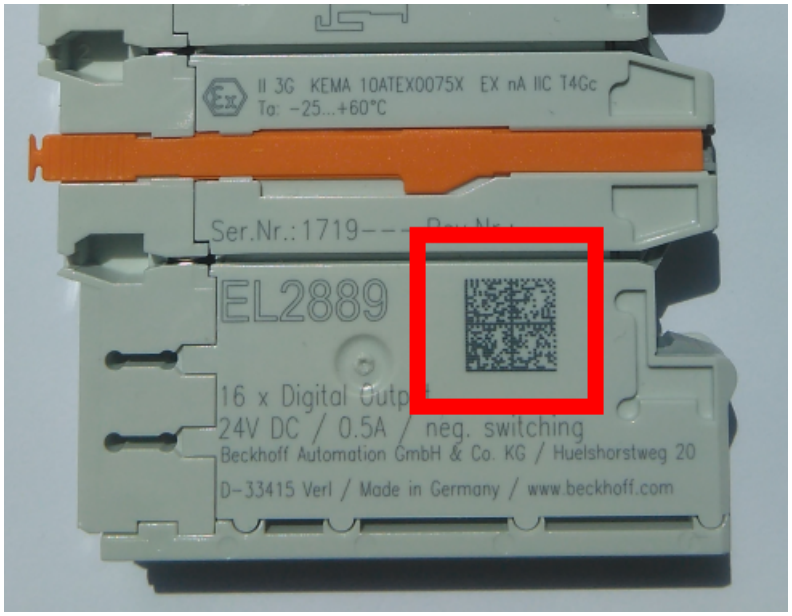


Fig. 3: BIC as data matrix code (DMC, code scheme ECC200)

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

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

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

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

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

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	SBTN	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

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

Structure of the BIC

Example of composite information from positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

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

Accordingly as DMC:

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

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTE

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

1.4.5 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

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

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

K-bus devices (IP20, IP67)

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

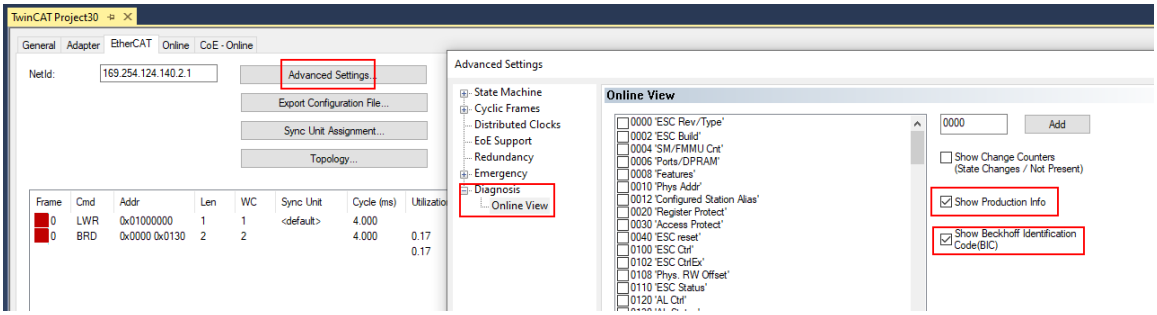
EtherCAT devices (IP20, IP67)

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

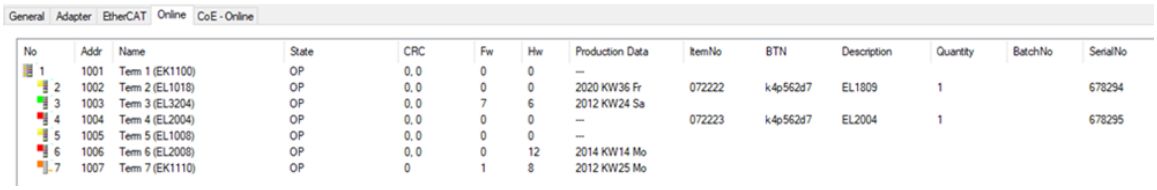
The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

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

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



  - The BTN and its contents are then displayed:



No	Addr	Name	State	CRC	Fw	Hw	Production Data	ItemNo	BTN	Description	Quantity	BatchNo	SerialNo
1	1001	Term 1 (EK1100)	OP	0, 0	0	0	--						
2	1002	Term 2 (EL1018)	OP	0, 0	0	0	2020 KW36 Fr	072222	k4p562d7	EL1809	1		678294
3	1003	Term 3 (EL3204)	OP	0, 0	7	6	2012 KW24 Sa						
4	1004	Term 4 (EL2004)	OP	0, 0	0	0	--	072223	k4p562d7	EL2004	1		678295
5	1005	Term 5 (EL1008)	OP	0, 0	0	0	--						
6	1006	Term 6 (EL2008)	OP	0, 0	0	12	2014 KW14 Mo						
7	1007	Term 7 (EK1110)	OP	0	1	8	2012 KW25 Mo						

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

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

  - the object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.

  - From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC and further eBIC auxiliary functions are available in the Tc2_EtherCAT Library from v3.3.19.0.
- Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
  The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored with the help of a category according to ETG.2010. ID 03 indicates to all EtherCAT masters that they must not overwrite these data in case of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see there. This results in a memory requirement of approx. 50..200 bytes in the EEPROM.
- Special cases
  - If multiple, hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC Information.
  - If multiple, non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC Information.
  - If the device consists of several sub-devices with their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

Profibus/Profinet/DeviceNet... Devices

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

2 Product overview

2.1 Overview multiplex terminals

Module	Signal connection	Number of outputs	Output current per channel AC/DC	Diagnostics	Distributed Clocks
EL2642	2 wire reed output, cageclamp	8	0.5 A	-	yes
ELM2642-0000	2 wire reed output, push-in cageclamp, service plug, 6-pin	8	0.5 A	-	yes
ELM2644-0000	2 wire reed output, push-in cageclamp, service plug, 6-pin	16	0.5 A	-	yes
ELM2742-0000	MOSFET output, push-in cageclamp, service plug, 6-pin	8	1 A	-	yes
ELM2744-0000	MOSFET output, push-in cageclamp, service plug, 6-pin	16	1 A	-	yes

2.2 EL2642

2.2.1 EL2642 - Introduction

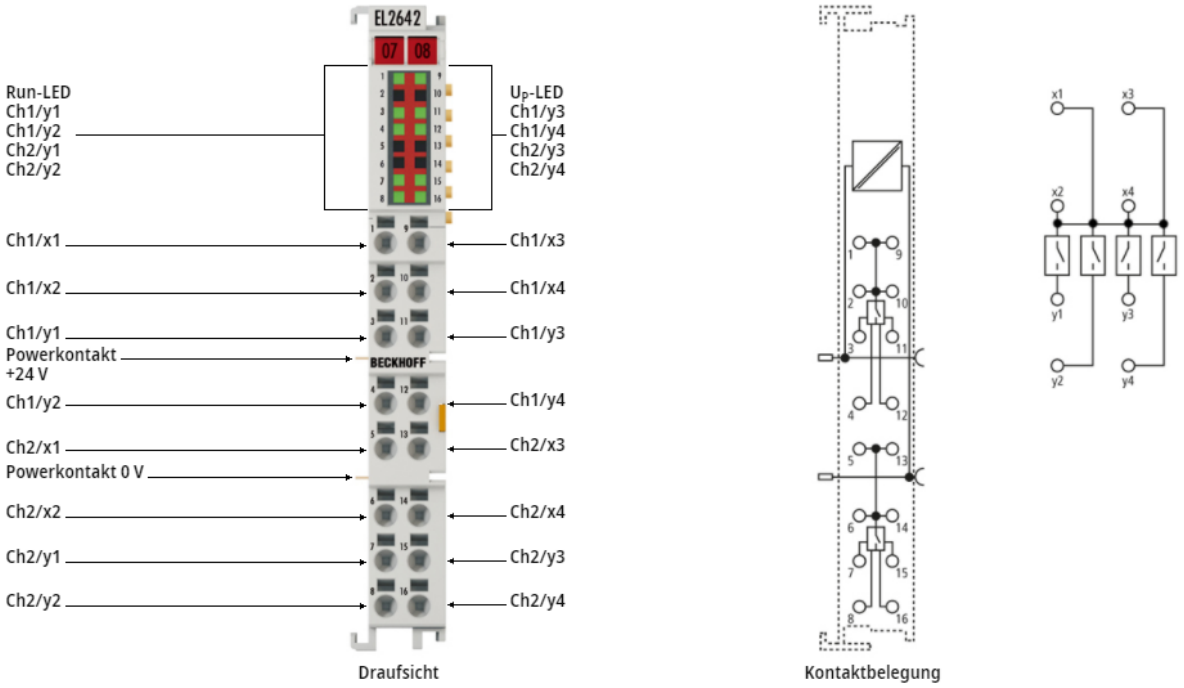


Fig. 5: EL2642

2-channel reed output terminal, multiplexer, 48 V AC/DC, 0.5 A (∑ 2 A), potential-free, 1 x 4

The EL2642 is equipped with reed relays as switching elements that allow fast switching cycles with a long service life. The absolute electrical isolation and the constant contact resistance of the reed switch allow the measurement even of sensitive signals up to fast scanning operation.

Quick-Links

- EtherCAT basics
- Mounting and wiring [▶ 120]
- Process data overview [▶ 31]
- LEDs and connection [▶ 140]
- CoE overview [▶ 33]

2.2.2 EL2642 - Technical data

Technical data	EL2642
Technology	Multiplexer
Connection technology	2-wire reed output
Connection type	Cage clamp
Number of outputs	2 x 4x multiplexer
Nominal voltage	48 V AC/DC (max. switching voltage)
Load type	ohmic
Distributed clocks	yes
Minimal EtherCAT cycle time	100 µs
Ohmic switching current	0.5 A AC/DC per channel (max. total current 1 A under UL-conditions, 2 A otherwise)
Inductive switching current	no data
Switching capacity	10 W max.
Continuous power	10 W max.
Peak current	1 A (100 ms)
Current consumption E-bus	typ. 60 mA
Electrical isolation channel/channel [1)]	Functional insulation, 100 V AC/DC (type test)
Electrical isolation channel/E-Bus [1)]	Functional insulation, 707 V DC (type test)
Electrical isolation channel/SGND [1)]	Functional insulation, 100 V AC/DC (type test)
Current consumption power contacts	20 mA typ.
Configuration	no address or configuration setting
On-resistance	0.2 Ω typ.
Switching on speed [2)]	1 ms typ., max. 5 ms
Switching off speed [2)]	1 ms typ., max. 5 ms
Breakdown voltage	170 V
Short-circuit current	not short-circuit proof
Electrical life span	Load-dependent; reference value for 10V/ 100mA ohmic: $10 \cdot 10^6$ switching cycles
Special features	fast reed relay
Weight	approx. 100 g
Permissible ambient temperature range during operation	-0…+55 °C
Permissible ambient temperature range during storage	-25…+85 °C
Approvals/markings (*	CE, UKCA, EAC

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

[1)] see notes to potential groups within chapter "Mounting and Wiring" / "Power supply, potential groups" [▶ 134]

[2)] it is not so much the Ton/Toff time of the switching element that is decisive with regard to the switching speed, but rather the time it takes for the device's internal firmware to control the switches.

2.3 ELM264x

2.3.1 ELM264x - Introduction

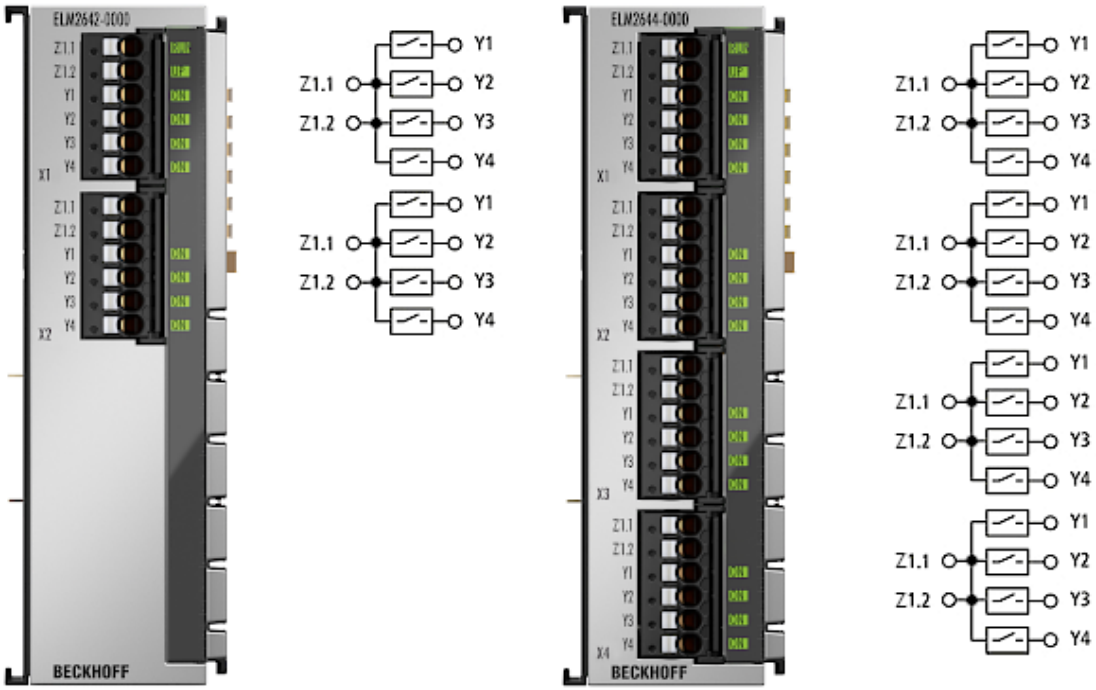


Fig. 6: ELM2642-0000, ELM2644-0000

2 and 4-channel multiplexer, 1 x 4 reed relay 48 V AC/DC, 0.5 A, potential-free make contact

The ELM2642 and ELM2644 are equipped with reed relays as switching elements that allow fast switching cycles with a long service life. The absolute electrical isolation and the constant contact resistance of the reed switch allow the measurement even of sensitive signals up to fast scanning operation.

Quick-Links

- EtherCAT basics
- Mounting and wiring [▶ 120]
- Process data overview [▶ 31]
- LEDs and connection [▶ 142]
- CoE overview [▶ 33]

2.3.2 ELM264x - Technical data

Technical data	ELM2642	ELM2644
Technology	Multiplexer	
Connection technology	2-wire reed output	
Number of outputs	2	4
Connection type	6-pin push-in cageclamp, service plug	
Number of outputs	2 x multiplexer 1 on 4	4 x multiplexer 1 on 4
Nominal voltage	48 V AC/DC (max. switching voltage)	
Load type	ohmic	
Distributed clocks	yes	
Minimal EtherCAT cycle time	100 µs	
Ohmic switching current	0.5 A AC/DC per channel	
Inductive switching current	no data	
Switching capacity	max. 10 W	
Continuous power	max. 10 W	
Current consumption E-bus	typ. 60 mA	
Electrical isolation channel/channel [1)]	Functional insulation, 100 V AC/DC (type test)	
Electrical isolation channel/E-Bus [1)]	Functional insulation, 707 V DC (type test)	
Electrical isolation channel/SGND [1)]	Functional insulation, 100 V AC/DC (type test)	
Current consumption power contacts	20 mA typ.	30 mA typ.
Configuration	no address or configuration setting	
On-resistance	0.2 Ω typ.	
Switching on speed [2)]	< 1 ms typ.	1 ms typ., max. 5 ms
Switching off speed [2)]	< 1 ms typ.	1 ms typ., max. 5 ms
Breakdown voltage	170 V	
Short-circuit current	not short-circuit proof, see Peak current	
Electrical life span	Load-dependent; reference value for 10V/100mA ohmic: $10 \cdot 10^6$ switching cycles	
Peak current	1 A (100 ms)	
Thermal dissipation	1 W typ.	2 W typ.
Special features	fast reed relay	
Weight	approx. 350 g	
Permissible ambient temperature range during operation	-0…+55 °C	
Permissible ambient temperature range during storage	-25…+85 °C	
Approvals/markings (*	CE, UKCA, EAC	

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

[1)] see notes to potential groups within chapter "Mounting and Wiring" / "Power supply, potential groups" [▶ 134]

[2)] it is not so much the Ton/Toff time of the switching element that is decisive with regard to the switching speed, but rather the time it takes for the device's internal firmware to control the switches.

2.4 ELM274x

2.4.1 ELM274x - Introduction

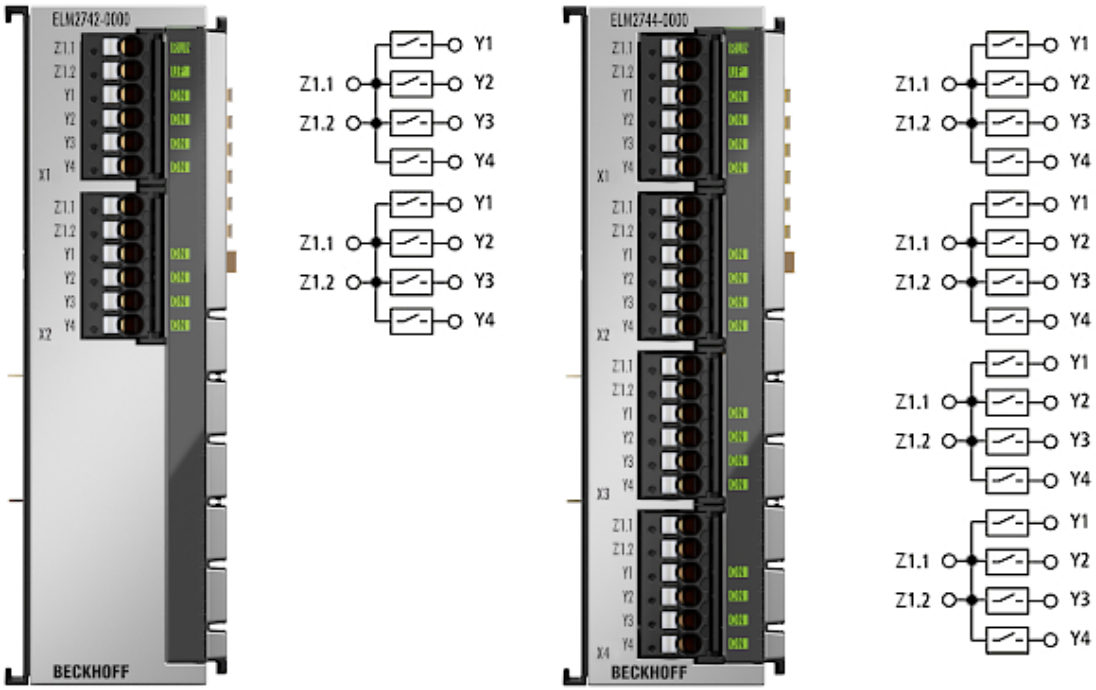


Fig. 7: ELM2742-0000, ELM2744-0000

2 and 4-channel multiplexer, 1 x 4 solid state relays 48 V AC/DC, 1 A, potential-free make contacts

The ELM2742 and ELM2744 are equipped with wear-free solid state relays (semiconductor switches) as switching elements, which are especially designed for small analog signals and thus stand out from solid-state terminals for general applications such as the EL279x. Due to the very low leakage current in the open state, the signals applied in parallel hardly affect one another

Quick-Links

- EtherCAT basics
- Mounting and wiring [▶ 120]
- Process data overview [▶ 31]
- LEDs and connection [▶ 142]
- CoE overview [▶ 33]

2.4.2 ELM274x - Technical data

Technical data	ELM2742	ELM2744
Technology	Multiplexer	
Connection technology	MOSFET output	
Number of outputs	2	4
Connection type	6-pin push-in cageclamp, service plug	
Number of outputs	2 x multiplexer 1 on 4	4 x multiplexer 1 on 4
Nominal voltage	48 V AC/DC (max. switching voltage)	
Load type	ohmic	
Distributed clocks	yes	
Minimal EtherCAT cycle time	100 µs	
Ohmic switching current	1 A AC/DC per channel (in the operating temperature range, 2 A at 25 °C ambient temperature)	
Inductive switching current	no data	
Leakage current off state	typ. 10 nA (reference value, dependent by ambient temperature and operationg voltage)	
Current consumption E-bus	typ. 50 mA	
Electrical isolation channel/channel [1)]	Functional insulation, 100 V AC/DC (type test)	
Electrical isolation channel/E-Bus [1)]	Functional insulation, 707 V DC (type test)	
Electrical isolation channel/SGND [1)]	Functional insulation, 100 V AC/DC (type test)	
Current consumption power contacts	20 mA typ.	30 mA typ.
Configuration	no address or configuration setting	
On-resistance	0.1 Ω typ. (increasing with ambient temperature)	
Switching on speed [2)]	1 ms typ., max. 5 ms	
Switching off speed [2)]	1 ms typ., max. 5 ms	
Breakdown voltage	80 V	
Short-circuit current	not short-circuit proof, *see Peak current*	
Peak current	5 A (100 ms)	
Special features	wear-free solid-state relay (semiconductor)	
Weight	approx. 350 g	
Permissible ambient temperature range during operation	-0…+55 °C	
Permissible ambient temperature range during storage	-25…+85 °C	
Approvals/markings (*	CE, UKCA, EAC	

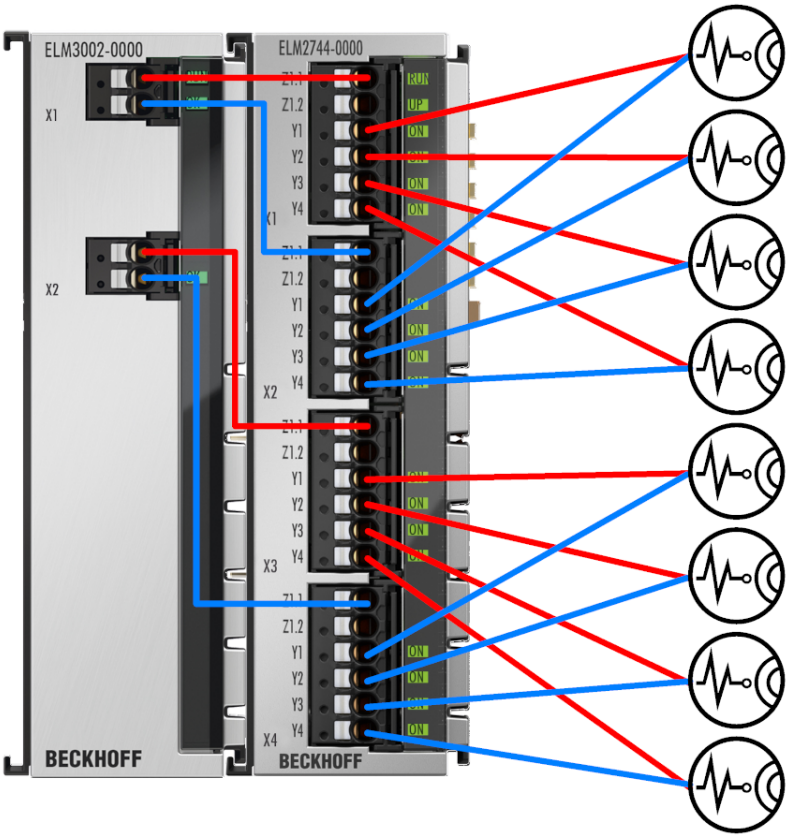
*) Real applicable approvals/markings see type plate on the side (product marking).

[1)] see notes to potential groups within chapter "Mounting and Wiring" / "Power supply, potential groups" [▶ 134]

[2)] it is not so much the Ton/Toff time of the switching element that is decisive with regard to the switching speed, but rather the time it takes for the device's internal firmware to control the switches.

3 Product description

With EtherCAT switching output terminals with various switching technologies, digital or analog signals can be simply distributed through the opening and closing of electrical connection in a system-integrated manner. This method is sometimes used when several sensor signals are to be evaluated by a central analog input for reasons of cost or to simplify the circuit technology, i.e. a 1:n circuit. The n sensor lines are then fed to the central analog input one after the other in a defined rhythm, the so-called multiplex method. Below, an example is shown in which eight vibration sensors are fed to two IEPE inputs:



The multiplex method can also be used on output potentials, i.e. an analog output can be distributed to several actuators in succession. In addition

- an n · m matrix structure can be realized by many switches,
- or lines can generally be temporarily disconnected, e.g. for diagnostic purposes and
- the principle can likewise be applied to digital signals (depending on the performance of the switches).

A recognizable disadvantage of this method is obvious and also determines the usage possibility: access to all sources is not continuously available. The application must allow a source to only be intermittently accessed. However, if a 1:1 connection of the signal sources with corresponding analog inputs is not necessary and the signals are given the necessary time to settle and stabilize after switching over, the multiplex method can often be the right solution in order to gain considerable space and cost advantages.

In relation to analog signals, the devices described in this documentation considerably extend the usage possibilities for standard and precision analog technology and enable:

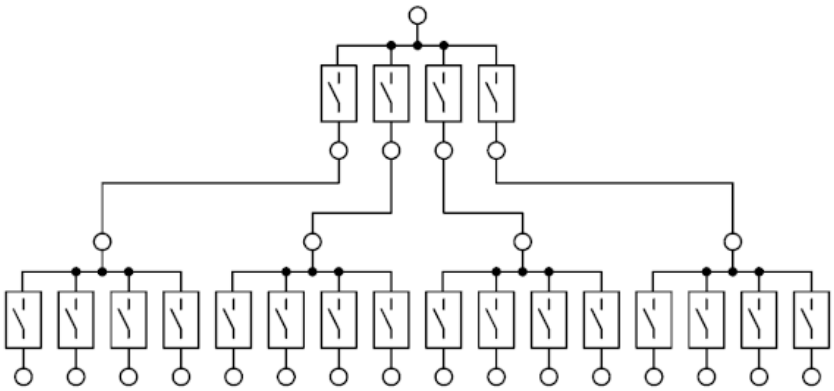
- ATE: automated test equipment,
- HIL: hardware-in-the-loop systems,
- multi-channel temperature measurement in building material development,
- multi-channel vibration monitoring (Condition Monitoring),
- resistance simulation (by parallel connection).

Depending on

1. the number of sources,
2. the number of lines to be switched (six for 6-wire full bridge anyhow) and
3. the number of analog inputs available,

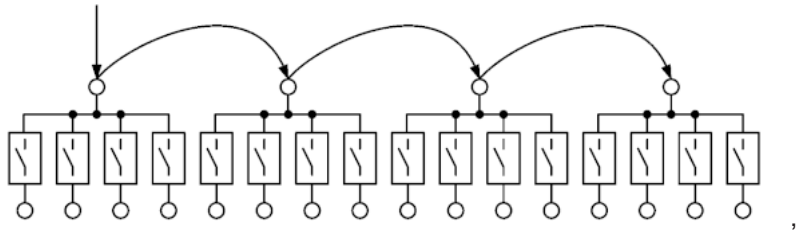
the limit from when a multiplexer solution causes less costs and space requirements can be inferred. The higher the second number of lines to be switched, the more the switching effort increases and the later the use of a multiplexer is worthwhile – with a 2 or 3-wire circuit, the use of a multiplexer is already worthwhile very early on.

Hint: the series connection of multiplexers (multiplicative use)



,

requires more switches than the pure concatenation (additive use)



,

which is possible with the devices mentioned here, since the individual switches are NO (normally open, thus de-energized open).

Overview of the properties of the terminals in this documentation

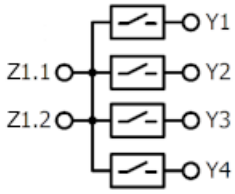
The multiplex terminals

- are integrated directly into the EtherCAT fieldbus system as EtherCAT Terminals and can be controlled,
- contain reed switches (ELM264x, EL2642) or solid state relays (semiconductor, ELM274x) as high-quality switching elements, which have been specially selected for use on "small" sensitive analog signals.
  The switching types have their individual advantages and disadvantages, see the recommendations for use.

Furthermore

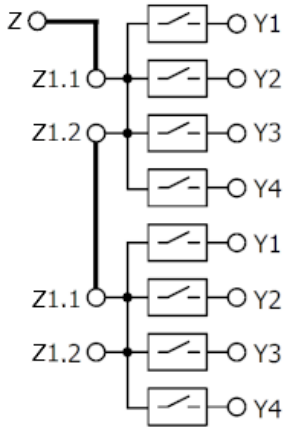
- each switching element can be switched on/off individually like a digital output, independent of the others. It is therefore a single-pole single-throw (SPST) switch,
- the switches are not self-latching or bistable; they remain closed as long, and only as long as the corresponding output signal of the controller is applied via EtherCAT.
- Each switching element is NO/normally open, i.e. "opened" in the off state.
- The switches of a terminal are switched (over) internally at the same time, in DistributedClocks operation according to the DC Sync.; this makes simultaneous switching possible across terminals,

- Four switches are always grouped into a so-called 1:4 multiplexer group: the four input points are connected and led out to two contact points Z1.1 and Z1.2 on the terminal:

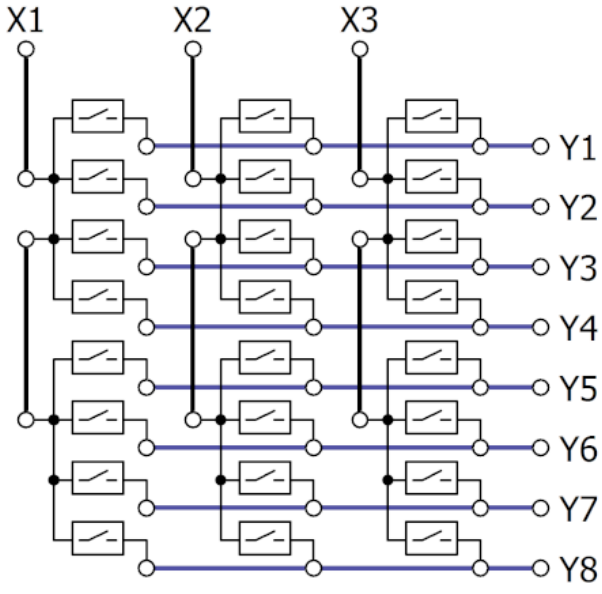


Due to the fact that the four switches are individually controllable (closable) and - if deactivated - all are open, this arrangement is strictly speaking not a pure multiplexer that always shows one (and only one) connection. Nevertheless, this term is widely used for the switching output terminals described here, as it best illustrates the typical use case.

- One terminal contains several of these multiplexers. The X1 points can be connected externally, representing chained multiplexers. It is thus easy to form a 1:8 multiplexer, etc. from two 1:4 multiplexers:



- Matrix systems can also be set up with the terminals, the corresponding interconnection must be external:



- The naming scheme of the multiplex terminals is:
  - EL/ELM26xx: EtherCAT Terminal Reed Relay,
  - EL/ELM27xx: EtherCAT Terminal Solid State (semiconductor switches),

    - EL/ELM2xab,
      - a: number of poles per multiplexer,
      - b: number of channels per terminal, corresponds to the number of multiplexers.
- The terminals require the 24 V power contact supply for self-operation.

The multi-channel acquisition and ranking of analog signals, as well as the flexible switchover of various test states in production-integrated test stations, is easily possible through the integration of the often necessary changeover switches into the EtherCAT system.

4 Commissioning

The terminals can be inserted into the terminal segment as simple digital output terminals and controlled via EtherCAT.

The multiplex function in combination with the corresponding analog/digital signals must be programmed on the application side.

Diagnostic Messages (DiagMessages)

Important messages are:

- 0x1201
  - Communication to field side re-established (Up back on?)
  - Communication to field side re-established (Up back on?)
- 0x8201
  - No communication to field side (Up missing?)
  - No communication to field side (Up missing?)

Additional functions

Counter in the CoE (image section from the CoE directory of the TwinCAT environment of the ELM2644):

Index	Name	Flags	Value	Unit
9000:0	DO Info data Group 1	RO	> 4 <	
9000:01	Y1 switch cycle counter	RO	0x00000034 (52)	
9000:02	Y2 switch cycle counter	RO	0x00000034 (52)	
9000:03	Y3 switch cycle counter	RO	0x00000034 (52)	
9000:04	Y4 switch cycle counter	RO	0x00000034 (52)	
9010:0	DO Info data Group 2	RO	> 4 <	
9020:0	DO Info data Group 3	RO	> 4 <	
9030:0	DO Info data Group 4	RO	> 4 <	
F000:0	Modular Device Profile	RO	> 2 <	
F008	Code word	RW	0x00000000 (0)	
F600:0	DO Device Status Inputs	RO	> 15 <	
F900:0	DO Device Info data	RO	> 18 <	
F900:12	Operating time	RO	0x0000007D (125)	min
FB00:0	DO Command	RO	> 3 <	

Fig. 8: Operating hour counter in minutes. A device operating minute counter can be called in 0xF900:12. It cannot be cleared.

Switching cycle counter

A switching cycle counter can be called for each switching element in 0x9000:nn. The counters count the number of contact closures and cannot be cleared.

4.1 Recommendations for use

Comparison of properties depending on switching technology

Technology	Reed	FET/Solid State	EMR, Electromagnetic Relay
Examples	ELM264x, EL264x	ELM274x, EL27xx	EL26xx
Switching characteristics	"real" switch, completely disconnects	semiconductor switches, always low leakage current	"real" switch, completely disconnects
Transition resistance	Higher	Low	Low
Repeat stability Ron	Good	Very well	Very well
Overload behavior	Sensitive	Less sensitive	Sensitive
Switching speed **)	Fast	Very fast	Medium
Dependence of properties on the ambient temperature	Low	Higher	Low
Leakage current when open	Very low	Higher, increasing with operating temperature, several nA	Very low
Resulting offset/ thermovoltage	Yes, several ±10 µV	Low, a few ±1 µV	Yes, higher than reed, also due to self-heating
Wear	Yes, gradual aging with normal use Typ. failure: non-conductive	Low, if at all, then sudden failure Typ. failure: conductive	Yes, gradual aging with normal use Typ. failure: non-conductive
AC behavior (alternating signals), crosstalk	Good	Less good, to be assessed applicatively	Good
Recommended use	Use at fluctuating ambient temperature 4-wire resistance measurements	Use at room temperature 2-wire resistance measurements Thermocouples (as long as the ambient temperature does not change significantly) Current peaks Frequent/fast switching	

[2]) it is not so much the Ton/Toff time of the switching element that is decisive with regard to the switching speed, but rather the time it takes for the device's internal firmware to control the switches.

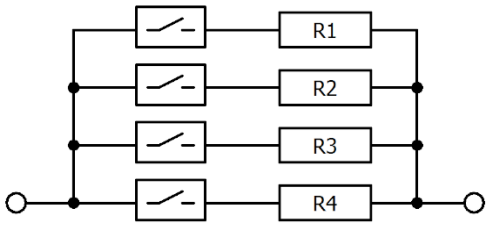
If a switching element such as the EL2xxx/ELM2xxx is used for the manipulation of demanding analog signals, it should always be borne in mind that temperature changes affect properties of all kinds and can change them adversely. This must be taken into account when designing the system.

In addition, the switching elements always bring ohmic/inductive/capacitive influences into the system. Offset matching is recommended for voltage or resistance measurement.

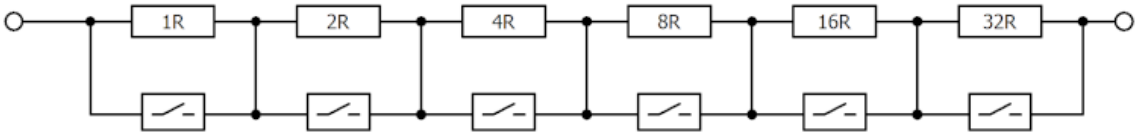
In general, "complete" all-pole switching of the signals is recommended, i.e. all six lines in the case of a 6-wire DMS connection. If this is deviated from and, for example, only the signal line (+) is switched in a 2-pin IEPE connection

- earth loop(s) may be created, depending on the design of the sensor and wiring,
- interference inductions are facilitated by an N-fold enlarged line network,
- the switched signal is to be given a sufficient settling time, e.g. a constant IEPE current signal must first stabilize for a few 100 ms,

- "bouncing" can occur when mechanical contacts close,
- the switched devices must be suitable for the on/off switching operation,
- in the case of resistance simulation, the parallel circuit

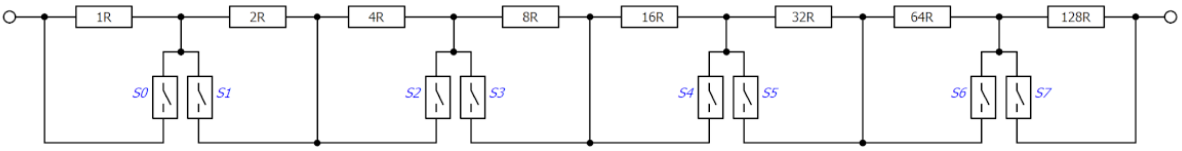


can be used.
To get nearly equal resistance gradations with as many switching combinations as possible, the serial circuit with e.g. binary stepped values is recommended:



It requires switches that are accessible on both sides, such as provided by the EL2624.

- To create a resistance simulation based on this serial circuit with the multiplex terminals is only possible with two switches of a group of 4 switches:



A combination of both circuit types leads to further possible resistance values and partially redundant switching combinations.

Like any technical device, a switching terminal can fail: on the one hand by non-switching although it is supposed to switch, e.g. "contact destroyed" and, on the other hand, by permanent connection although it is supposed to disconnect, e.g. "contact sticks". Sufficient plausibility queries should be provided for this purpose on the receiving side.

Additional notes:

- An external protective circuit is necessary with inductive/capacitive loads, e.g. a short-circuit diode.
- External strong magnetic fields or vibrations/shocks can affect the function, for example, of reed relays
- The devices discussed here have switching cycle counters in the CoE and it is recommended to observe them.
- With long/frequent use, aging switching contacts exhibit a slowly increasing resistance before complete failure. Occasional checking of the switching function can be useful.
- The repair of individual switches in the Beckhoff Service Dept. is possible.

Concrete examples of MUX applications with Beckhoff analog inputs

RTD to EL320x terminals

- Multiplexing introduces transition resistances into the sensor connection. Therefore, for the combination RTD + multiplexing, the 4-wire connection with +R(U_{Exc}), +RL(+sense), -RL(-sense) and -R(AGND) is recommended.
- The 4-wire connection is supported, for example, by the EL3201-xxxx, EL3202-xxxx, EL3751, ELM350x, ELM370x and others.
- An EL320x is only designed to supply one RTD sensor. Therefore, the +R/U_{Exc} connection cannot be used to power multiple RTD sensors. So it can't be muxed.
- -R(AGND) can be connected through for multiple RTDs, so it does not need to be muxed.
- +RL(+sense) and -RL(-sense) must be muxed as feedback lines.

- Terminals from the ELM26xx, ELM27xx or EL26xx series can be used for multiplexing.

Interference of multiplexers in the signal curve

Ideally, the sensor and the evaluation input (measuring device) are connected directly (without connectors) and via the shortest possible cable. The insertion of multiplexers (or other elements, such as connectors, extension cables, etc.) between the sensor and the evaluation input therefore fundamentally changes the *direct* signal flow from source to sink in a real setup. This is intentional when the switch is open, but also especially - rather unintentionally - when the switch is closed. The switching element, including the necessary cables and plug-in transitions, generally generate interference that changes or falsifies the measurement results. This must generally be observed when using switches in the signal flow. Depending on the type of signal, the special operation of the switches in multiplex mode also has an influence on the signal; among other things, crosstalk, attenuation and reflections may occur and settling times may have to be taken into account during switching operations.

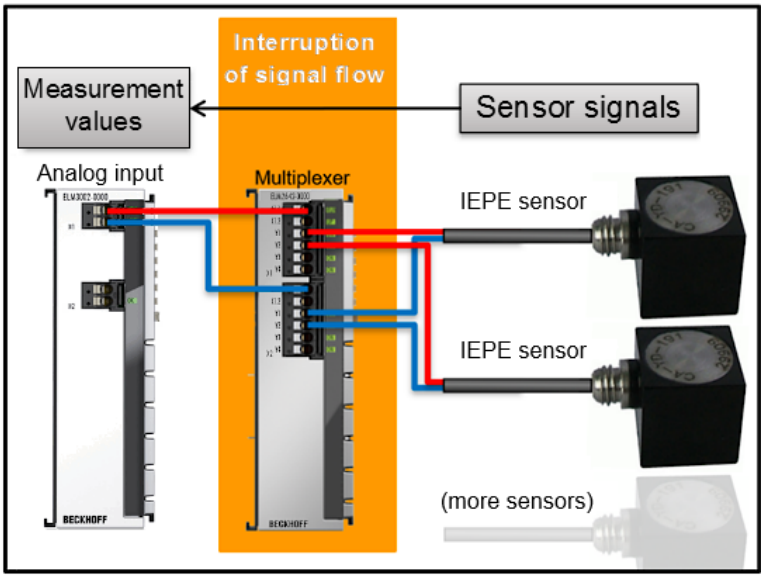


Fig. 9: Interruption of the "direct" signal flow of the sensor signals to the measurement input in the real setup

Potential influence of switches on high-frequency signals

Depending on signal frequency/frequencies and amplitude (signal strength), frequency-dependent effects may occur in the multiplexer and between the unavoidable lines. Here are some observations and notes:

- Crosstalk
  - "Real" switches such as EMR and reed separate "exemplarily"; crosstalk attenuation values of -90 dB have been observed (the larger the value the greater the signal distance between the interferer and victim channels, i.e. the better the crosstalk attenuation).
  - Semiconductor switches (FET, SolidState) have a feedback effect on the control due to their non-electrical isolation and thus crosstalk into adjacent switches, starting with a crosstalk attenuation of -80 dB at 100 Hz, at frequencies from 1 kHz up to -60 dB were observable. The values given do not represent specification values, but are intended to provide orientation for your own application-specific considerations.
- Attenuation
  - Attenuations of less than -0.02 dB were determined for each switch in the range 0...1 kHz, corresponding to approx. 0.23 % (2300 ppm) amplitude.
  - Semiconductor switches tend to have somewhat lower attenuation.
- Reflections
  - no observations available yet.

Potential influence of switches on IEPE signals

IEPE sensors generate an offset voltage (bias) of approx. 10..14 V, on which the information-relevant AC voltage of, for example, ±5 V is modulated. The offset voltage must first be adjusted after switching on; the high-pass (HP) filter, which is usually present, must first settle in order to suppress the DC offset for evaluation (the faster (i.e. higher) the HP filter frequency is set, the faster the settling takes place, but the less sensitive the measurement becomes for low-frequency signals).
All in all, this results in the effect that a settling time must be observed after the switch-on process. With direct connections between sensor and evaluation unit, it is necessary to wait for settling to complete once at the start of operation. In industrial plants, whose ramp-up times can be considerable, the settling time is therefore often not even noticed.

In multiplexing mode, however, the supply of current-supplied sensors is interrupted with each switchover and the bias voltage must *constantly* be re-established by the subsequent switch-on process in this operation mode. Measured data recorded during the settling time are to be considered invalid. In order to minimize the loss of system performance when signal monitoring is required to assess the validity of the measured data, it is instead advisable to determine the settling time as a function of relevant parameters. This allows a delay time to be taken into account in the measuring system between switching over and the start of measurement, which does not last longer than necessary. The duration of the settling time depends on (in descending order of weighting):

- high-pass filter settings in the measuring device (e.g. ELM3602 or EL3632),
- the IEPE current intensity (2, 4, ... mA: the higher, the shorter the settling time),
- cable length (the longer, the longer the settling time),
- cable diameter (the thicker, the shorter the settling time),
- from the design of the IEPE sensor

and can range from a few seconds to the high minute range.

The following diagram shows examples of settling times during switch-on processes with one of the combinations of ELM3602 (24 bits, 50 kSps) and ELM2742 (SolidState):

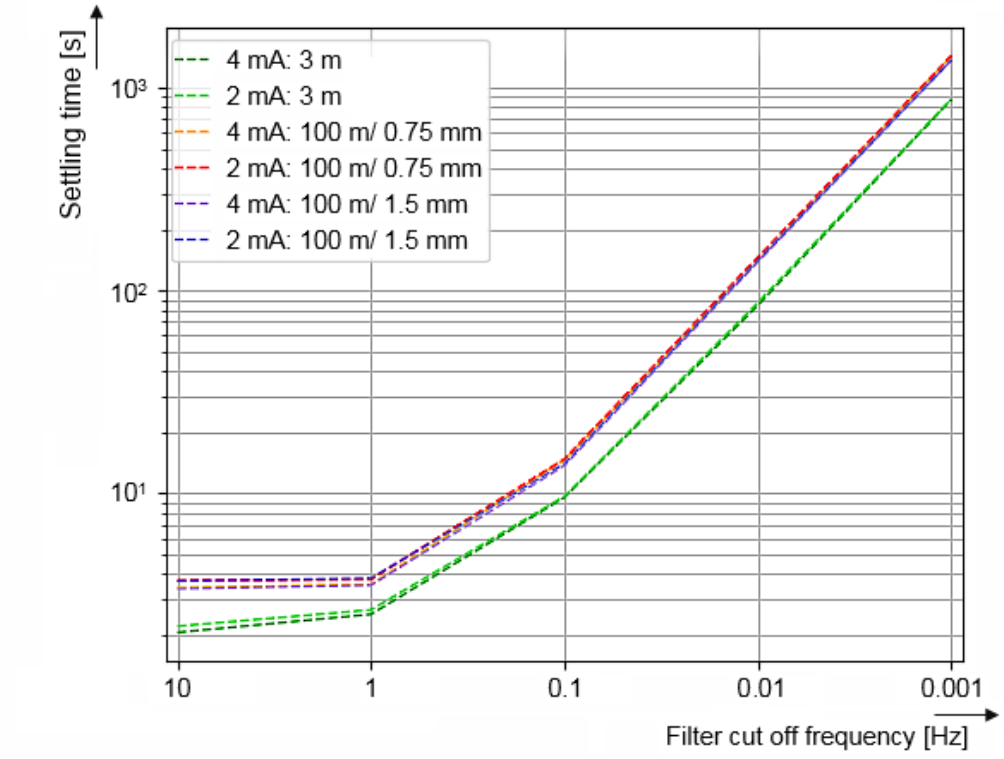


Fig. 10: Examples of settling time as a function of filter cut-off frequency (ELM3602 in combination with ELM2742)

Recommendation: if a high-pass filter can be set in the measuring device by the controller,

- a "fast" HP filter should be selected in the switching case,

- when using the ELM360x, a delay time of at least 2 s is taken into account and then,
- if necessary, the intended "slow" HP filter can be activated.

The requirement for this is that measuring device is suitable for this HP switching and does not change the measured value by the filter switching as is the case with the ELM360x IEPE terminal, for example.

4.2 Continuative documentation for I/O components with analog in and outputs

NOTE	
	**Continuative documentation Continuative documentation for I/O components with analog in and outputs** Pay also attention to the continuative documentation **I/O Analog Manual** Notes on I/O components with analog inputs and outputs, which is available in the Beckhoff Information-System and for download on the Beckhoff homepage www.beckhoff.com on the respective product pages! It explains the basics of sensor technology and contains notes on analog measured values.

4.3 Process data overview

Process data outputs

PDO		Member	Data type	Pos.	Meaning
DO Group m Outputs (m=1..4)	Outputs Channel m	Y1	BIT	0	Make contact X1, X2 → Y1 switch through
		Y2	BIT	0	Make contact X1, X2 → Y2 switch through
		Y3	BIT	0	Make contact X1, X2 → Y3 switch through
		Y4	BIT	0	Make contact X1, X2 → Y4 switch through

Process data inputs

PDO		Member	Data type	Pos.	Meaning
DO Inputs Device	Device Status	Error	BIT	0	Error within the channel, for details see further status bits or DiagnoseHistory
		Diag	BIT	4	There is a DiagMessage in the CoE 0x10F3, it can be read from the terminal, e.g. via the TwinCAT Logger
		Input cycle counter	BIT2	6, 7	Increments by 1 in each EtherCAT cycle, 0... 3. The controller can monitor this counter. If it does not increment by 1 in a cycle, this indicates a transmission error during this cycle. If necessary, adjust the ShiftTime in Distributed Clocks mode.
WcState		WcState	BIT	0	TRUE: The terminal does not participate in the EtherCAT data exchange, because it is not present or incorrectly configured FALSE: The terminal regularly participates in the data exchange. Check the validity of the data with the respective status etc.
		InputToggle	BIT	0	Toggles 0/1 when the slave sends new data via EtherCAT

PDO		Member	Data type	Pos.	Meaning
InfoData	State		UINT	0	State of the EtherCAT state machine: INIT, PreOP, SafeOP, OP
	AdsAddr	netId	AMSNETID	0	ADS address (NetID/port) through which the CoE of the slave can be reached from the application
		port	WORD	0	
	DcOutputShift		DINT	0	*~Time between SYNC0 event and reading of the outputs (in ns, only DC mode)*

PDO	Member	Data type	Pos.	Meaning
	DcInputShift	DINT	0	*~Time between SYNC0 event and reading of the inputs (in ns, only DC mode)*

Structure of the process data (TwinCAT 3) ELM264x/ ELM274x:

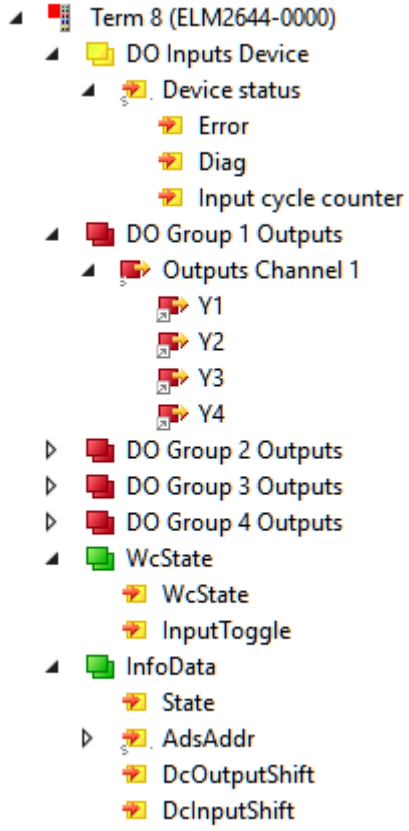


Fig. 11: PDO of the ELM2644 valid for ELM2642, ELM274x and EL2642, too (according to the number of channels)

4.4 CoE overview EL264x, ELM264x, ELM274x

4.4.1 0x70n0 DO Outputs Group [n+1]

0 ≤ n ≤ m, n+1 = channel number; m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
70n0:0	DO Outputs Group [n+1]	Max. Subindex	UINT8	RO	0x04 (4_{dec})
70n0:01	Y1	Process data value by relay Y1 at Mux [n+1]	Bit	RW	0x00 (0_{dec})
70n0:02	Y2	Process data value by relay Y2 at Mux [n+1]	Bit	RW	0x00 (0_{dec})
70n0:03	Y3	Process data value by relay Y3 at Mux [n+1]	Bit	RW	0x00 (0_{dec})
70n0:04	Y4	Process data value by relay Y4 at Mux [n+1]	Bit	RW	0x00 (0_{dec})

4.4.2 0x90n0 DO Info data Group [n+1]

0 ≤ n ≤ m, n+1 = channel number; m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:0	DO Info data Group [n+1]	Max. Subindex	UINT8	RO	0x04 (4_{dec})
90n0:01	Y1 switch cycle counter	Switching cycle counter by relay Y1 at Mux [n+1] since production	UINT32	RO	0x00000000 (0_{dec})
90n0:02	Y2 switch cycle counter	Switching cycle counter by relay Y2 at Mux [n+1] since production	UINT32	RO	0x00000000 (0_{dec})
90n0:03	Y3 switch cycle counter	Switching cycle counter by relay Y3 at Mux [n+1] since production	UINT32	RO	0x00000000 (0_{dec})
90n0:04	Y4 switch cycle counter	Switching cycle counter by relay Y4 at Mux [n+1] since production	UINT32	RO	0x00000000 (0_{dec})

4.4.3 0xF600 DO Device Status Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
F600:0	DO Device Status Inputs	Max. Subindex	UINT8	RO	0x0F (15_{dec})
F600:01	Error	Error state. See Diagnosis messages.	Bit	RO	0x00 (0_{dec})
F600:0D	Diag	A new diagnosis message is available.	Bit	RO	0x00 (0_{dec})
F600:0F	Input cycle counter	Will be incremented in each EtherCAT cycle (0.1.2.0.1.2...).	BIT2	RO	0x00 (0_{dec})

4.4.4 0xF900 DO Device Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
F900:0	DO Device Info data	Max. Subindex	UINT8	RO	0x12 (18_{dec})
F900:12	Operating time	Duty cycle since production in minutes.	UINT32	RO	0x00000000 (0_{dec})

5 Commissioning on EtherCAT Master

5.1 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 to the controlling task for diagnosis that is accurate for the current cycle when in operation (not during commissioning) are discussed below.

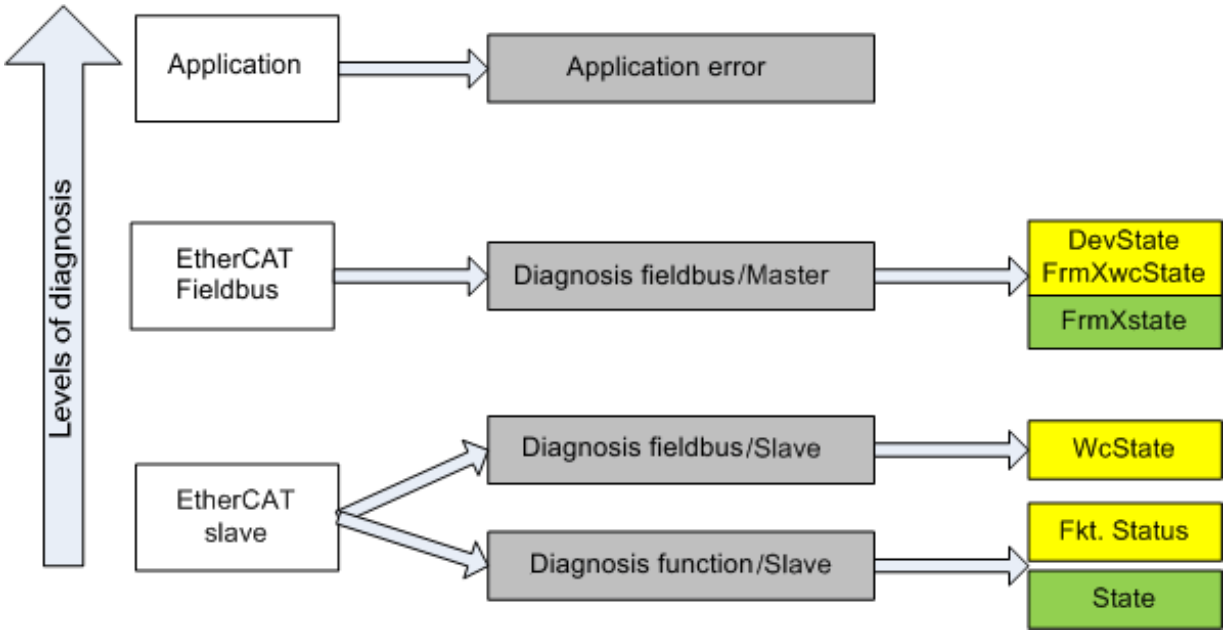


Fig. 12: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

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

  as well as
- function diagnosis typical for a channel (device-dependent)
  See the corresponding device documentation

The colors in Fig. *Selection of the diagnostic information of an EtherCAT Slave* also correspond to the variable colors in the System Manager, see Fig. *Basic EtherCAT Slave Diagnosis in the PLC*.

Colour	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

Colour	Meaning
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 of an implementation of basic EtherCAT Slave Diagnosis. A Beckhoff EL3102 (2-channel analogue 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.

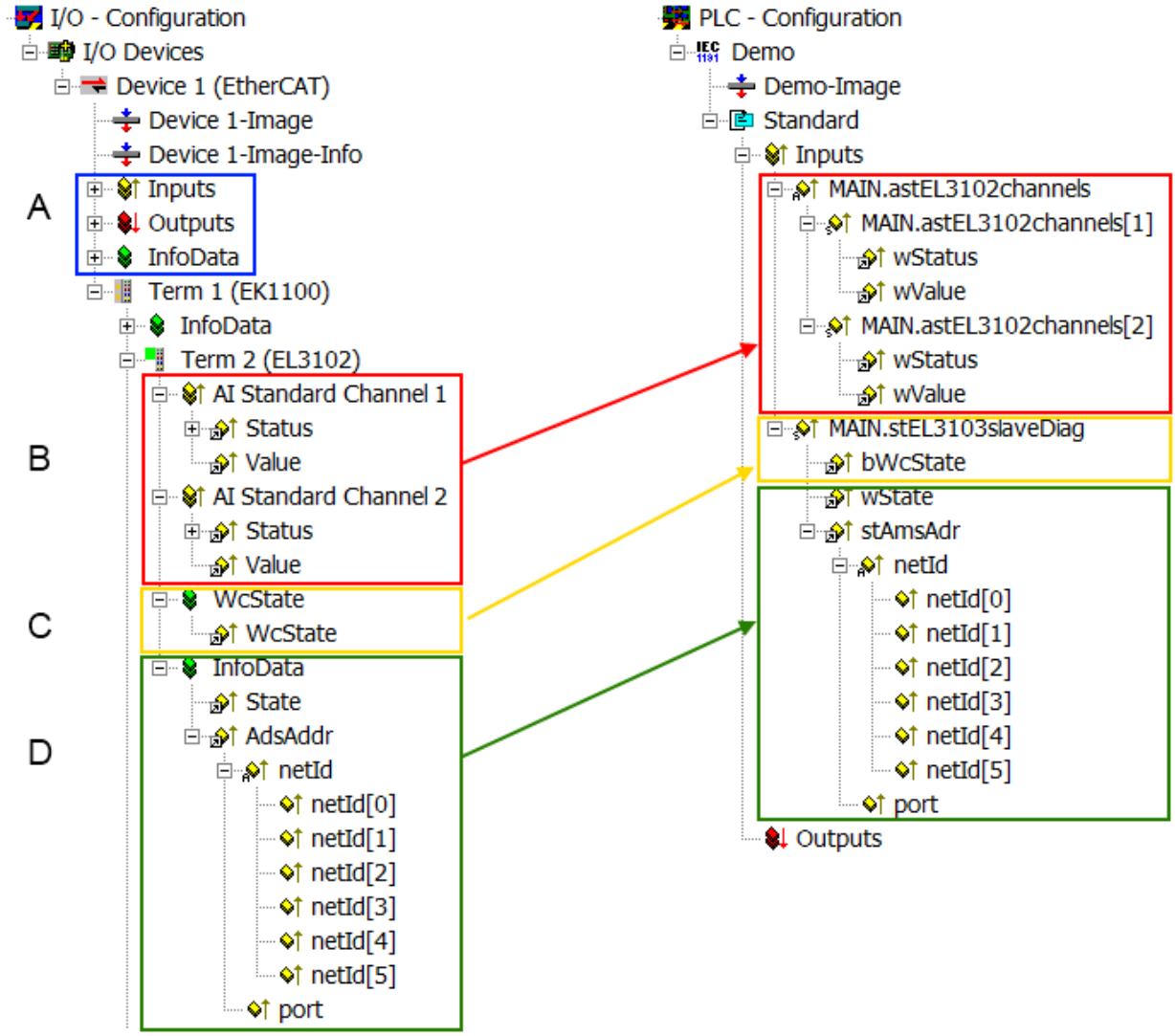


Fig. 13: Basic EtherCAT Slave Diagnosis in the PLC

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

Code	Function	Implementation	Application/evaluation
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.

NOTE

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 complex EtherCAT Slave. It can be accessed through the TwinCAT System Manager, see Fig. *EL3102, CoE directory*:

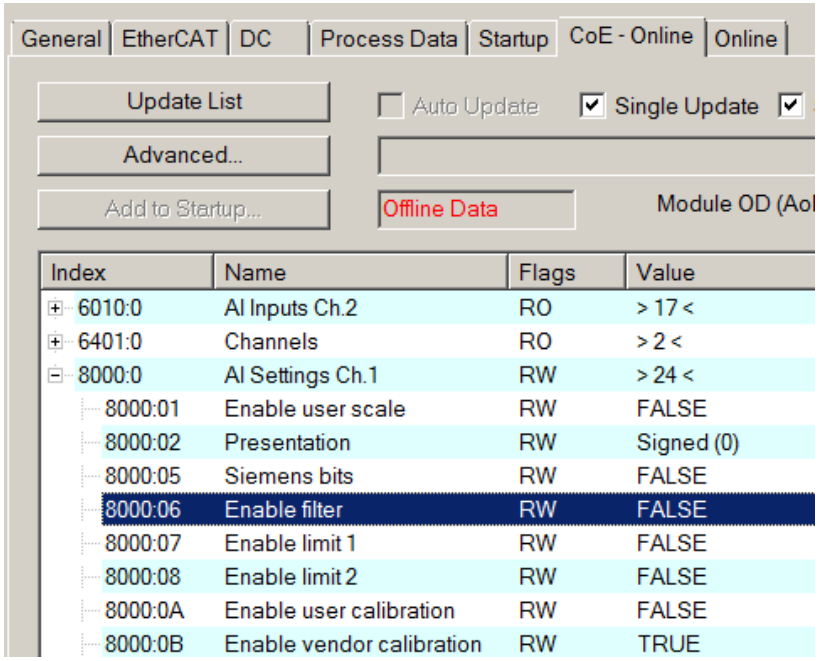


Fig. 14: EL3102, CoE directory

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 TwinCAT System Managers from TwinCAT 2.11R2 and above. They are integrated into the System Manager through appropriately extended ESI configuration files.

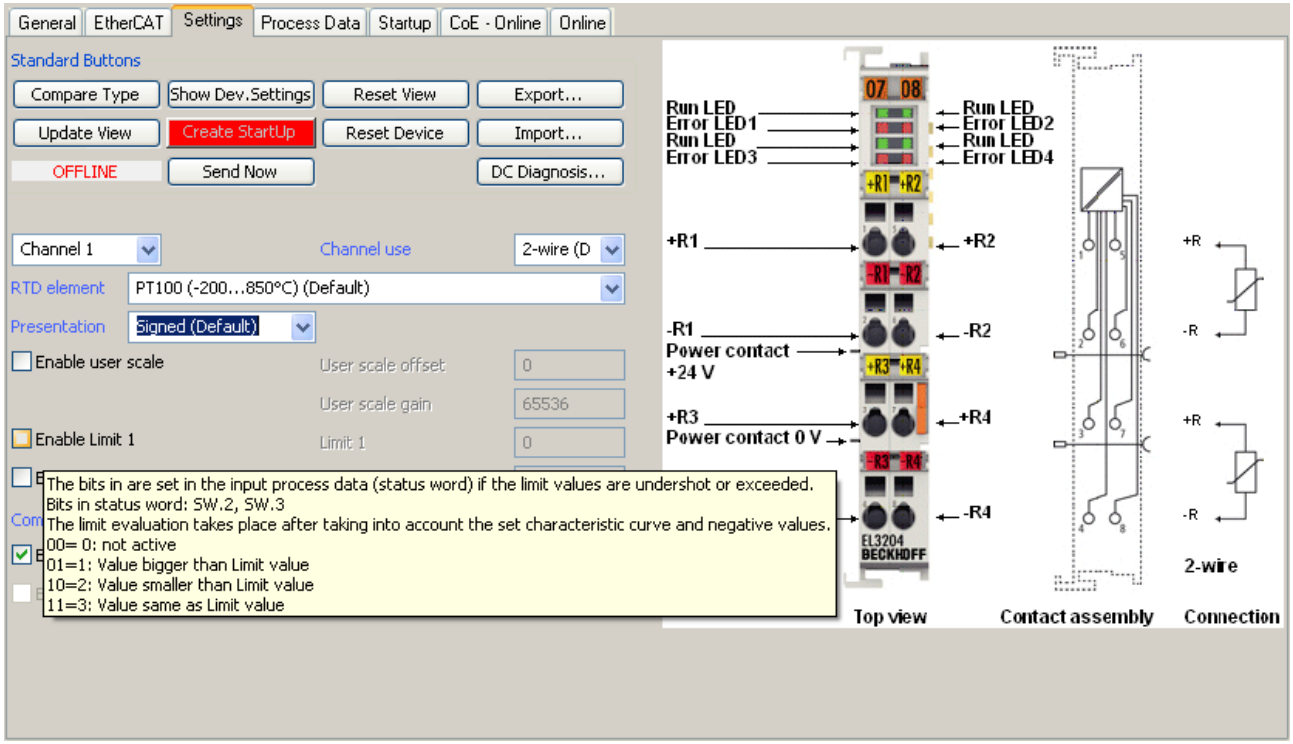


Fig. 15: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

- CoE Parameter Directory
- DC/FreeRun mode
- the available process data records (PDO)

Although the “Process Data”, “DC”, “Startup” and “CoE-Online” that used to be necessary for this are still displayed, it is recommended that, if the commissioning aid is used, the automatically generated settings are not changed by it.

The commissioning tool does not cover every possible application of an EL/EP device. If the available setting options are not adequate, the user can make the DC, PDO and CoE settings manually, as in the past.

EtherCAT State: automatic default behaviour of the TwinCAT System Manager and manual operation

After the operating power is switched on, an EtherCAT Slave must go through the following statuses

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT Master directs these statuses 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 the section on "Principles of Communication, EtherCAT State Machine [▶ 111]" in this connection. Depending how much configuration has to be done, and on the overall communication, 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 wanted by the user, and which is brought about automatically at start-up by TwinCAT, can be set in the System Manager. As soon as TwinCAT reaches the status RUN, the TwinCAT EtherCAT Master will approach 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.

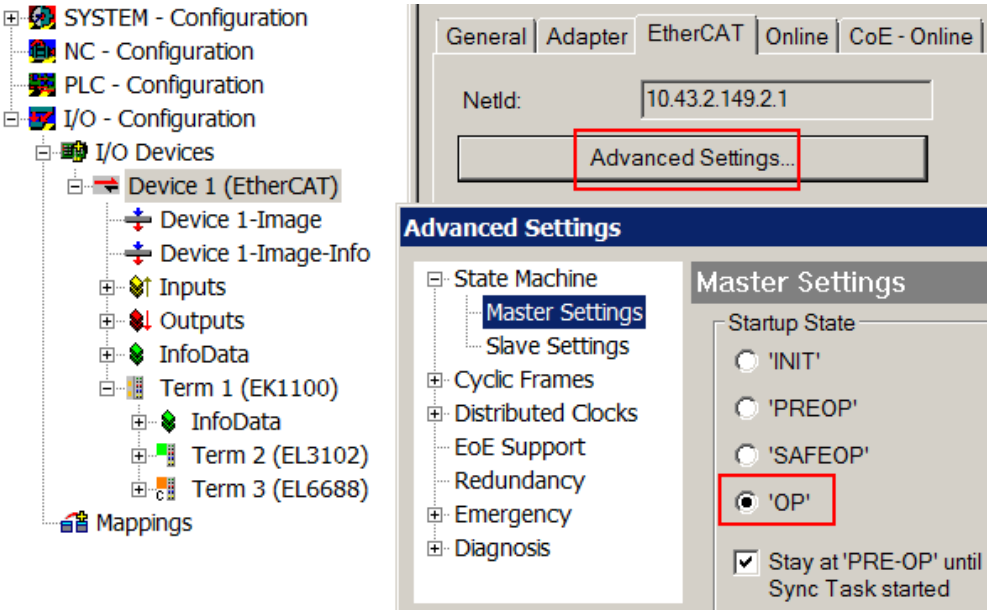


Fig. 16: Default behaviour of the System Manager

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

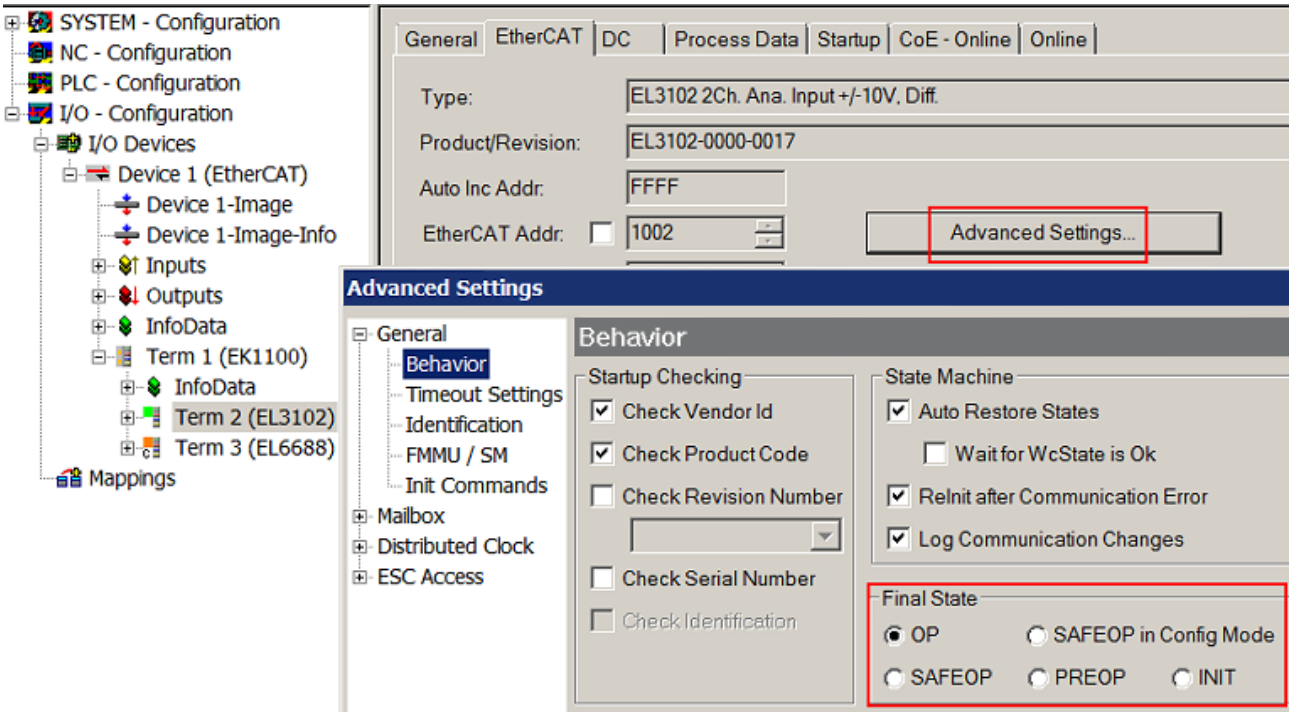


Fig. 17: Default target state in the Slave

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.

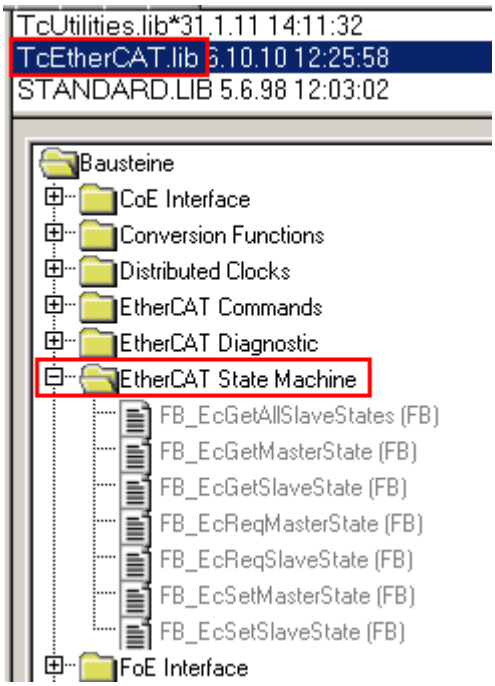
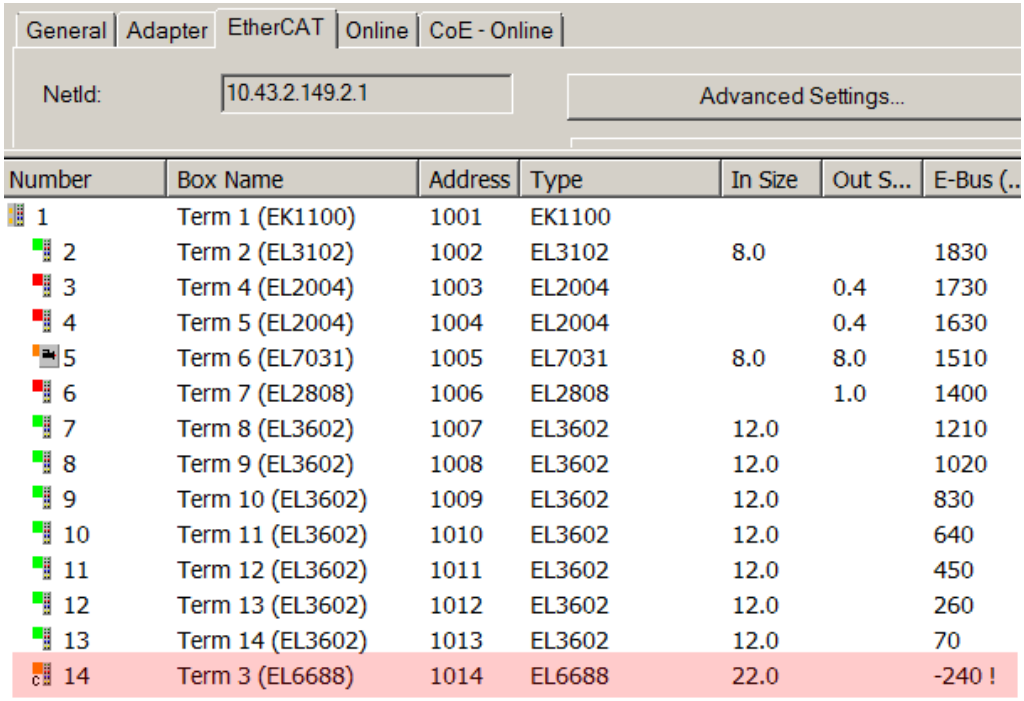


Fig. 18: PLC function blocks

Note regarding E-Bus current

EL/ES terminals are placed on the DIN rail at a coupler on the terminal strand. A Bus Coupler can supply the EL terminals added to it with the E-bus system voltage of 5 V; a coupler is thereby loadable up to 2 A as a rule. Information on how much current each EL terminal requires from the E-bus supply is available online and in the catalogue. If the added terminals require more current than the coupler can supply, then power feed terminals (e.g. EL9410) must be inserted at appropriate places in the terminal strand.

The pre-calculated theoretical maximum E-Bus current is displayed in the TwinCAT System Manager as a column value. A shortfall is marked by a negative total amount and an exclamation mark; a power feed terminal is to be placed before such a position.



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. 19: Illegally exceeding the 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:

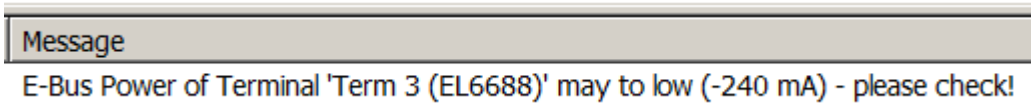


Fig. 20: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

The same ground potential must be used for the E-Bus supply of all EtherCAT terminals in a terminal block!

5.2 TwinCAT Quick Start

TwinCAT is a development environment for real-time control including a 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 System Manual:**
  Fieldbus Components → EtherCAT Terminals → EtherCAT System Documentation → Setup in the TwinCAT System Manager
- **TwinCAT 2** → TwinCAT System Manager → I/O Configuration
- In particular, for TwinCAT – driver installation:
  Fieldbus components → Fieldbus Cards and Switches → FC900x – PCI Cards for Ethernet → Installation

Devices contain the relevant 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 the offline mode can be found under http://infosys.beckhoff.com:
    TwinCAT 2 → TwinCAT System Manager → IO Configuration → Add 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 between the user PC and individual control elements:

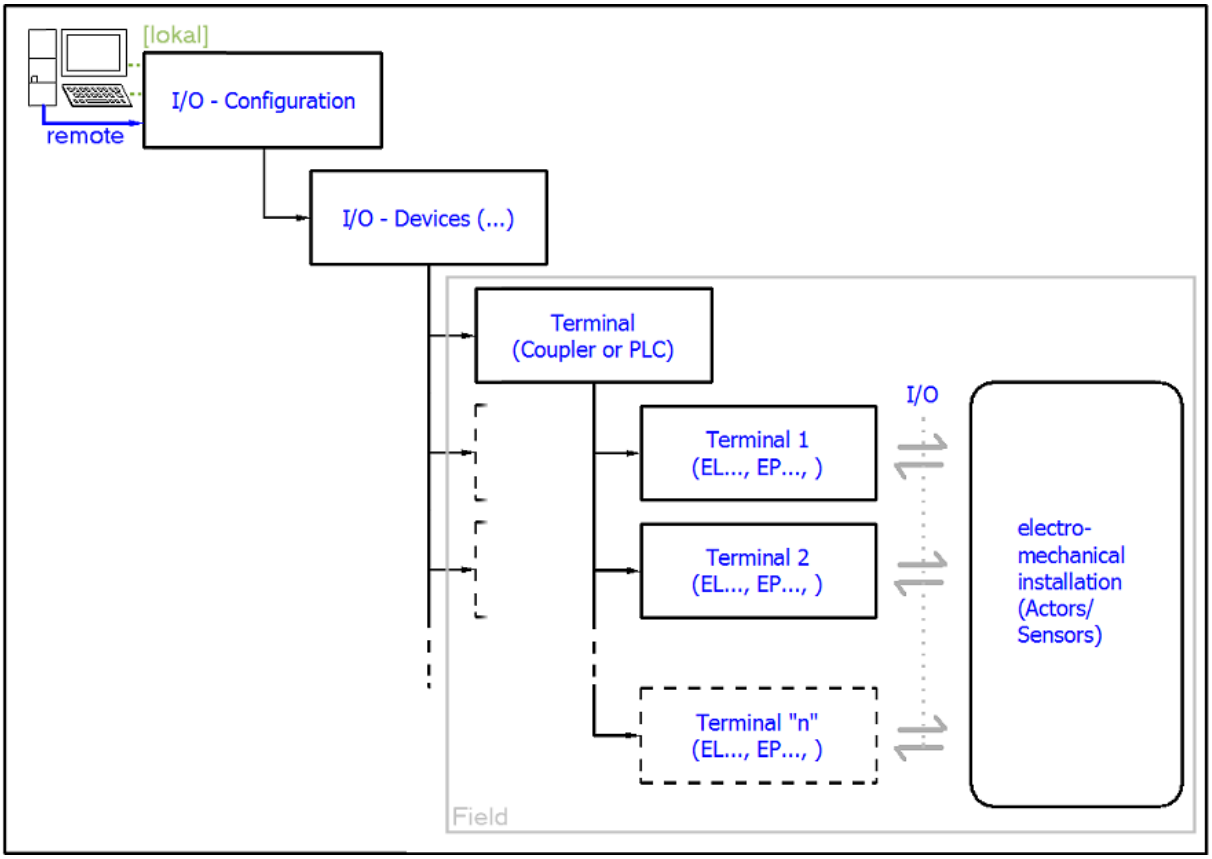


Fig. 21: Relationship between user side (commissioning) and installation

Insertion of certain components (I/O device, terminal, box...) by users functions the same way as in TwinCAT 2 and TwinCAT 3. The descriptions below relate solely to the online procedure.

Example configuration (actual configuration)

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

- **CX2040** control system (PLC) including **CX2100-0004** power supply unit
- Connected to 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)

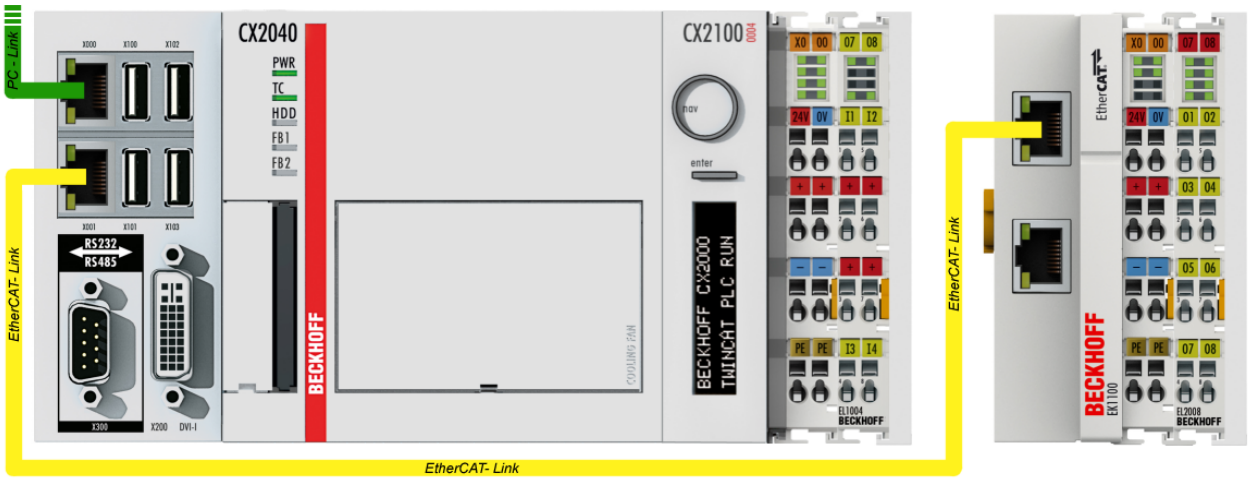


Fig. 22: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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.

5.2.1 TwinCAT 2

Startup

TwinCAT 2 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:

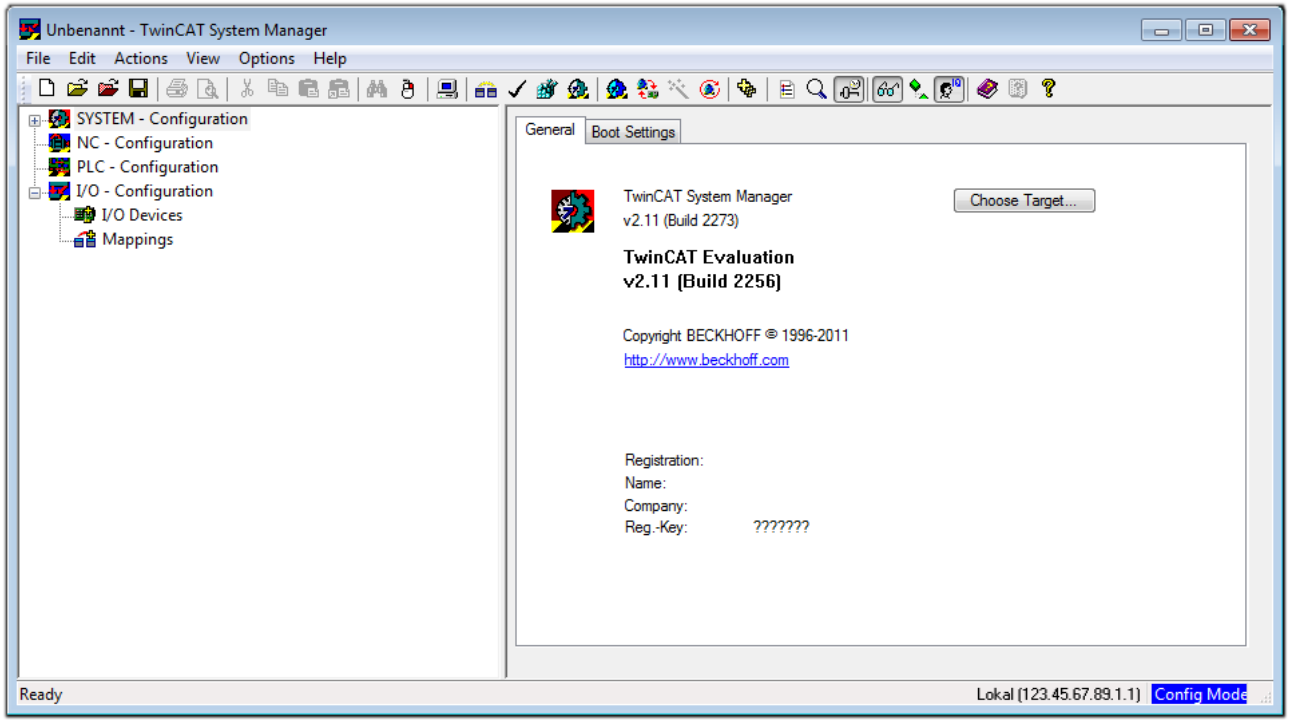


Fig. 23: Initial TwinCAT 2 user interface

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 thus the next step is “Insert Device [▶ 47]”.

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

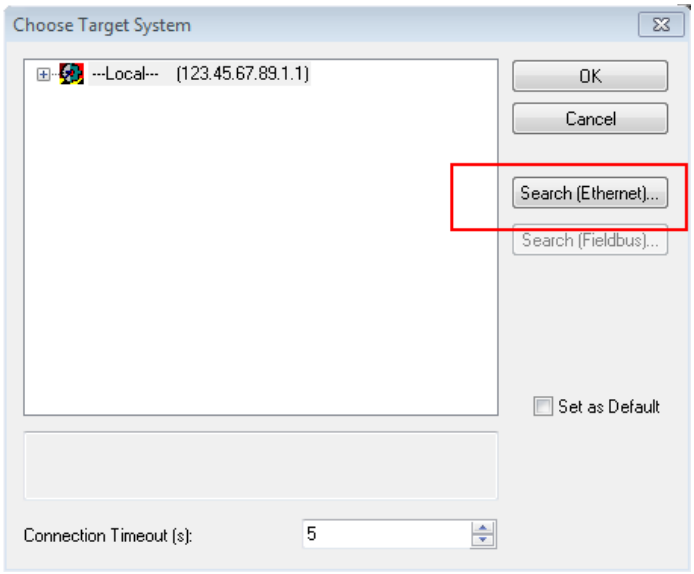


Fig. 24: Selection of the target system

Use “Search (Ethernet)...” to enter the target system. Thus another 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

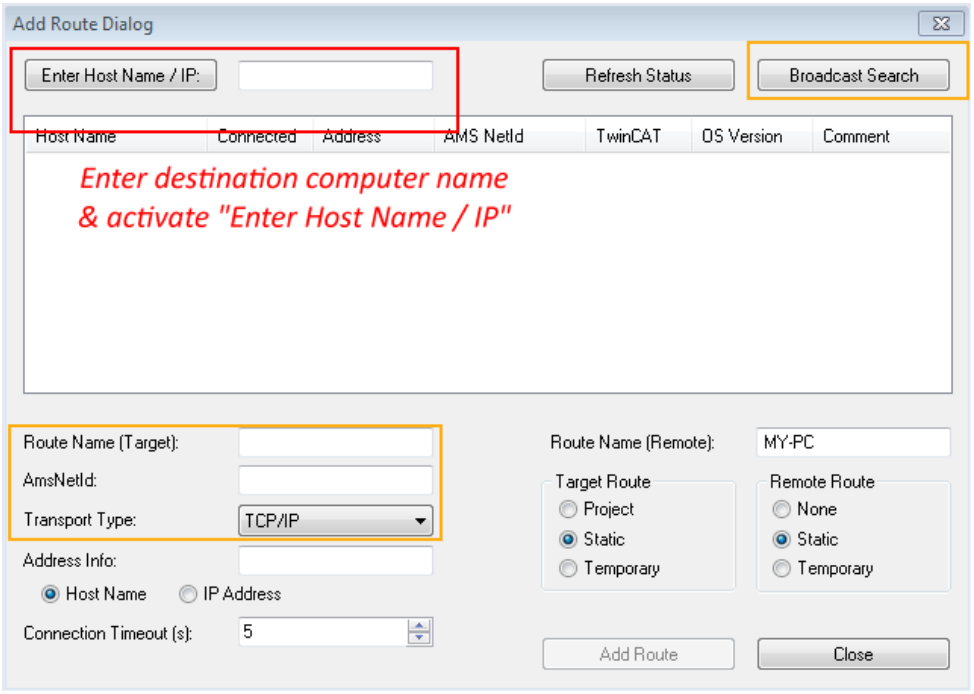
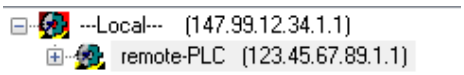


Fig. 25: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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 the menu
“Actions” → “Set/Reset TwinCAT to Config Mode…” (Shift + F4).

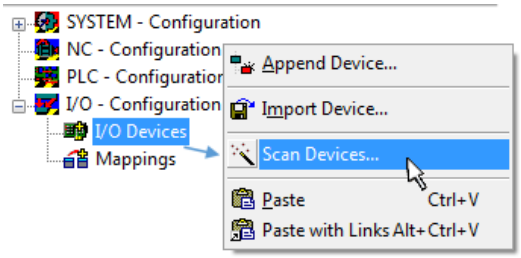


Fig. 26: Select “Scan Devices...”

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

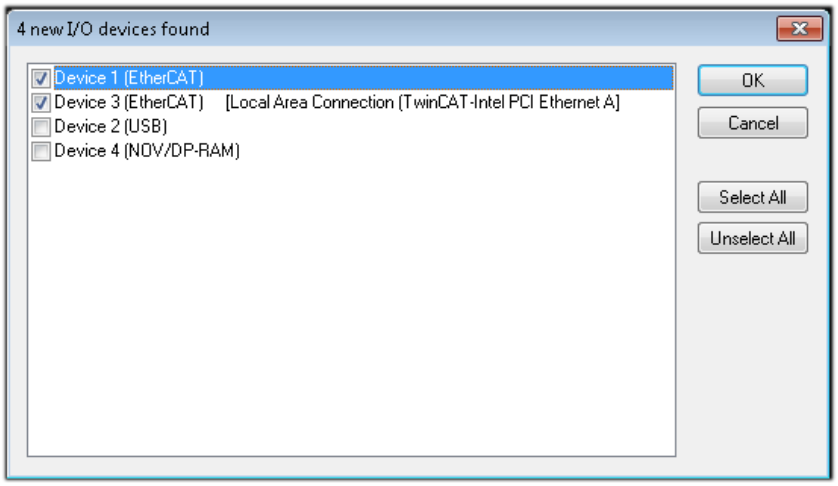


Fig. 27: Automatic detection of I/O devices: selection of the devices to be integrated

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 example configuration [▶ 44] described at the beginning of this section, the result is as follows:

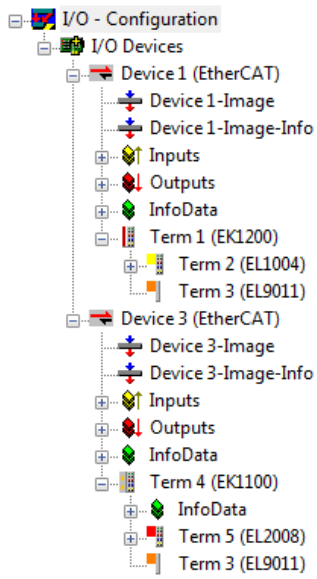


Fig. 28: Mapping of the configuration in the TwinCAT 2 System Manager

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

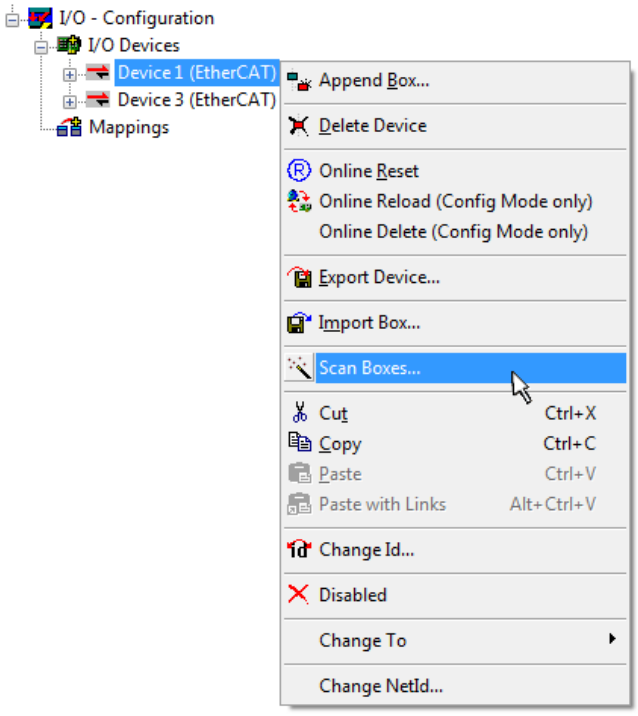


Fig. 29: Reading of individual terminals connected to a device

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 generating 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 solely to Structured Text (ST).

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

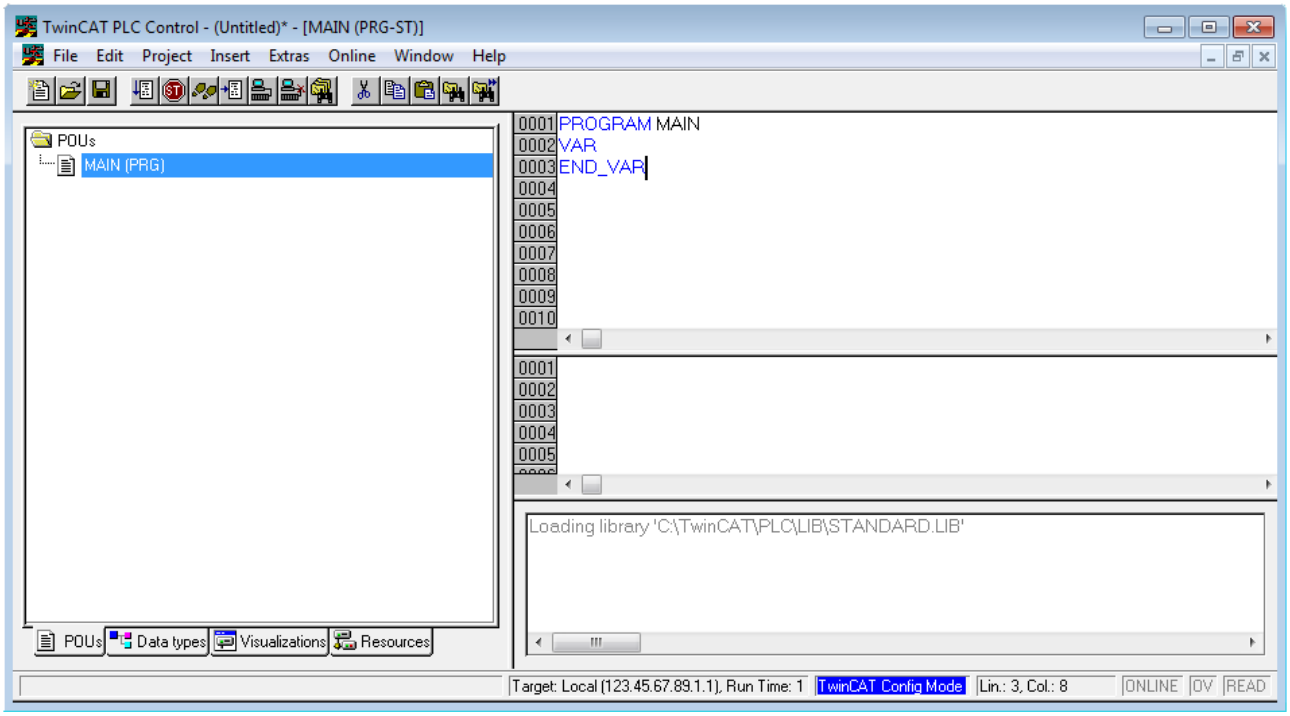


Fig. 30: TwinCAT PLC Control after startup

Example variables and an example program have been created and stored under the name “PLC_example.pro”:

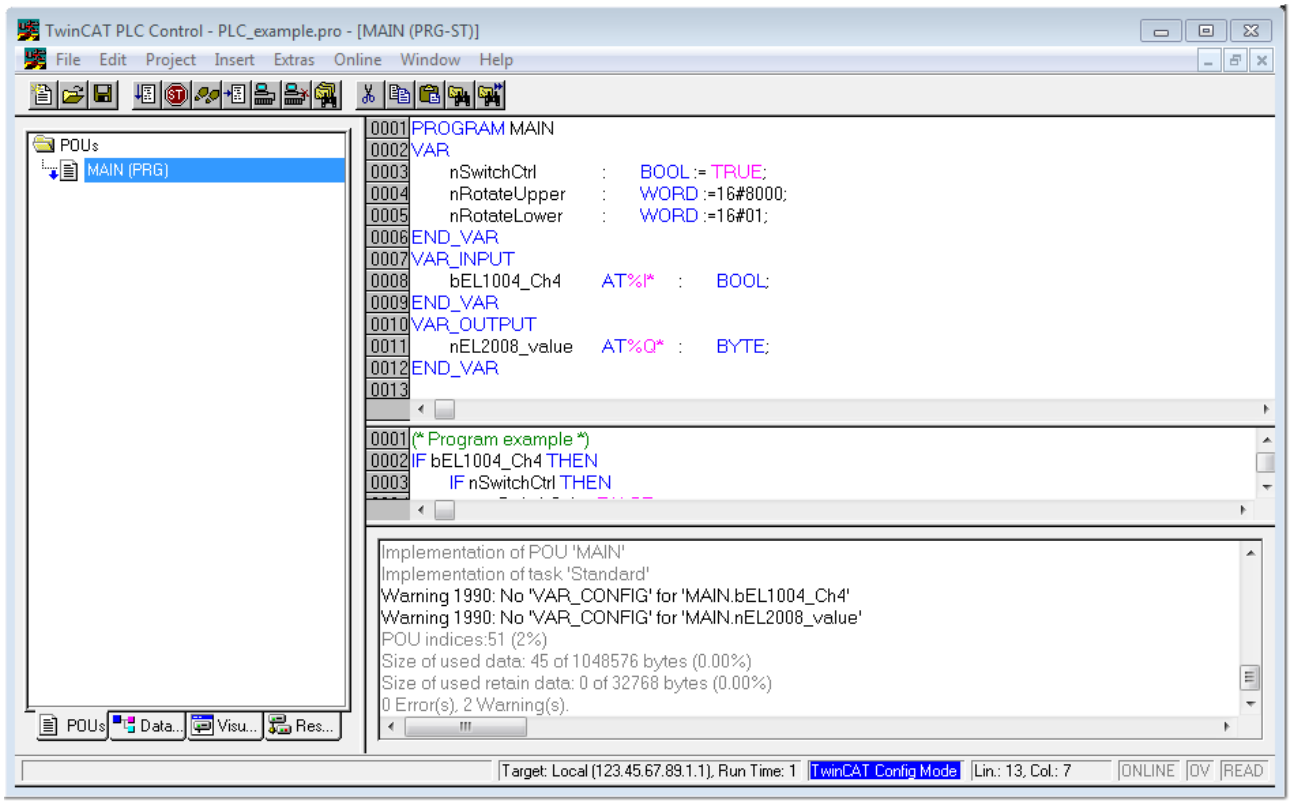


Fig. 31: Example program with variables after a compile process (without variable integration)

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**. This is performed via the context menu of the PLC configuration (right-click) and selecting “Append PLC Project…”:

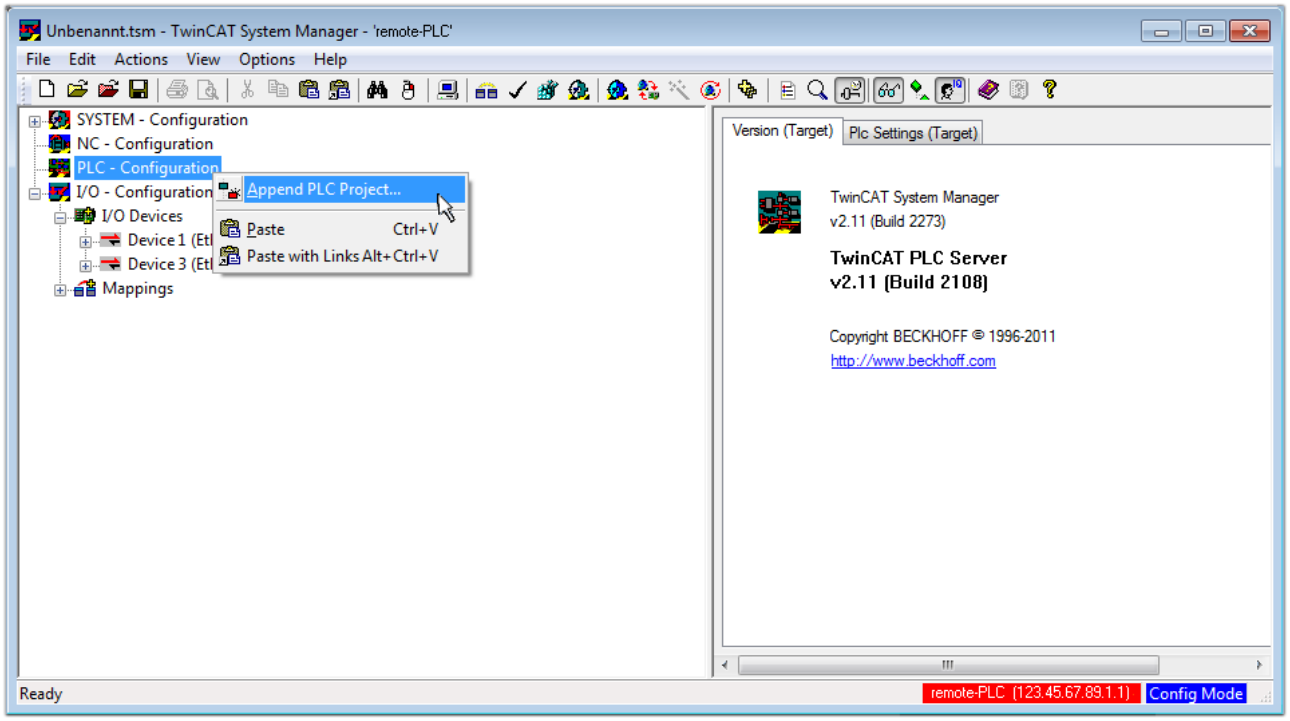


Fig. 32: Appending the TwinCAT PLC Control project

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:

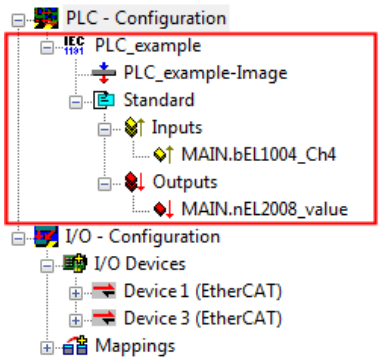


Fig. 33: PLC project integrated in the PLC configuration of the System Manager

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”:

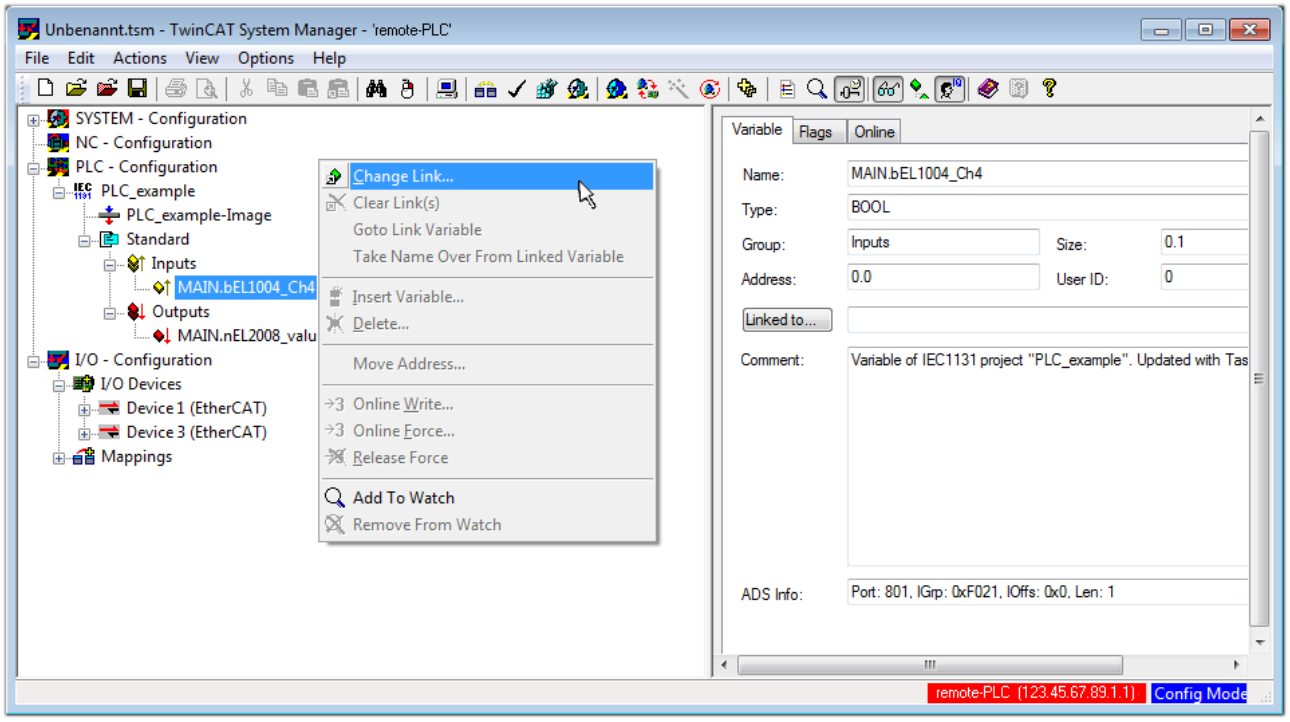


Fig. 34: Creating the links between PLC variables and process objects

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

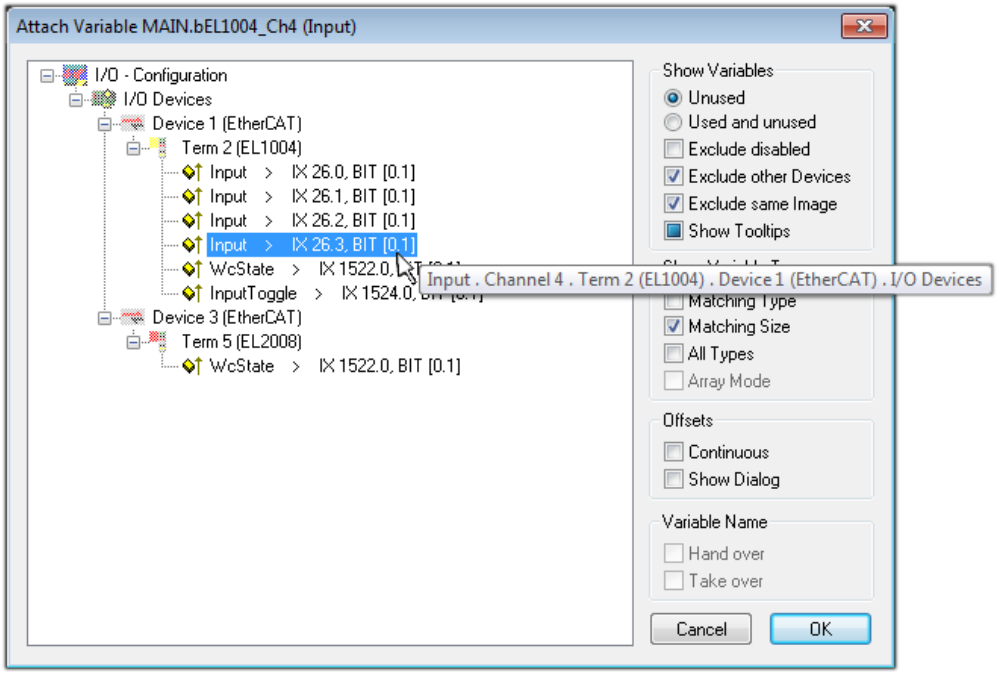


Fig. 35: Selecting BOOL-type PDO

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

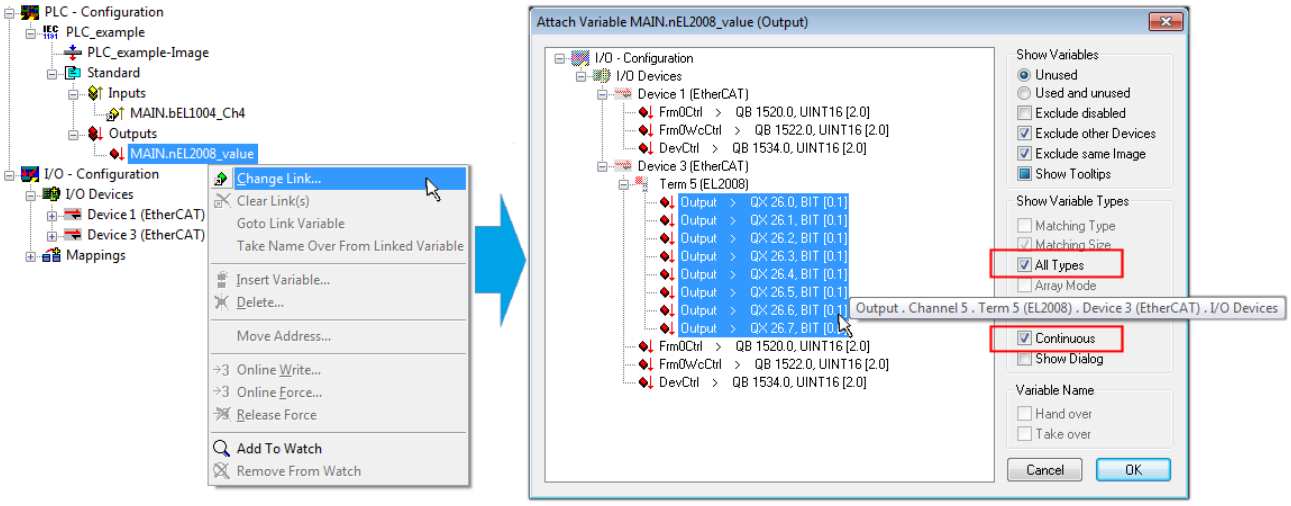


Fig. 36: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus 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 ( ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

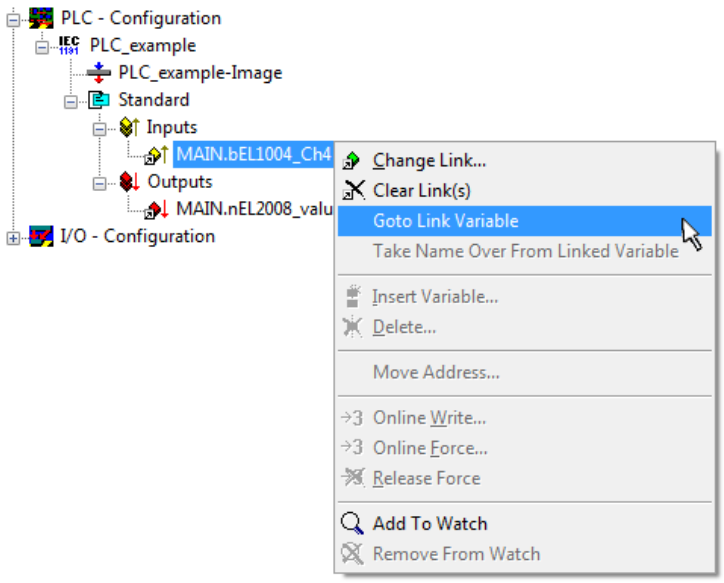
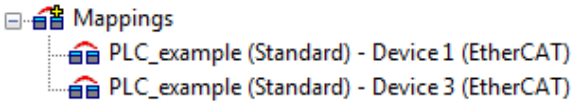


Fig. 37: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of assigning variables to the PDO is completed via the menu option “Actions” → “Create assignment”, or via  .

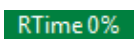
This can be visualized in the configuration:



The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would 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 also possible to allocate this to a set of bit-standardized variables. Here, too, a “Goto Link Variable” 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 will be 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 the Ethernet via “Online” → “Choose Runtime System…”:

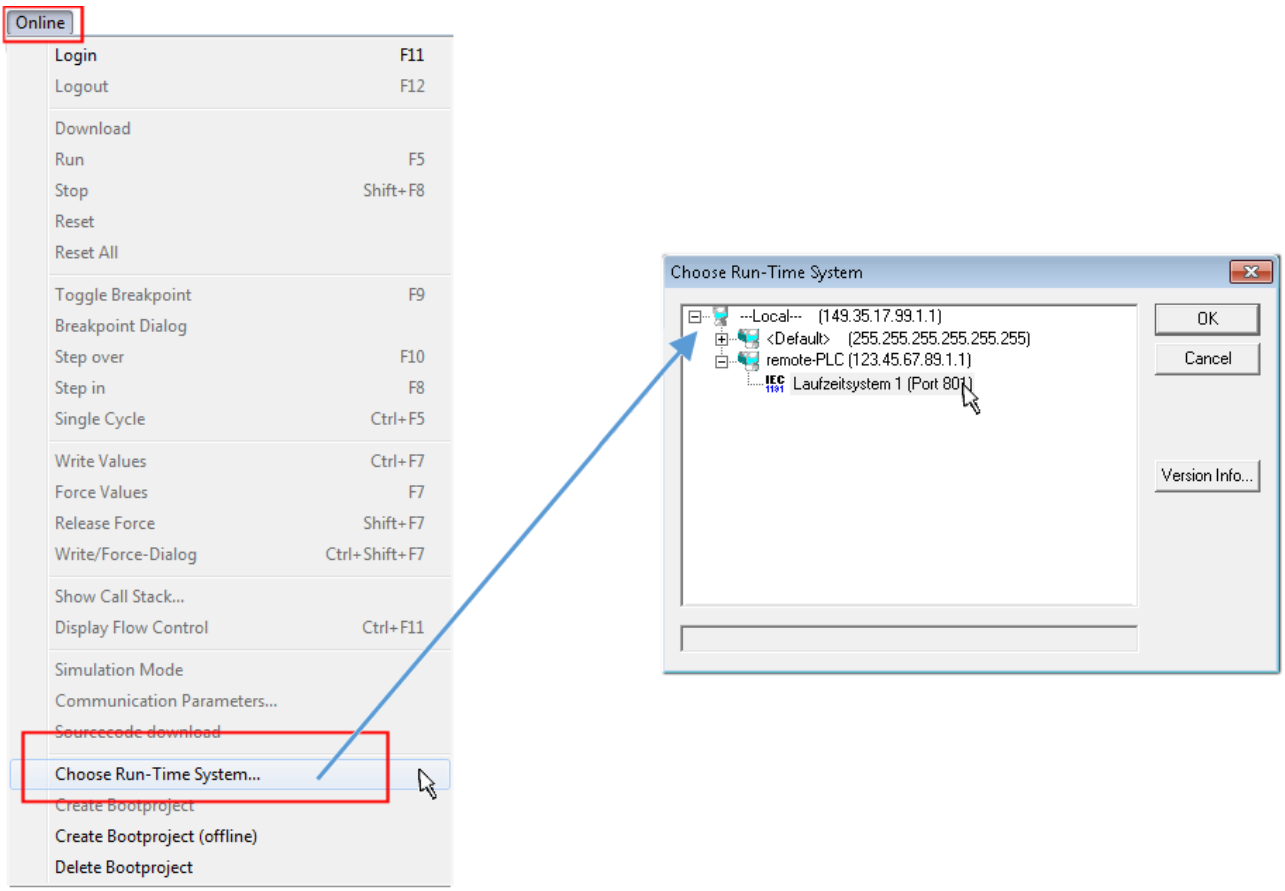


Fig. 38: Choose target system (remote)

In this example, “Runtime system 1 (port 801)” is selected and confirmed. Link the PLC with the real-time system via the 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 confirmed with “Yes”. The runtime environment is ready for the program start:

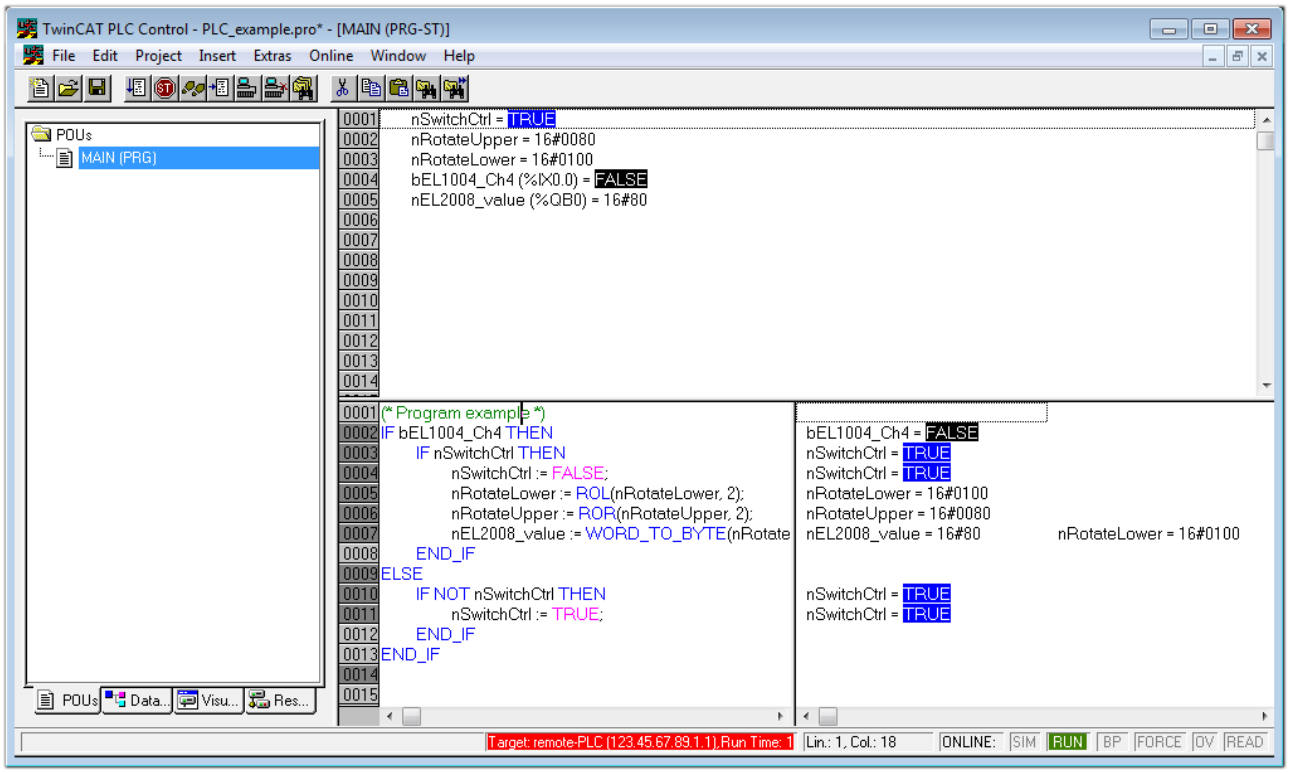


Fig. 39: PLC Control logged in, ready for program startup

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

5.2.2 TwinCAT 3

Startup

TwinCAT 3 makes the development environment areas available all together, with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (see “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:

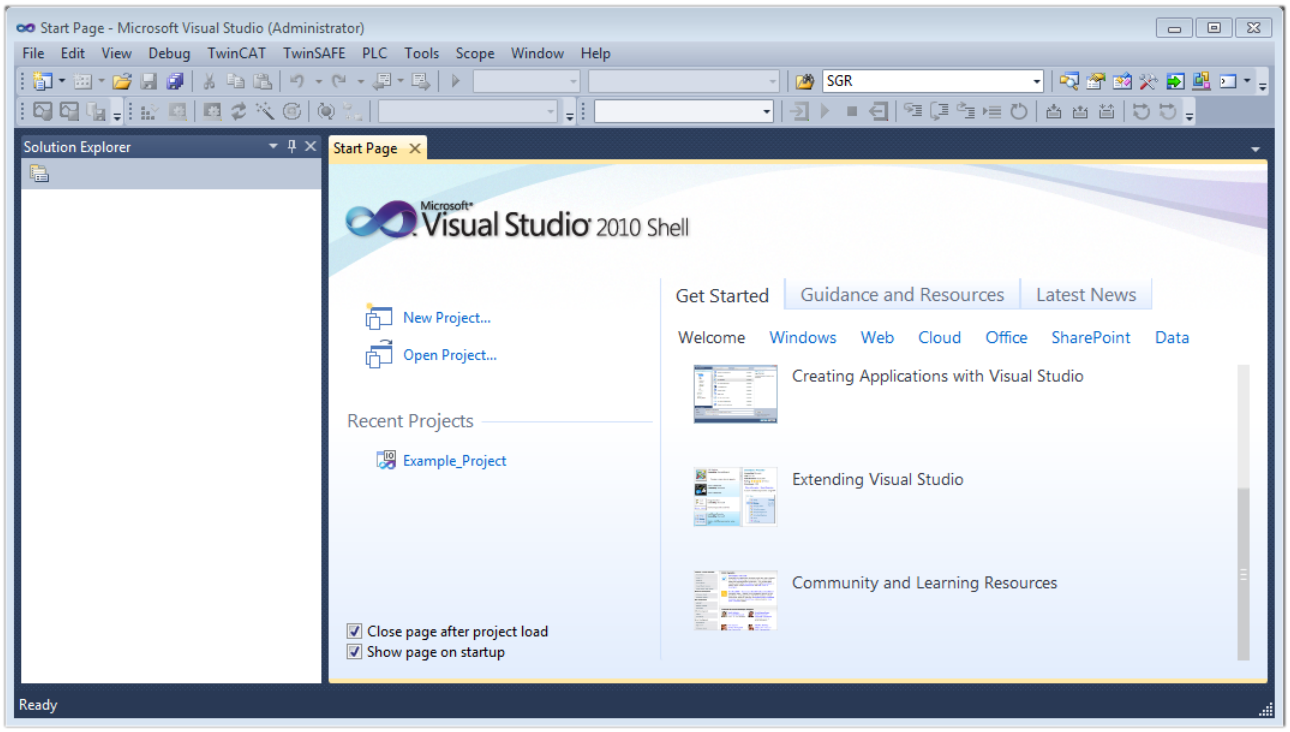


Fig. 40: Initial TwinCAT 3 user interface

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):

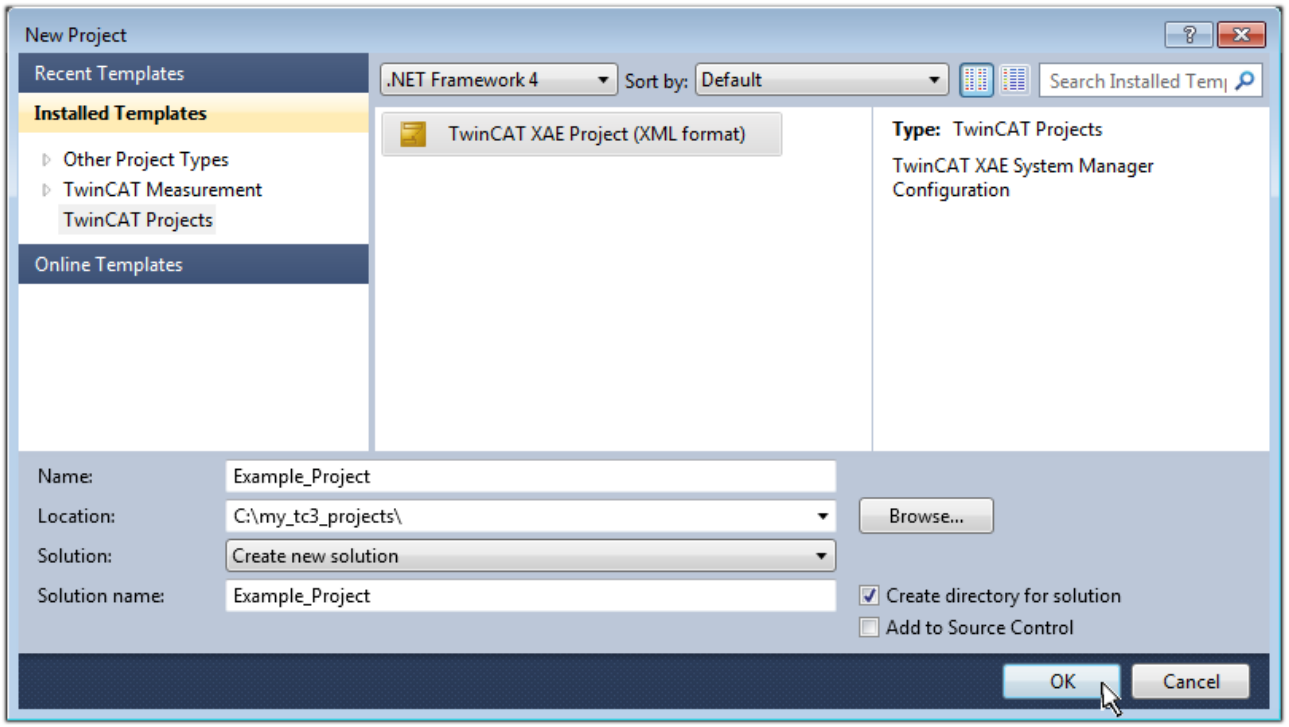


Fig. 41: Create new TwinCAT 3 project

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

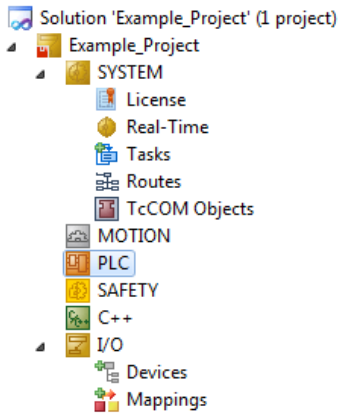
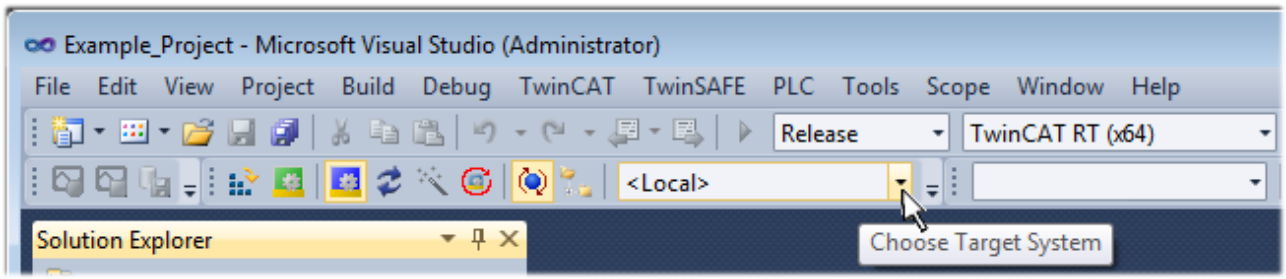


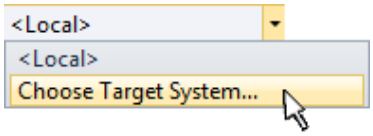
Fig. 42: New TwinCAT 3 project in the project folder explorer

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 (locally), TwinCAT can be used in local mode and the process can be continued with the next step, "Insert Device [▶ 58]".

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, 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:

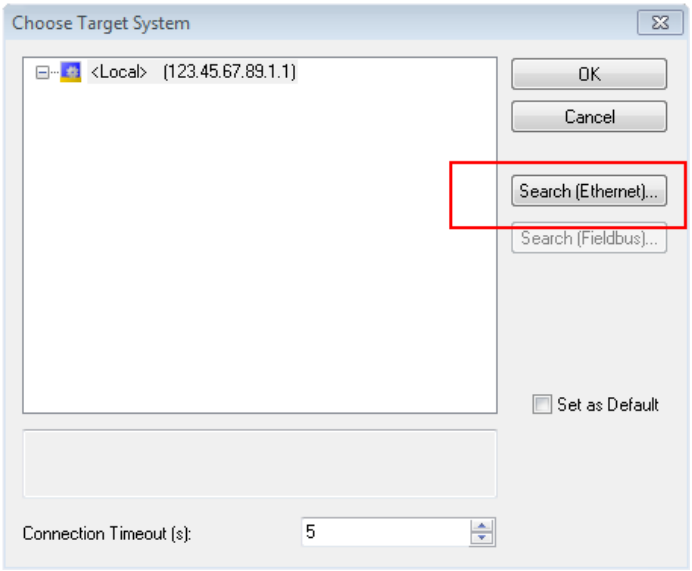


Fig. 43: Selection dialog: Choose the target system

Use “Search (Ethernet)...” to enter the target system. Thus another 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

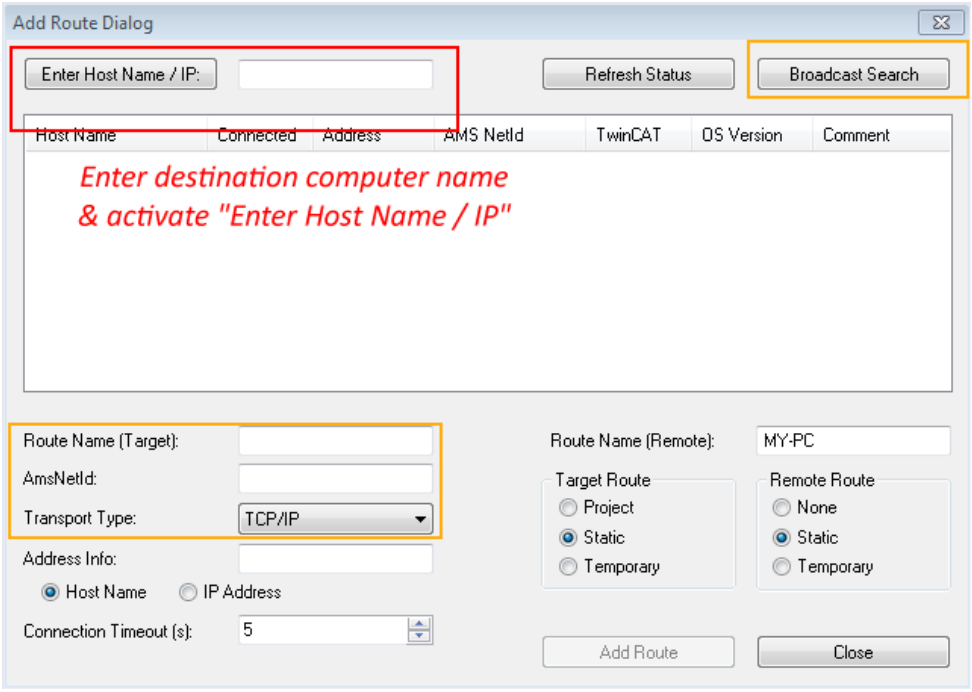
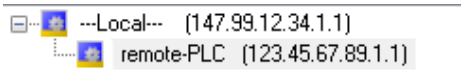


Fig. 44: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

In the project folder explorer on the left of the Visual Studio shell user interface, select “Devices” within the 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)”.

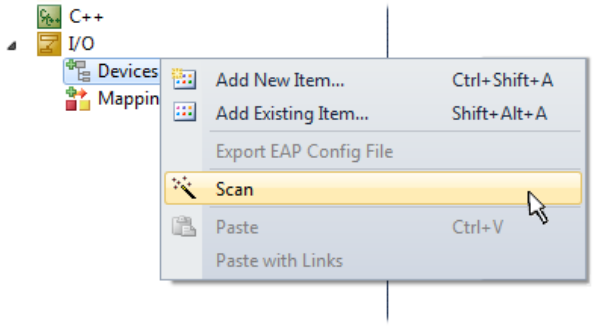


Fig. 45: Select “Scan”

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

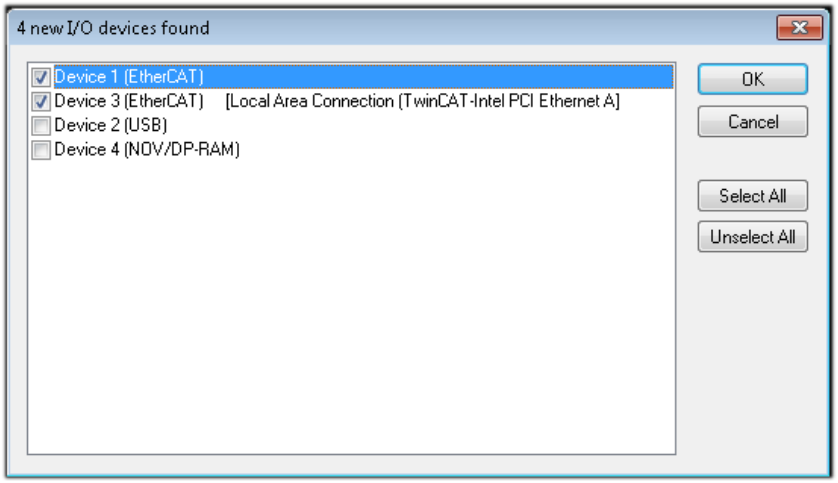


Fig. 46: Automatic detection of I/O devices: selection of the devices to be integrated

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 example configuration [▶ 44] described at the beginning of this section, the result is as follows:

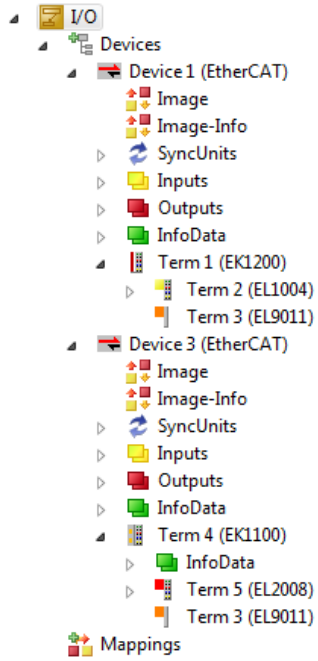


Fig. 47: Mapping of the configuration in VS shell of the TwinCAT 3 environment

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

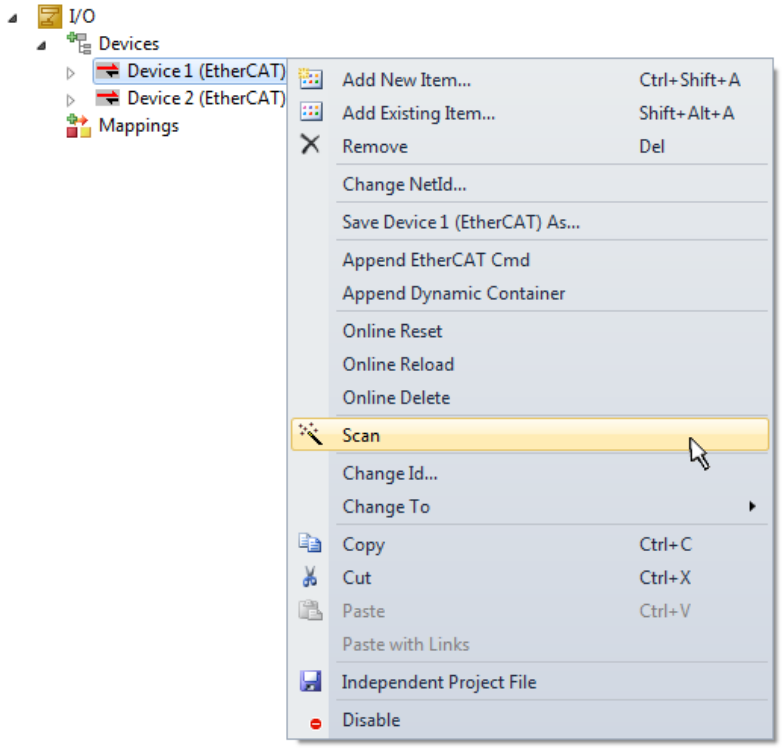


Fig. 48: Reading of individual terminals connected to a device

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

Programming the PLC

TwinCAT PLC Control is the development environment for generating 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 solely to Structured Text (ST).

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

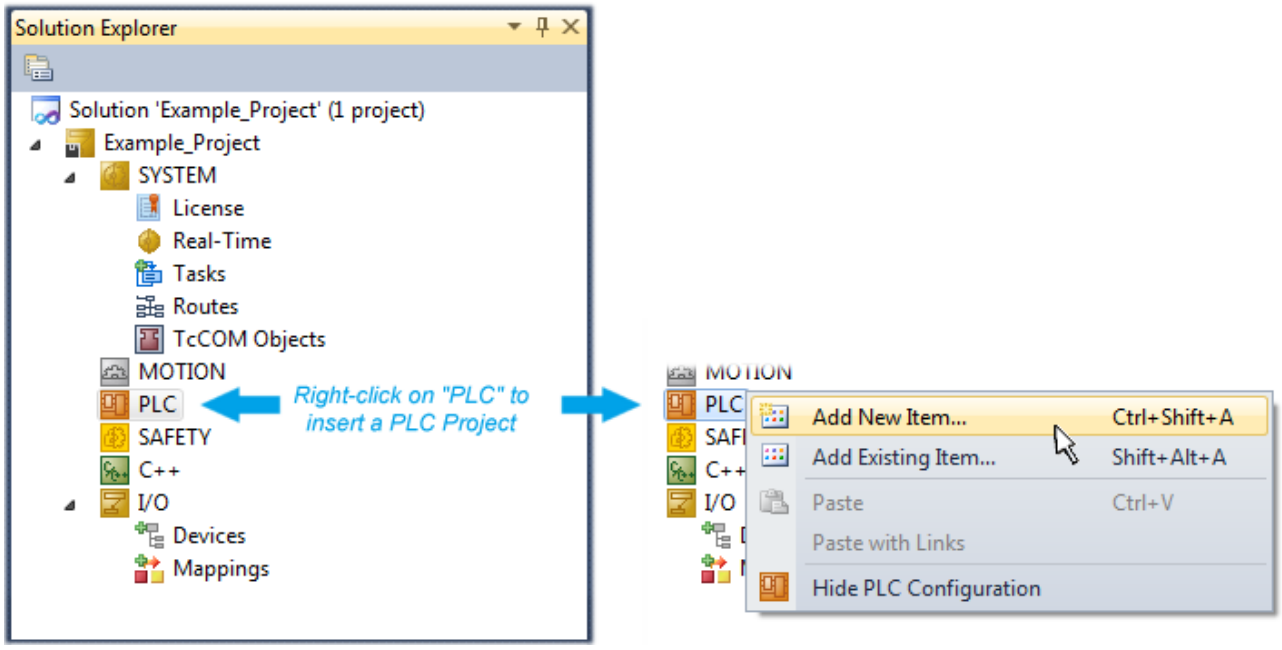


Fig. 49: Adding the programming environment in “PLC”

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

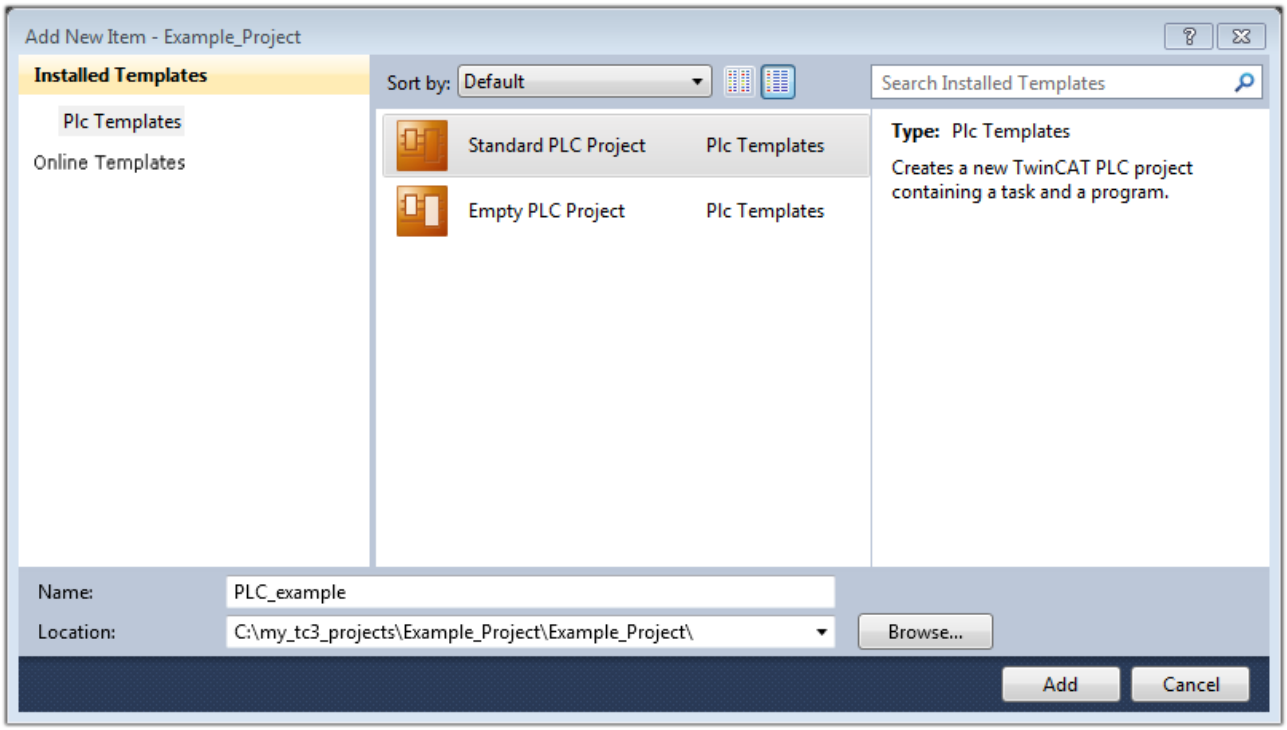


Fig. 50: Specifying the name and directory for the PLC programming environment

The “Main” program, which already exists due to 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:

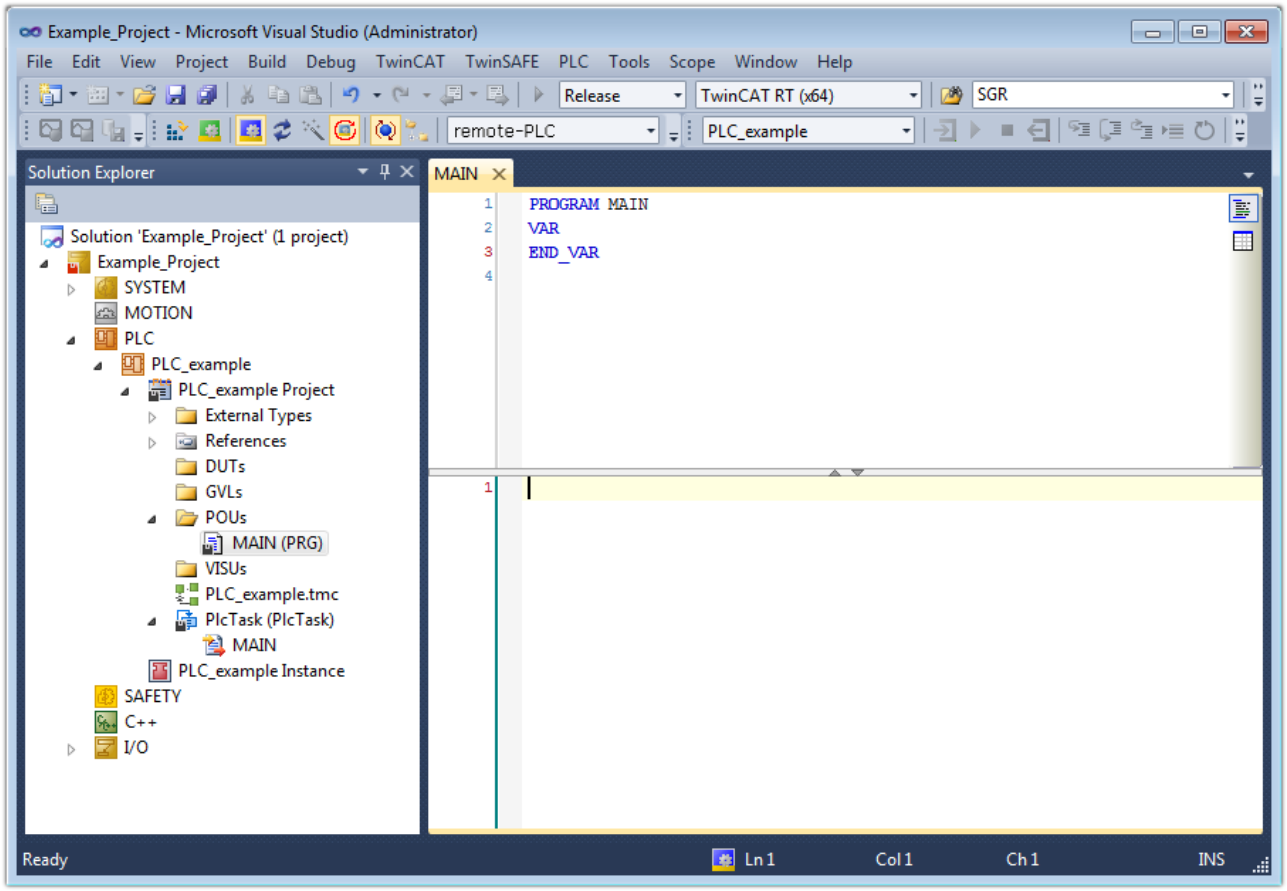


Fig. 51: Initial “Main” program for the standard PLC project

Now example variables and an example program have been created for the next stage of the process:

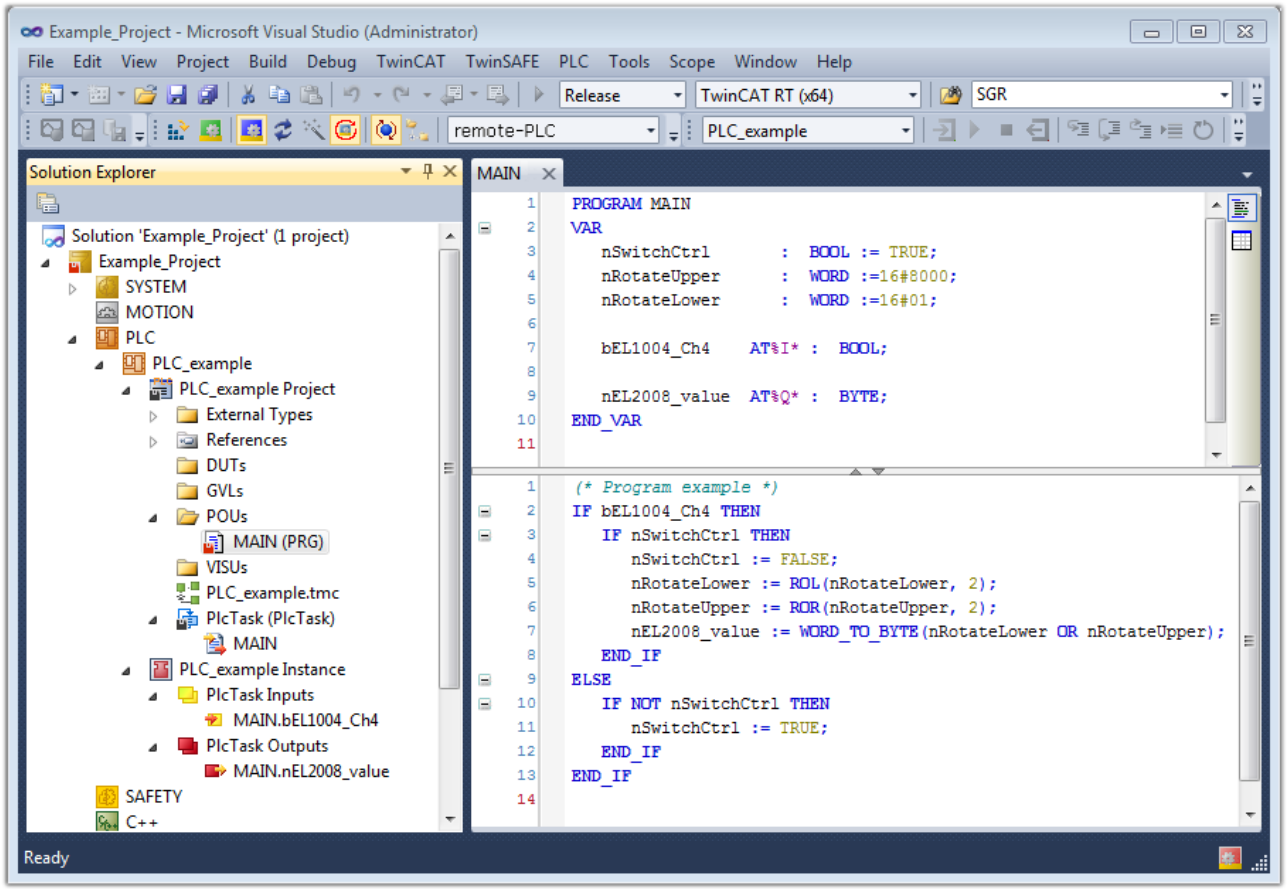


Fig. 52: Example program with variables after a compile process (without variable integration)

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

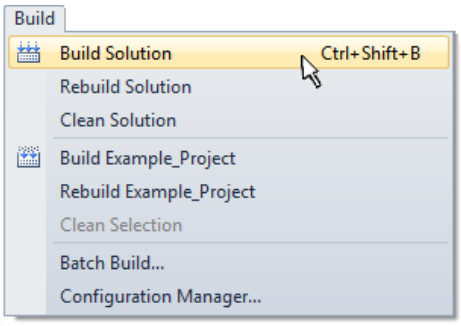
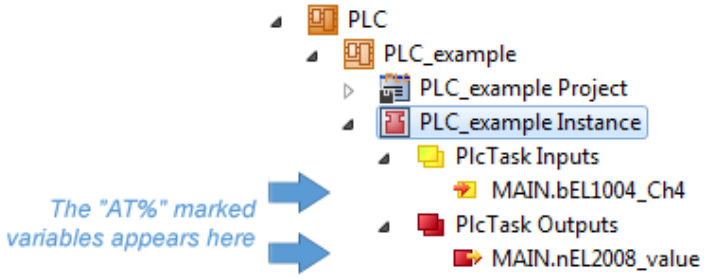


Fig. 53: Start program compilation

The following variables, identified in the ST/PLC program with “AT%”, are then available 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 to select a suitable process object (PDO) for linking:

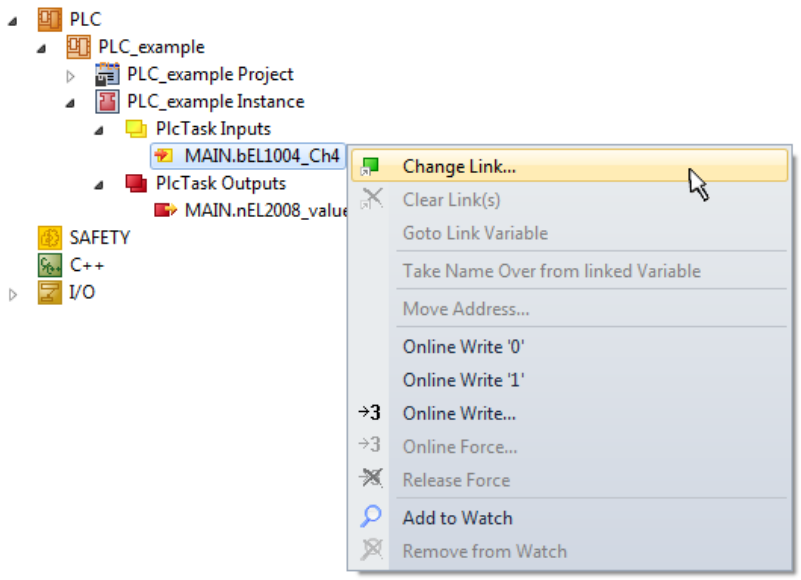


Fig. 54: Creating the links between PLC variables and process objects

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

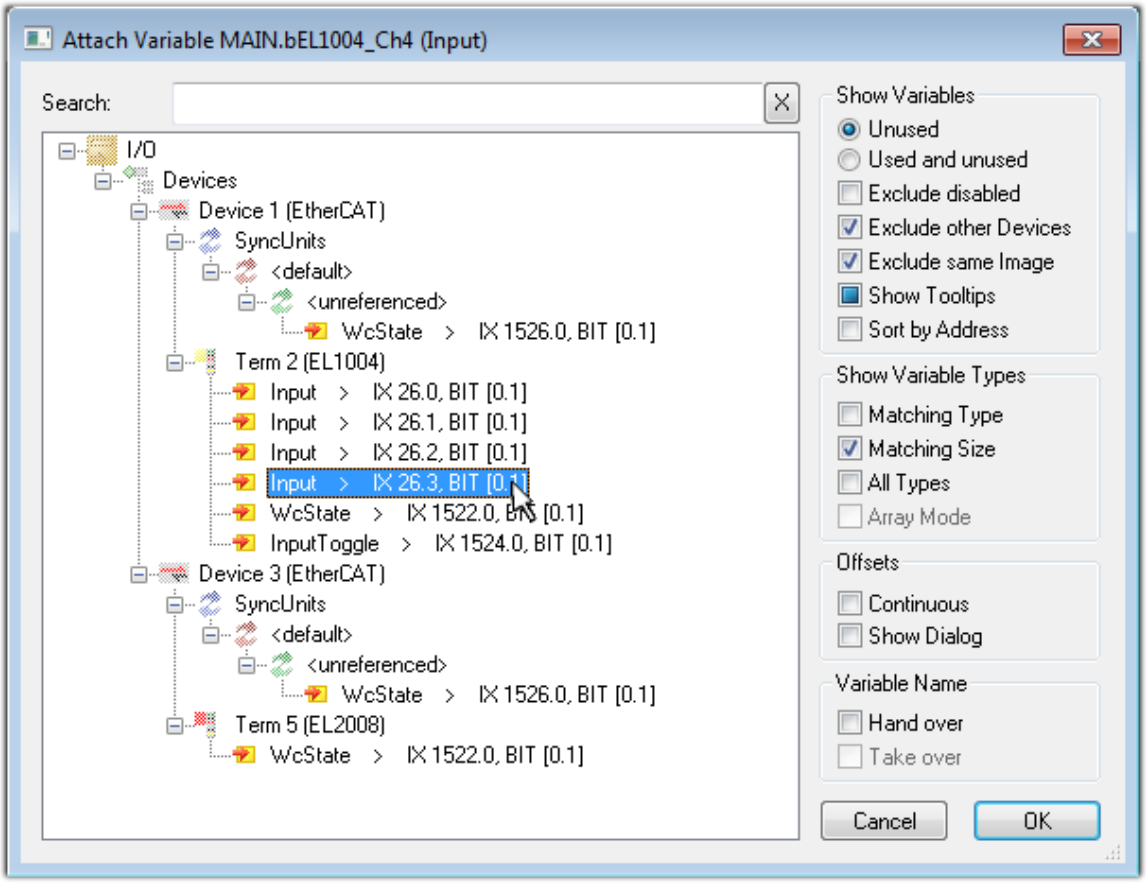


Fig. 55: Selecting BOOL-type PDO

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

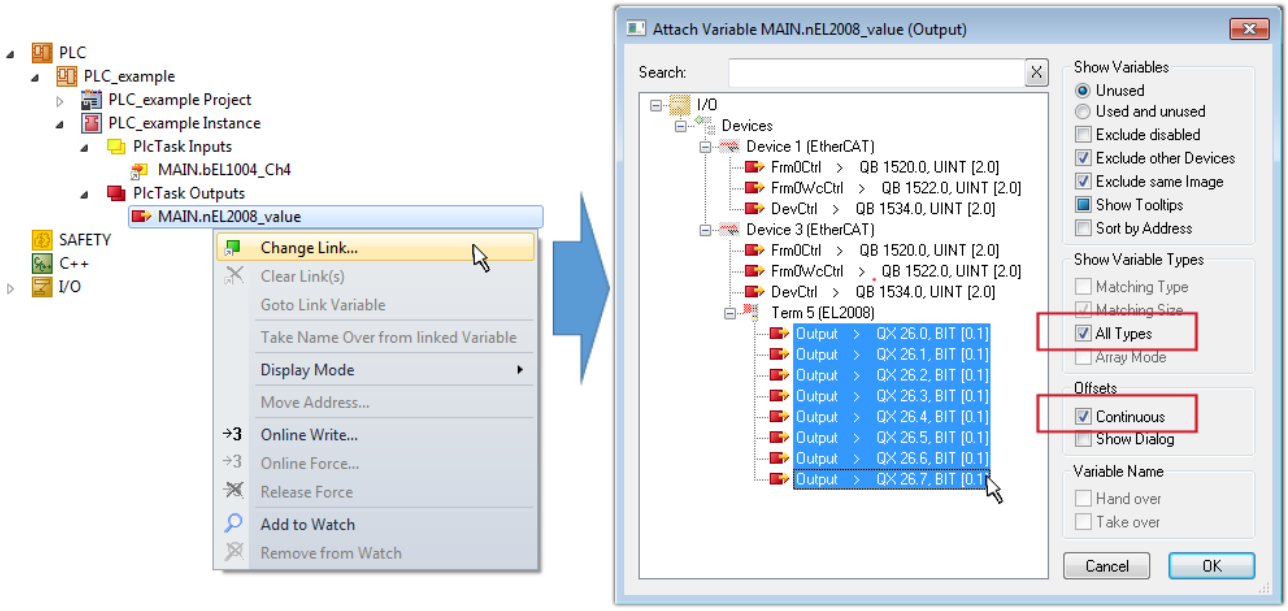


Fig. 56: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus 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 ( ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

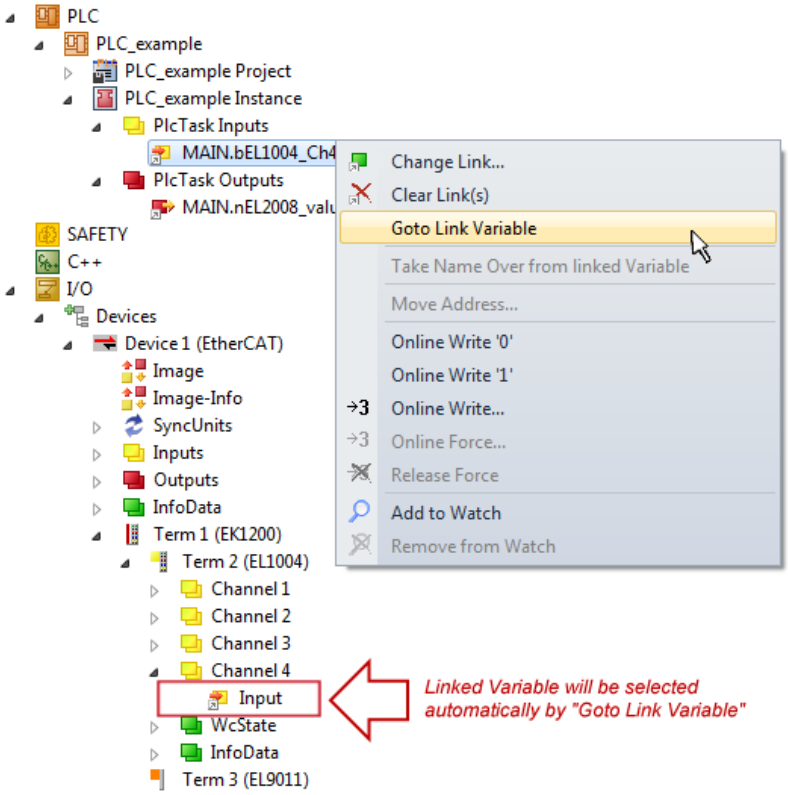


Fig. 57: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would 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 also possible to allocate this to a set of bit-standardized variables. Here, too, a “Goto Link Variable” can be executed in the other direction, so that the respective PLC instance can then be selected.

Note on 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, a structure can be created 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 “PLC” tab via the check box.
3. The data type in the “Data Type” field can then be copied using the “Copy” button.

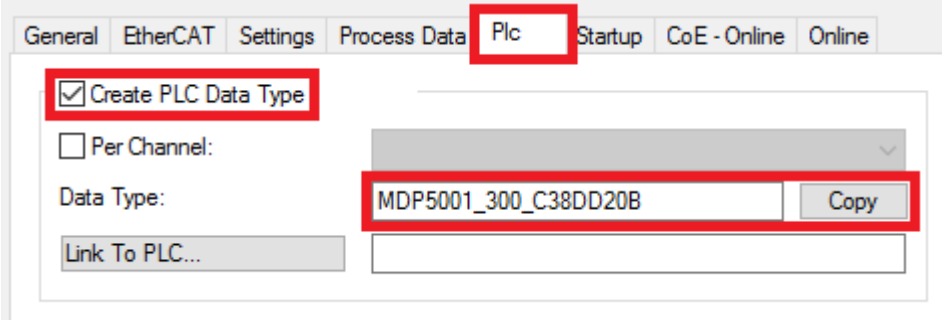


Fig. 58: Creating a PLC data type

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. 59: 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.

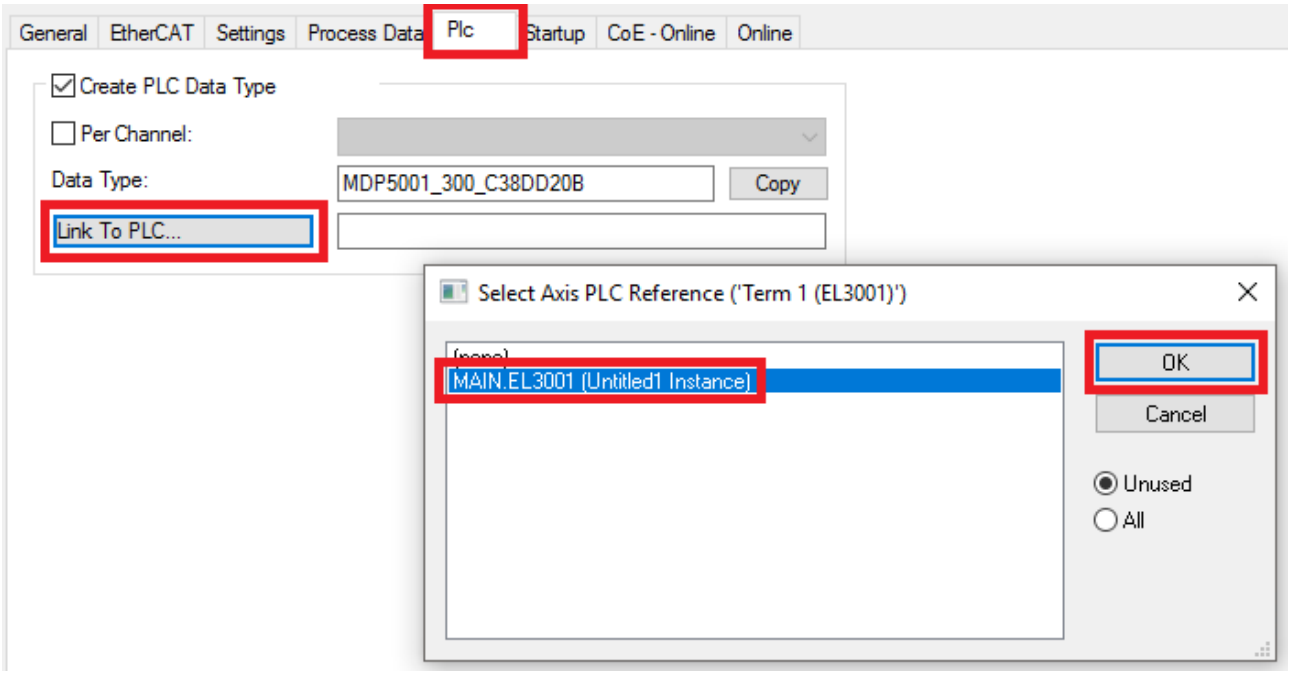


Fig. 60: Linking the structure

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

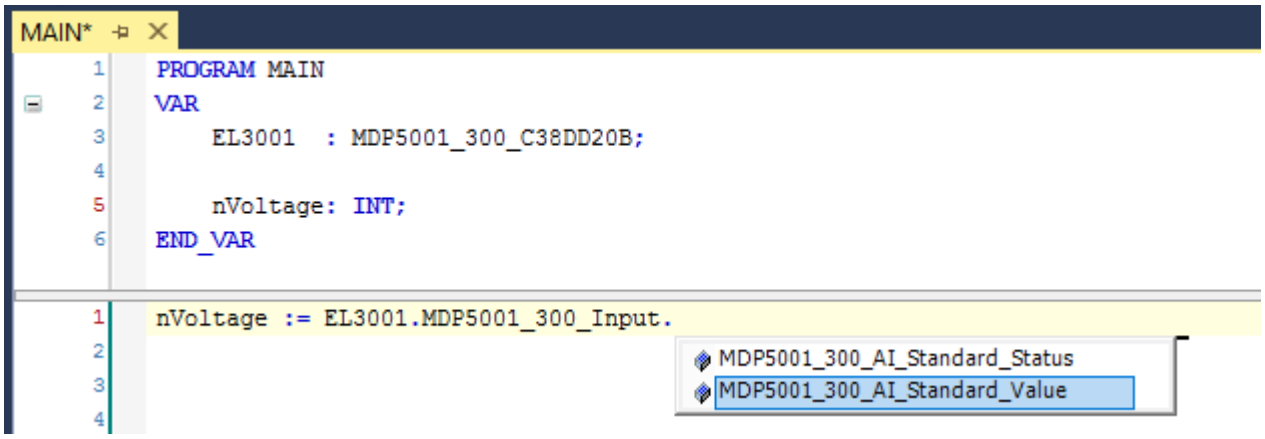
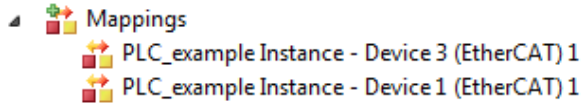


Fig. 61: Reading a variable from the structure of the process data

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 the settings of the development environment to the runtime system. Confirm the messages “Old configurations will be 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 the program to be started by clicking on symbol  , the “F5” key or via “PLC” in the menu, by selecting “Start”. The started programming environment shows the runtime values of individual variables:

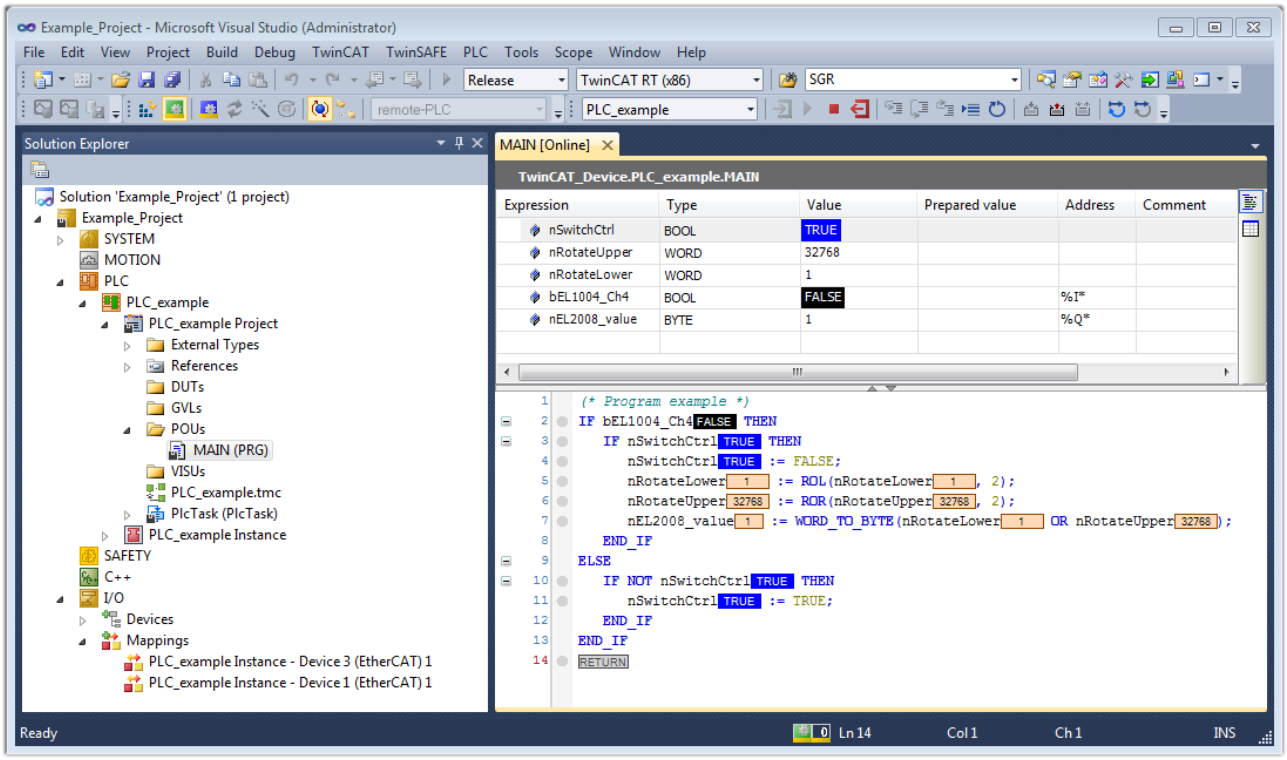


Fig. 62: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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

5.3 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- and 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.

5.3.1 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.

A: Via the TwinCAT Adapter dialog

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

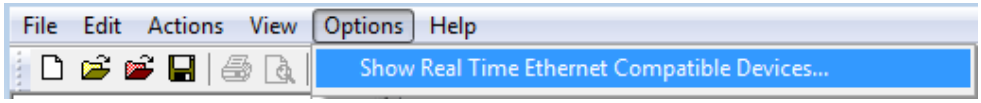


Fig. 63: System Manager “Options” (TwinCAT 2)

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

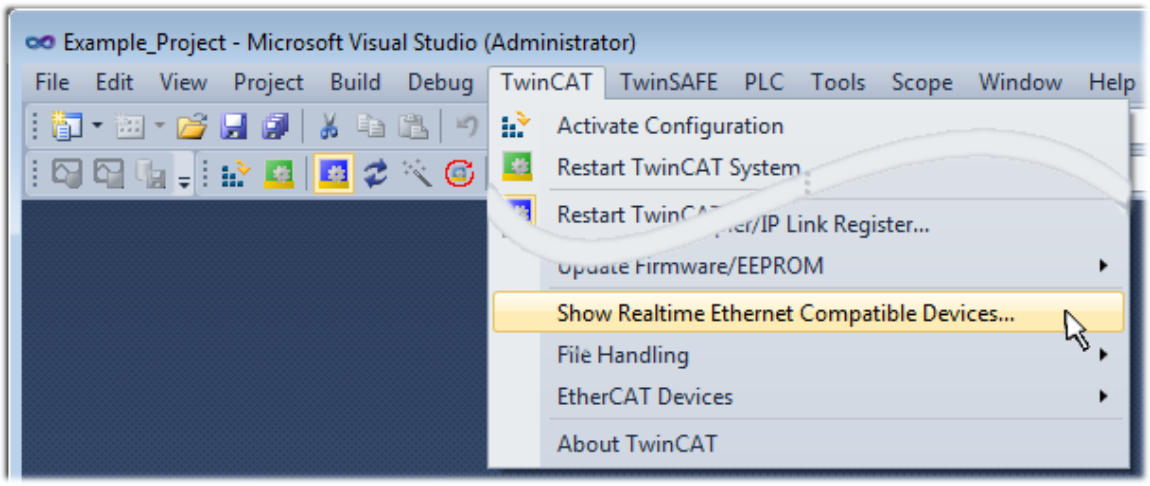


Fig. 64: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

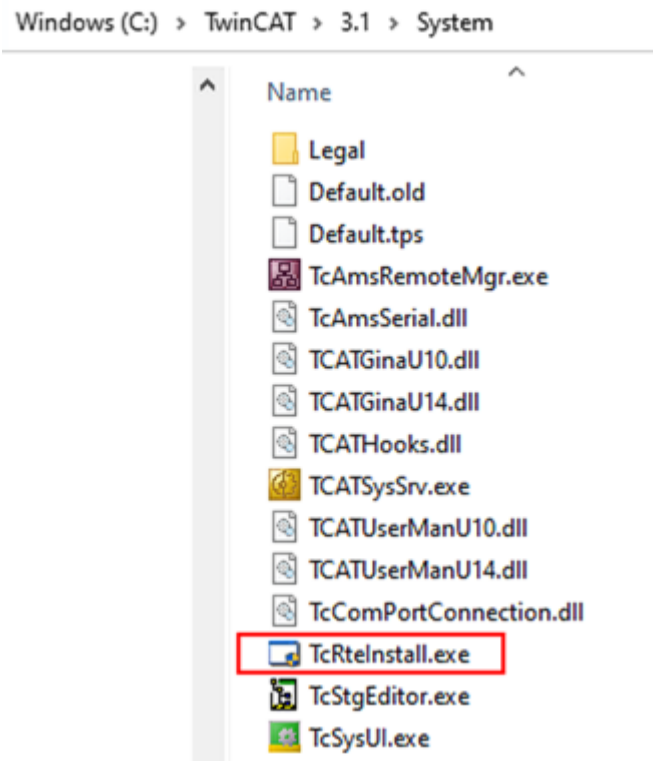


Fig. 65: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

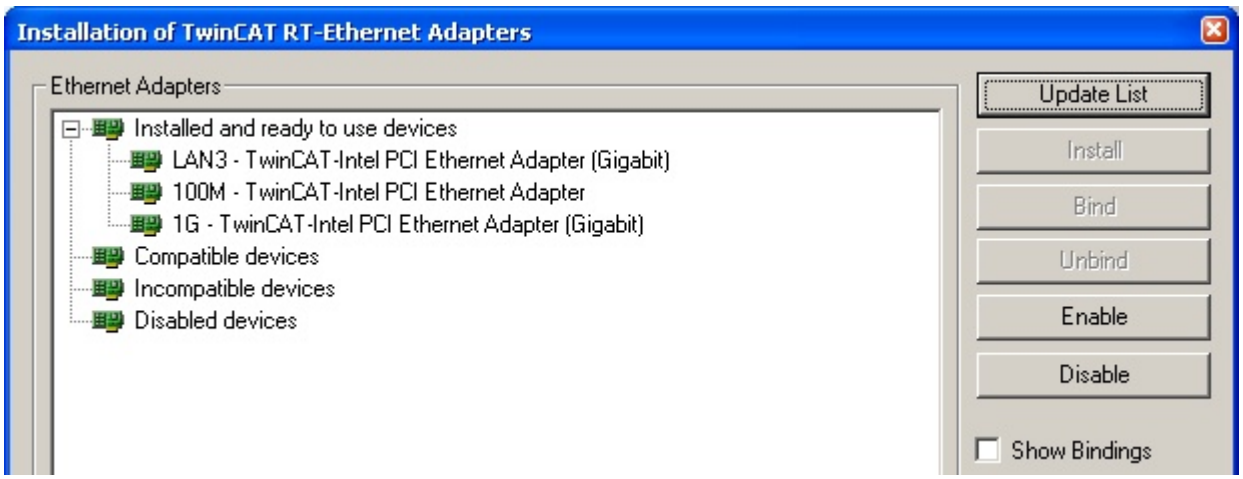


Fig. 66: Overview of network interfaces

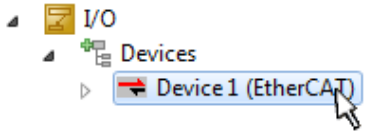
Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

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” [▶ 80] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):

Fig. 67: EtherCAT device properties (TwinCAT 2): click on “Compatible Devices…” of tab “Adapter”

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)

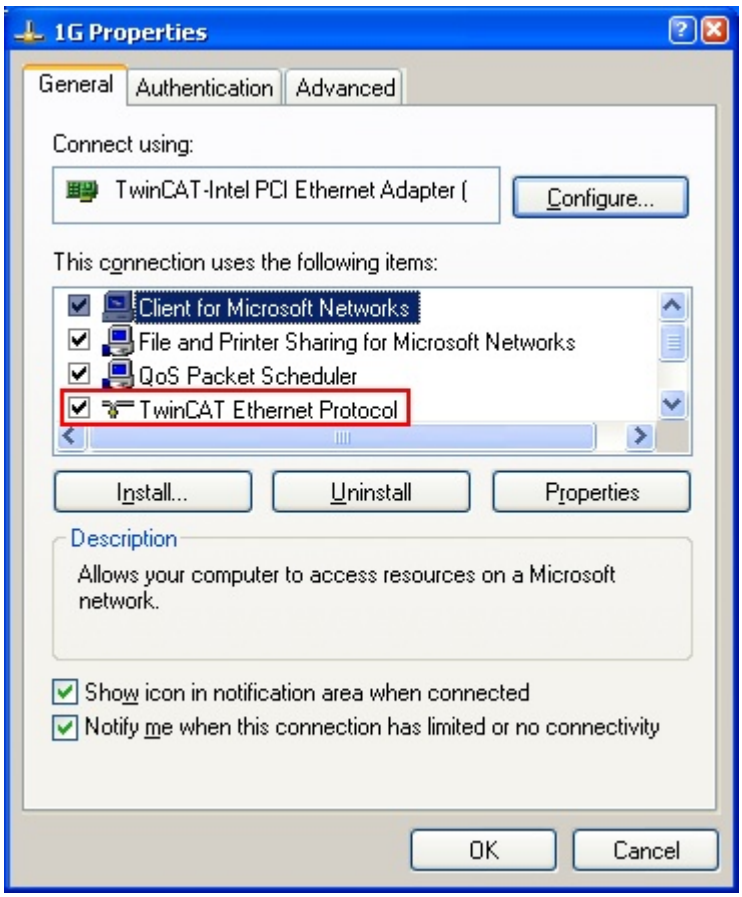


Fig. 68: Windows properties of the network interface

A correct setting of the driver could be:

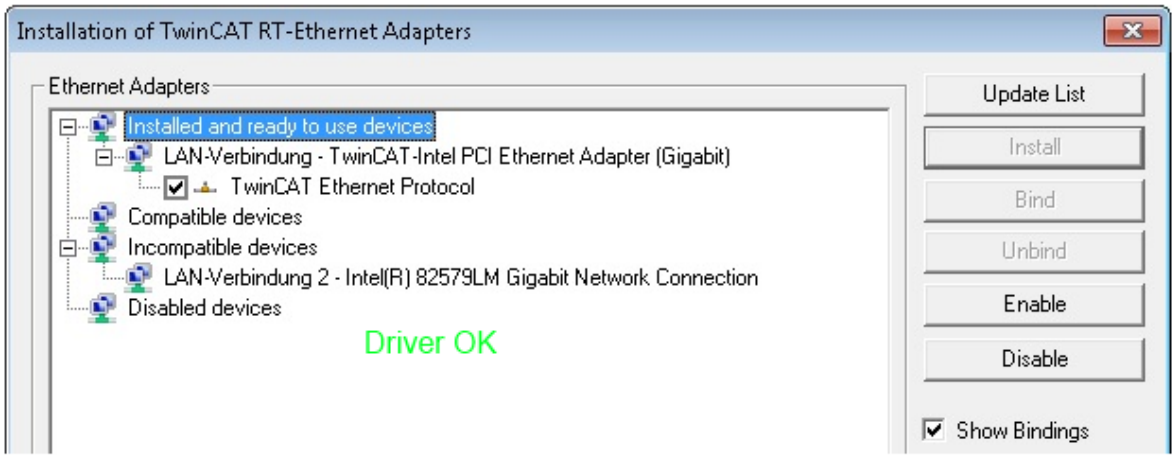


Fig. 69: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

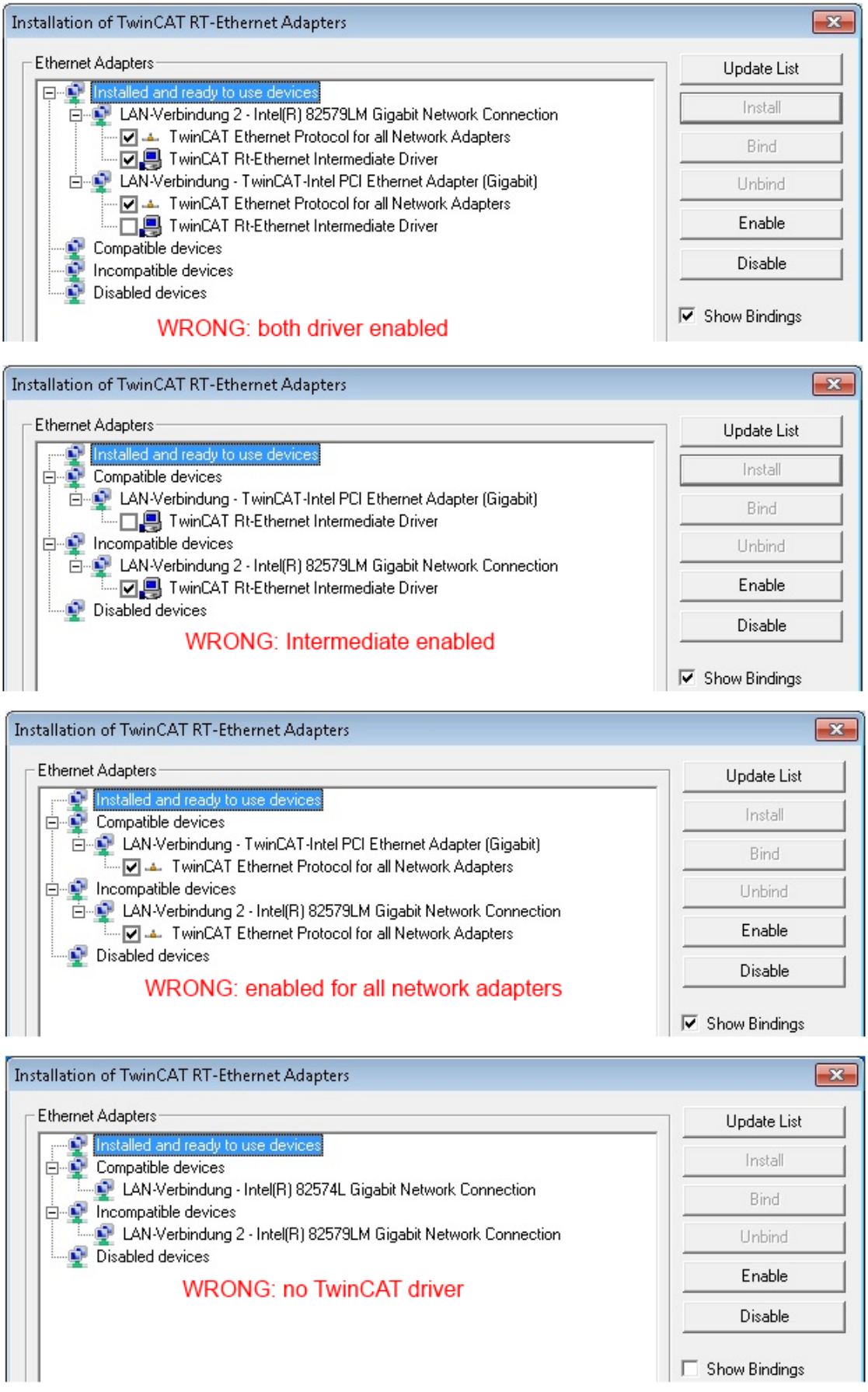


Fig. 70: Incorrect driver settings for the Ethernet port

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.

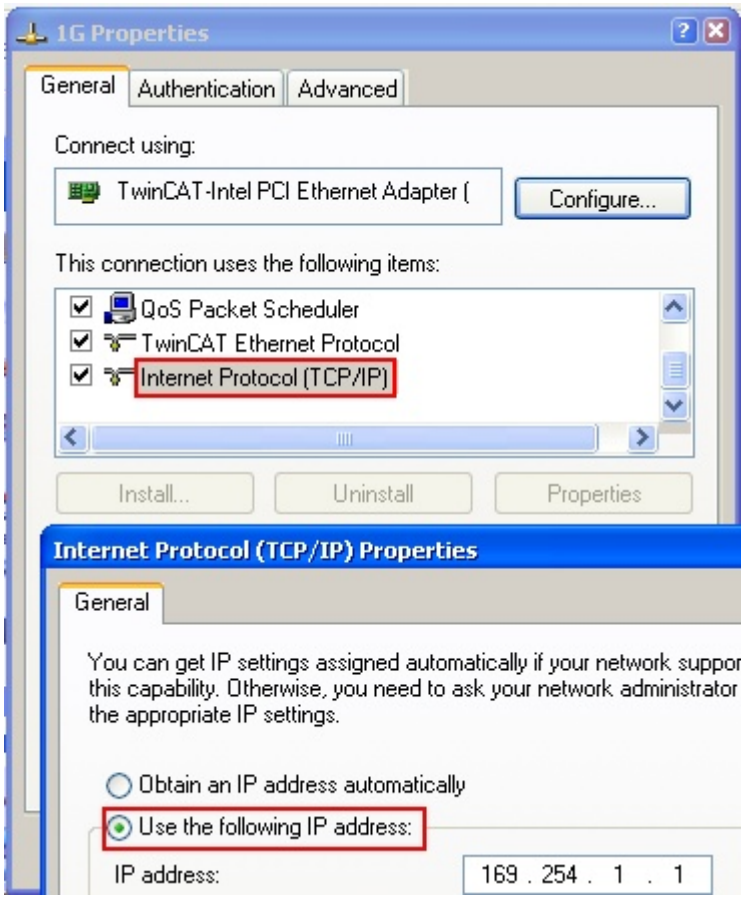


Fig. 71: TCP/IP setting for the Ethernet port

5.3.2 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 generate 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 manufacturer and are made available for download. An *.xml file may contain several device descriptions.

The ESI files for Beckhoff EtherCAT devices are available on the Beckhoff website.

The ESI files should 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.

For TwinCAT 2.11/TwinCAT 3 and higher, the ESI directory can be updated from the System Manager, if the programming PC is connected to the Internet; by

- **TwinCAT 2**: Option → “Update EtherCAT Device Descriptions”
- **TwinCAT 3**: TwinCAT → EtherCAT Devices → “Update Device Descriptions (via ETG Website)…”

The TwinCAT ESI Updater [▶ 79] 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”

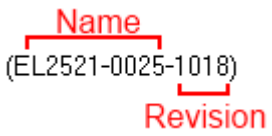


Fig. 72: Identifier structure

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 [▶ 7].

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.

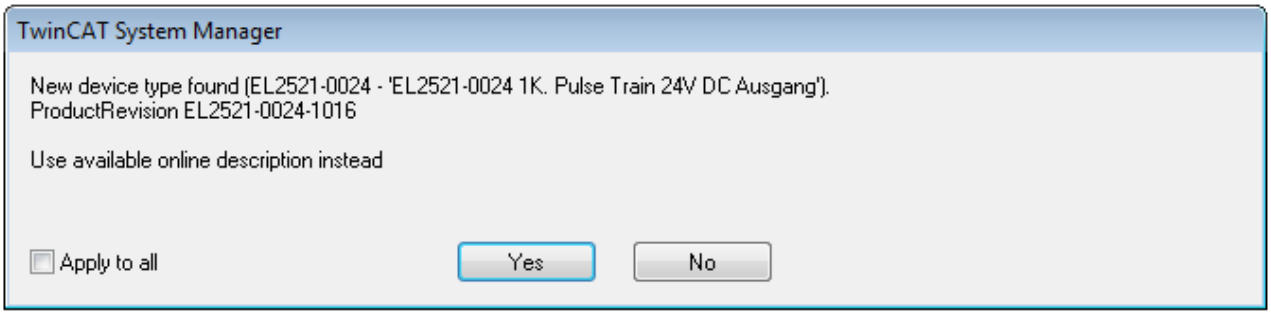


Fig. 73: OnlineDescription information window (TwinCAT 2)

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

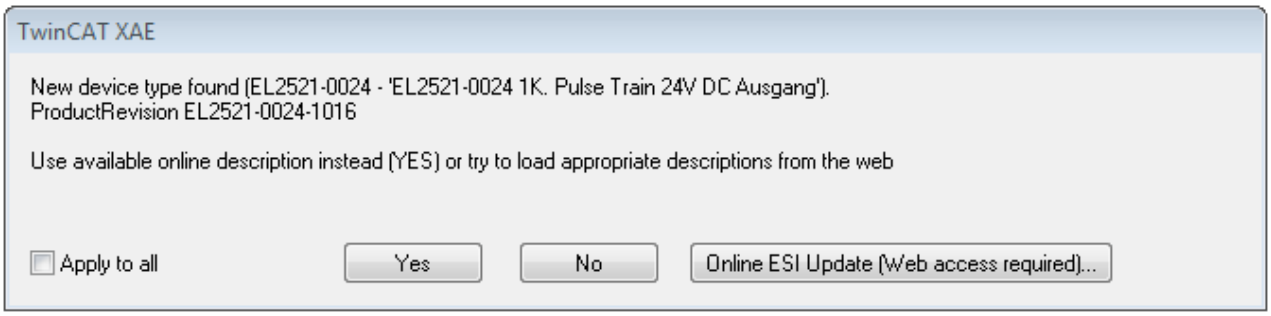


Fig. 74: Information window OnlineDescription (TwinCAT 3)

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.

NOTE

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 [▶ 80]".

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.

OnlineDescriptionCache00000002.xml

Fig. 75: 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)*.

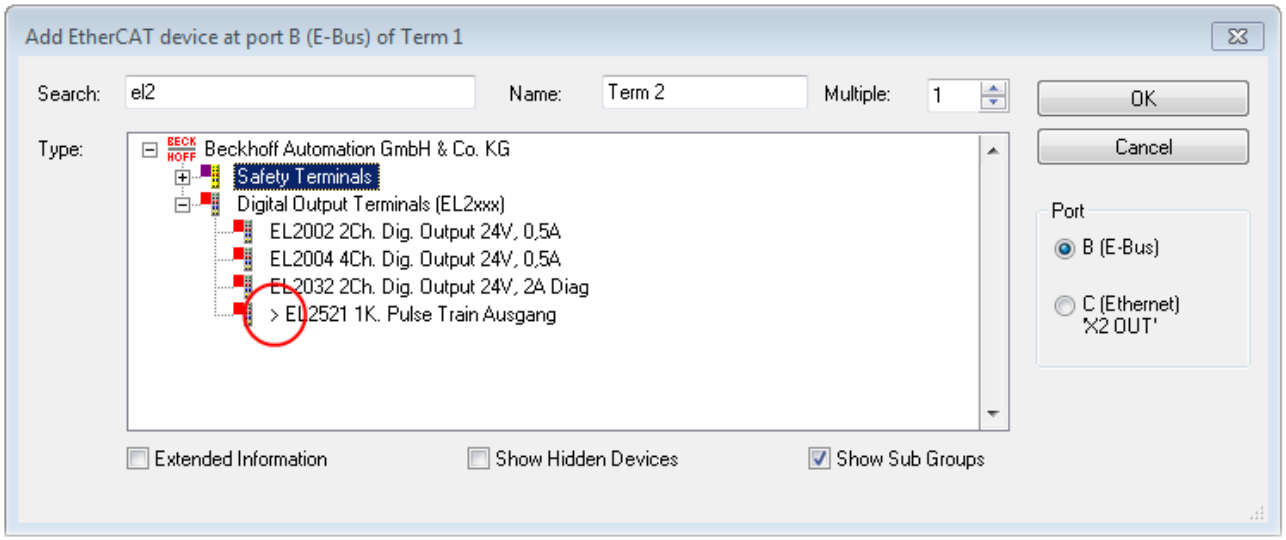


Fig. 76: Indication of an online recorded ESI of EL2521 as an example

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.

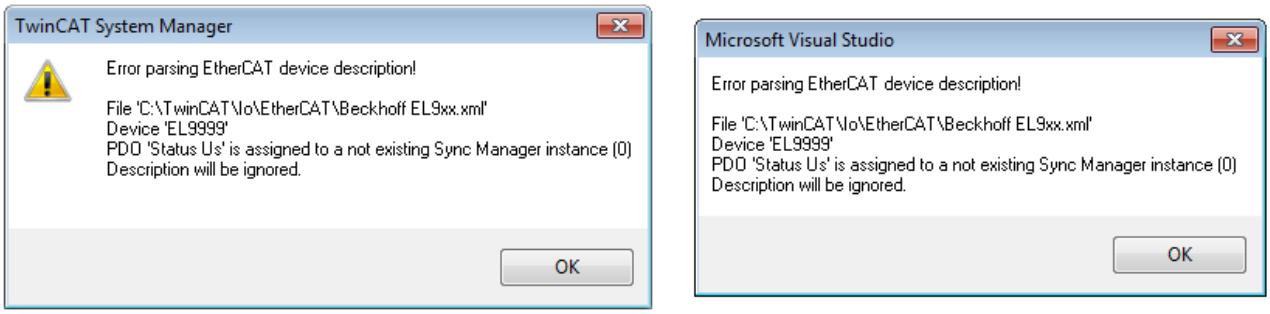


Fig. 77: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

5.3.3 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:

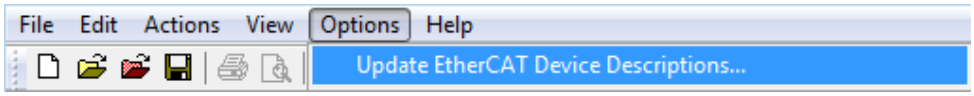


Fig. 78: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

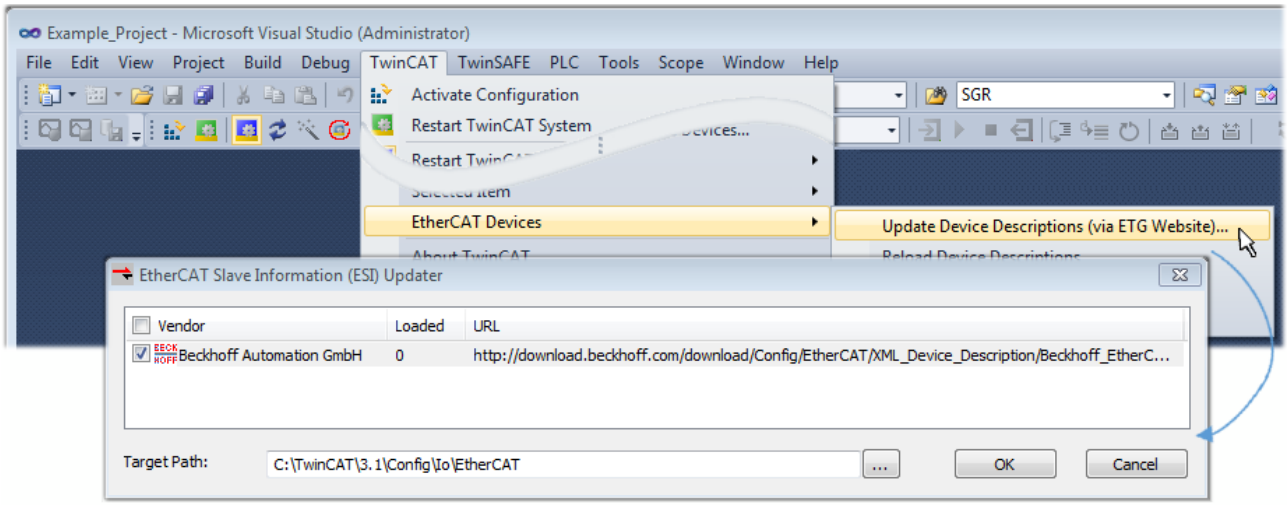


Fig. 79: Using the ESI Updater (TwinCAT 3)

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)…”.

5.3.4 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” [▶ 75].

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 [▶ 85] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 86]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 89]

The scan with existing configuration [▶ 90] can also be carried out for comparison.

5.3.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

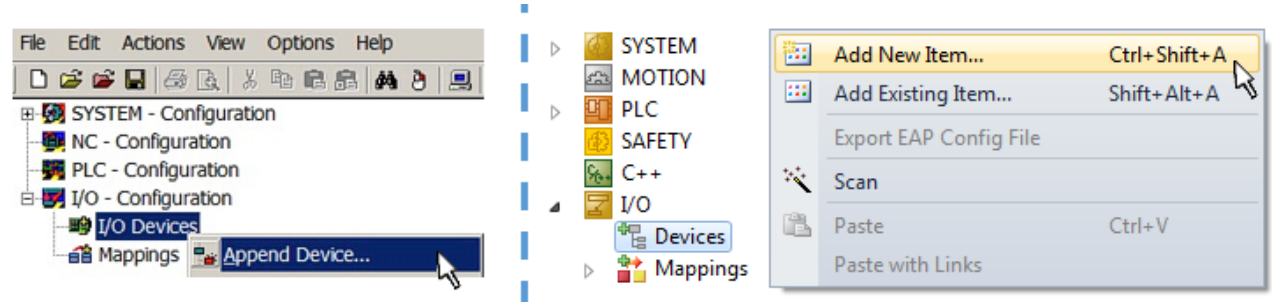


Fig. 80: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

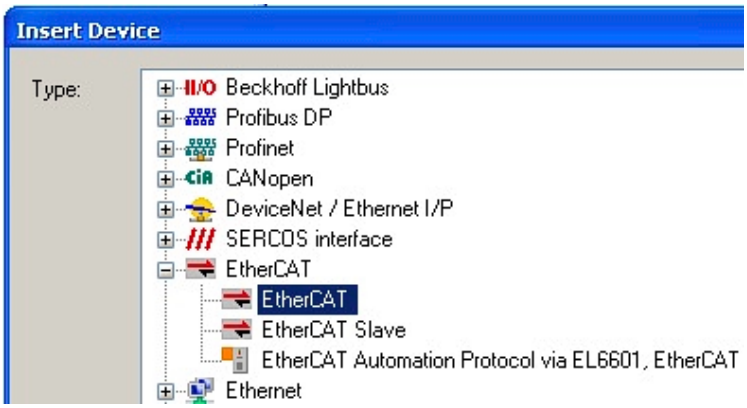


Fig. 81: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

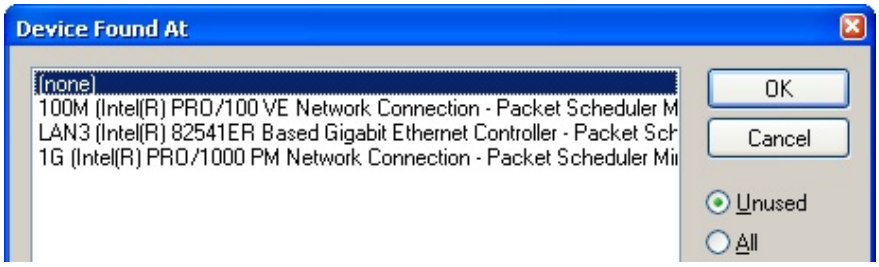


Fig. 82: Selecting the Ethernet port

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)”.

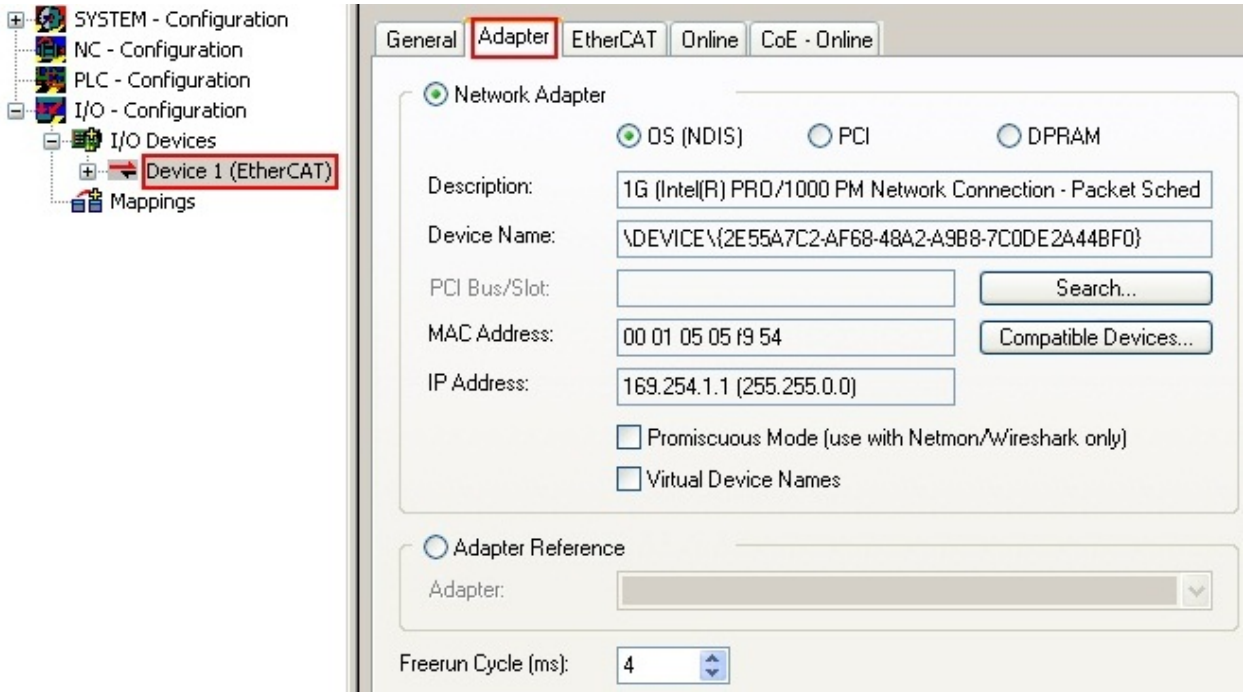
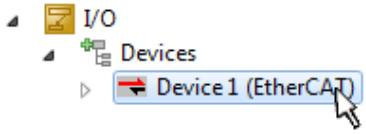


Fig. 83: EtherCAT device properties (TwinCAT 2)

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 [▶ 69].

Defining EtherCAT slaves

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

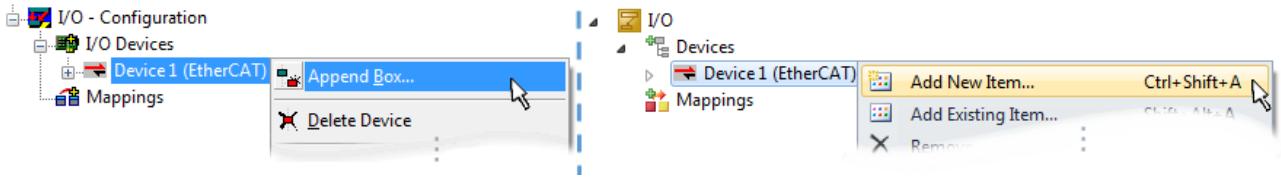


Fig. 84: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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: couplers, box modules, devices with RJ45/M8/M12 connector

- “E-Bus”: LVDS “terminal bus”, EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

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

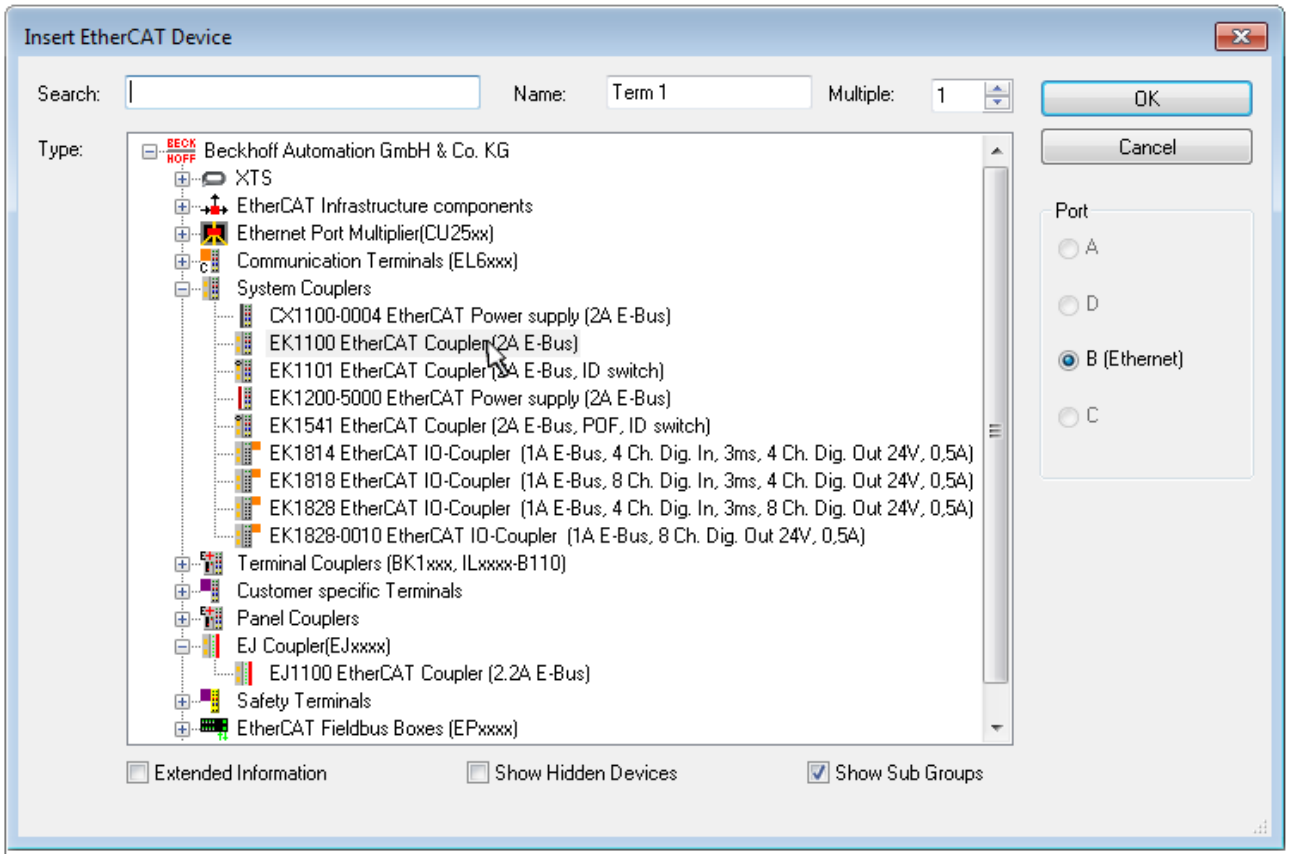


Fig. 85: Selection dialog for new EtherCAT device

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”.

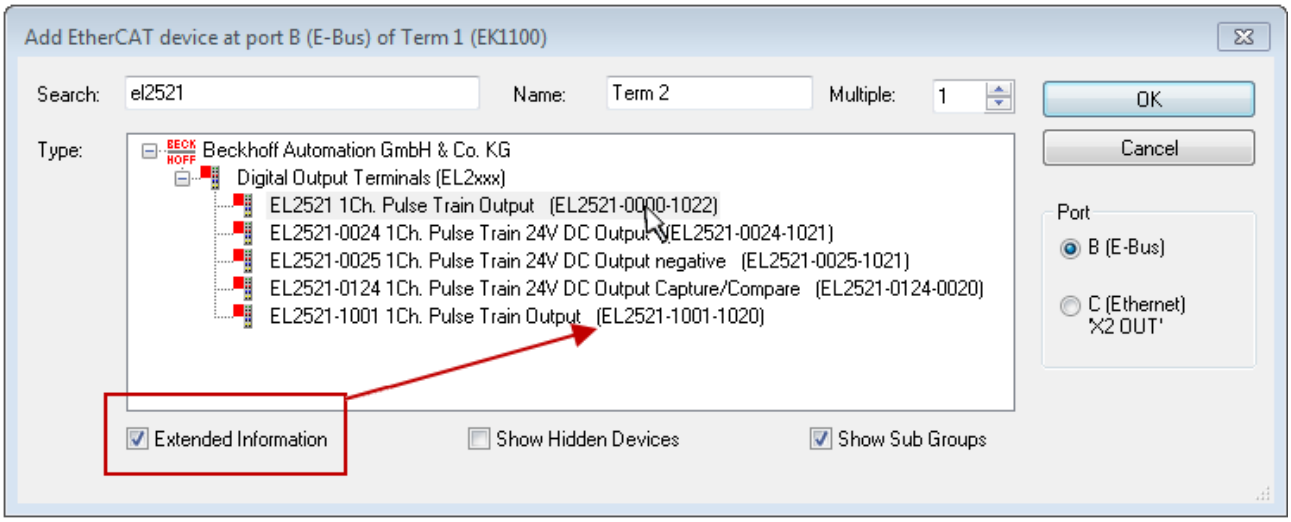


Fig. 86: Display of device revision

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”.

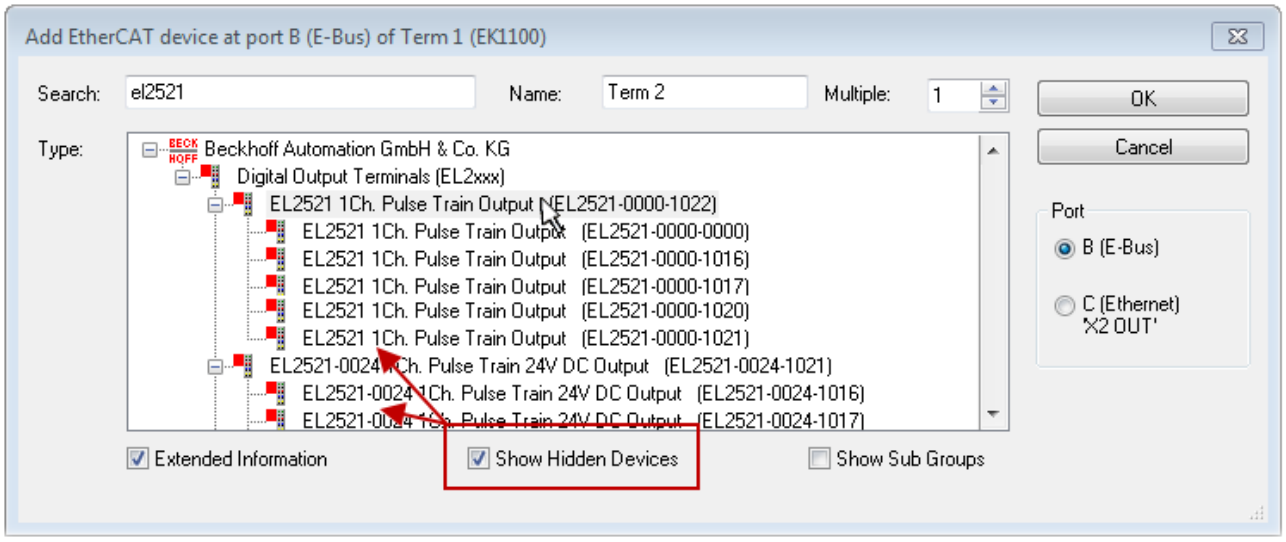


Fig. 87: Display of previous revisions

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.

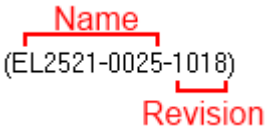


Fig. 88: Name/revision of the terminal

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, ...

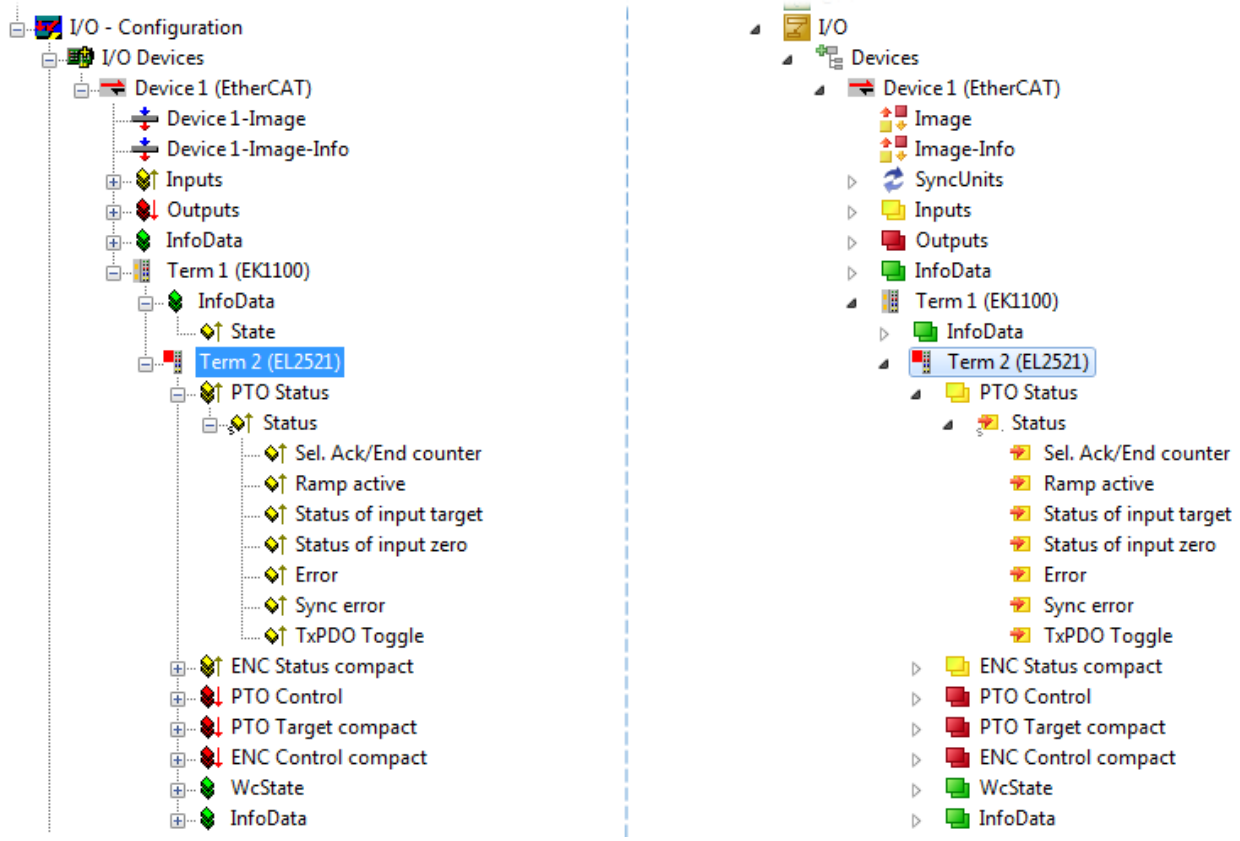


Fig. 89: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.3.6 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.

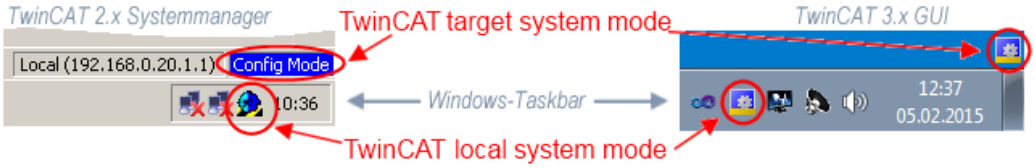


Fig. 90: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

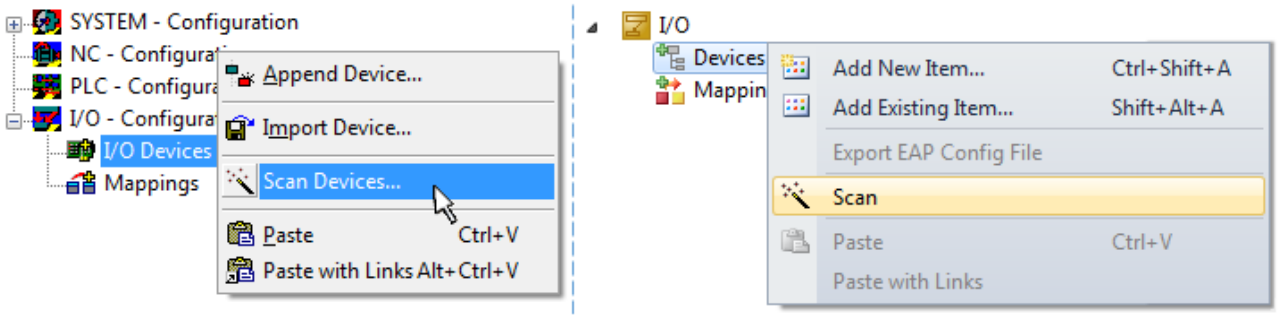


Fig. 91: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

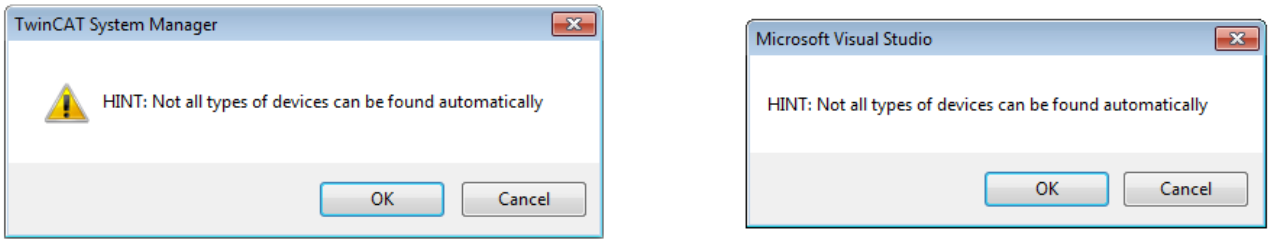


Fig. 92: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

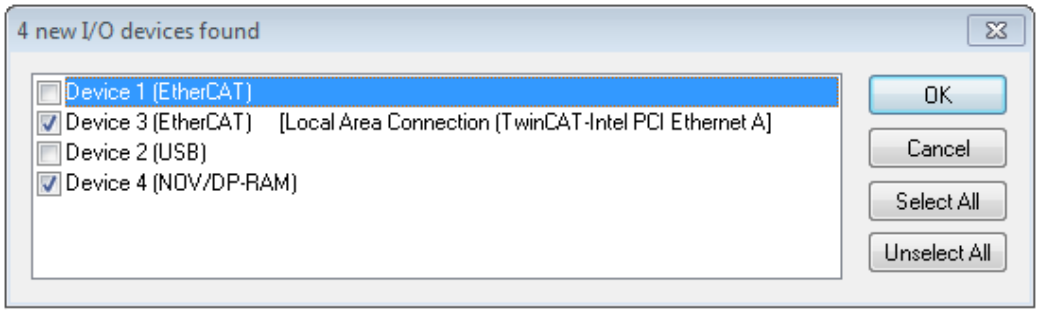


Fig. 93: Detected Ethernet devices

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 [▶ 69].

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.

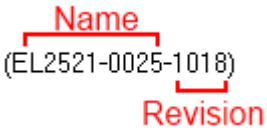


Fig. 94: Example default state

NOTE

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 [▶ 90] 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:

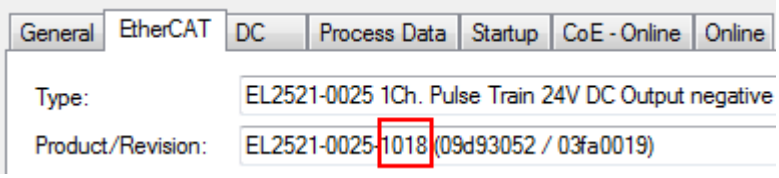


Fig. 95: Installing EthetCAT terminal with revision -1018

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 [▶ 90] 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:

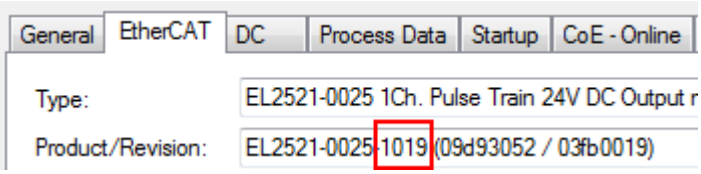


Fig. 96: Detection of EtherCAT terminal with revision -1019

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

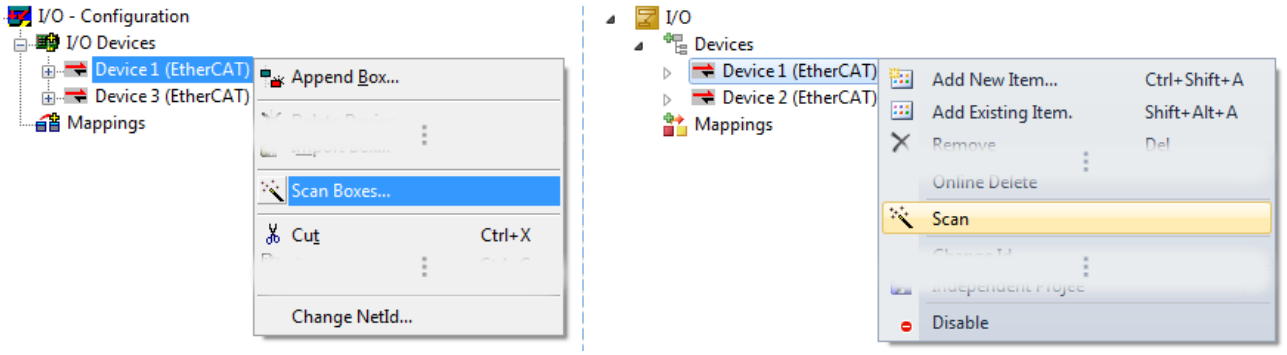


Fig. 98: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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.

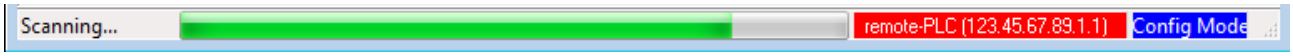


Fig. 99: *Scan progressexemplary by TwinCAT 2*

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

Fig. 100: 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).

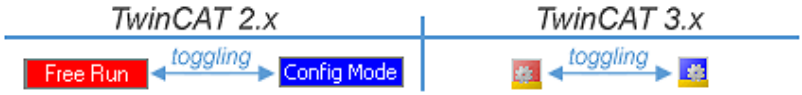


Fig. 101: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

Fig. 102: 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*.

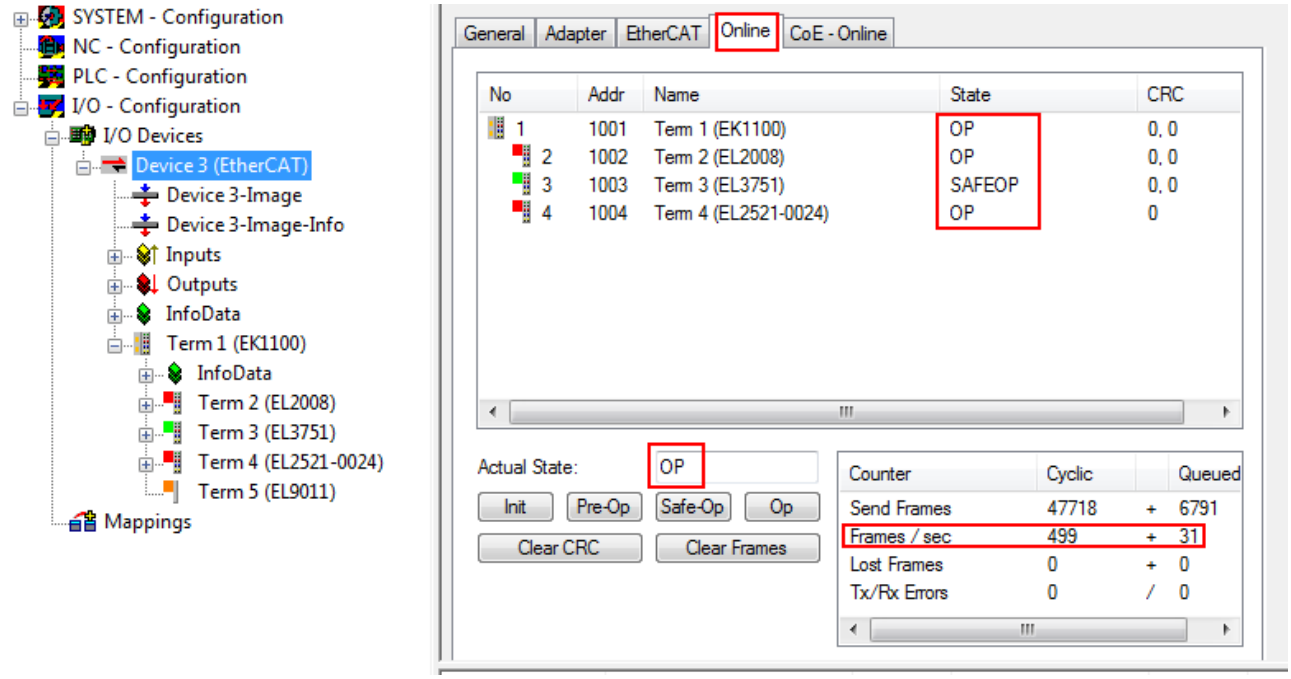


Fig. 103: Online display example

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 [▶ 80].

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.

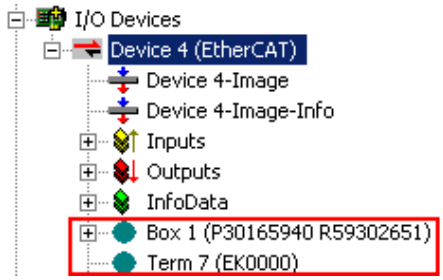


Fig. 104: Faulty identification

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

NOTE

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. 105: 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.

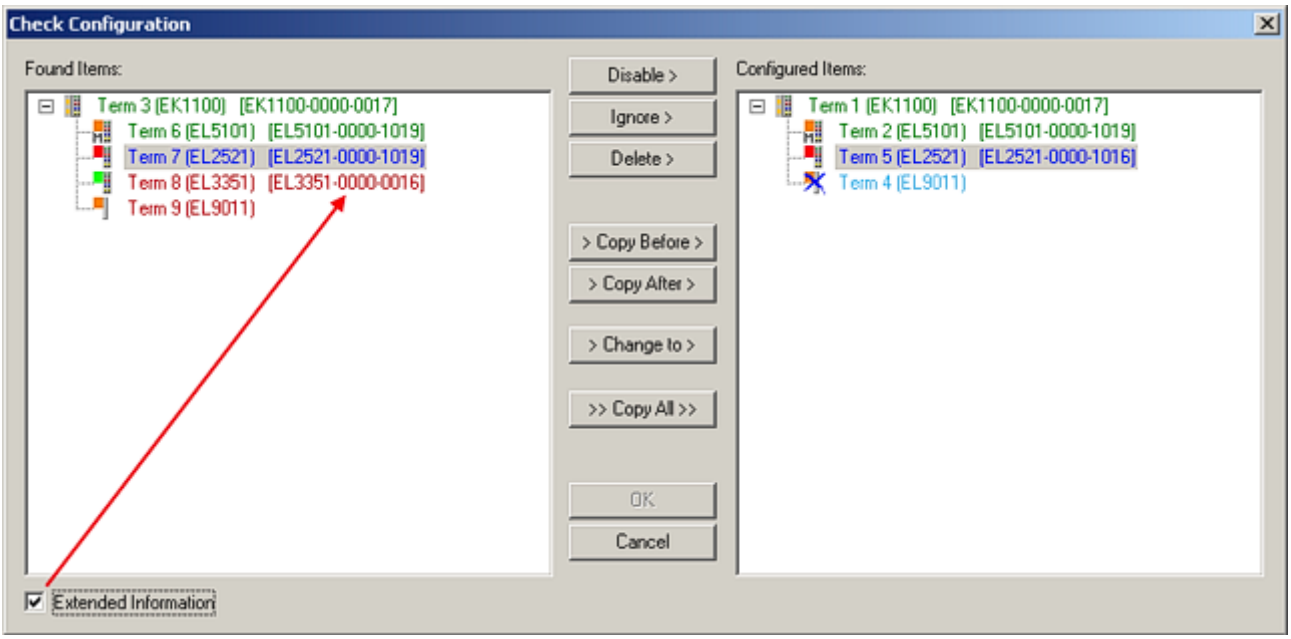


Fig. 106: Correction dialog

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)

Color	Explanation
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.

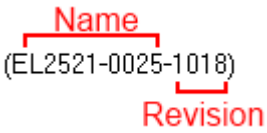


Fig. 107: Name/revision of the terminal

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, ...

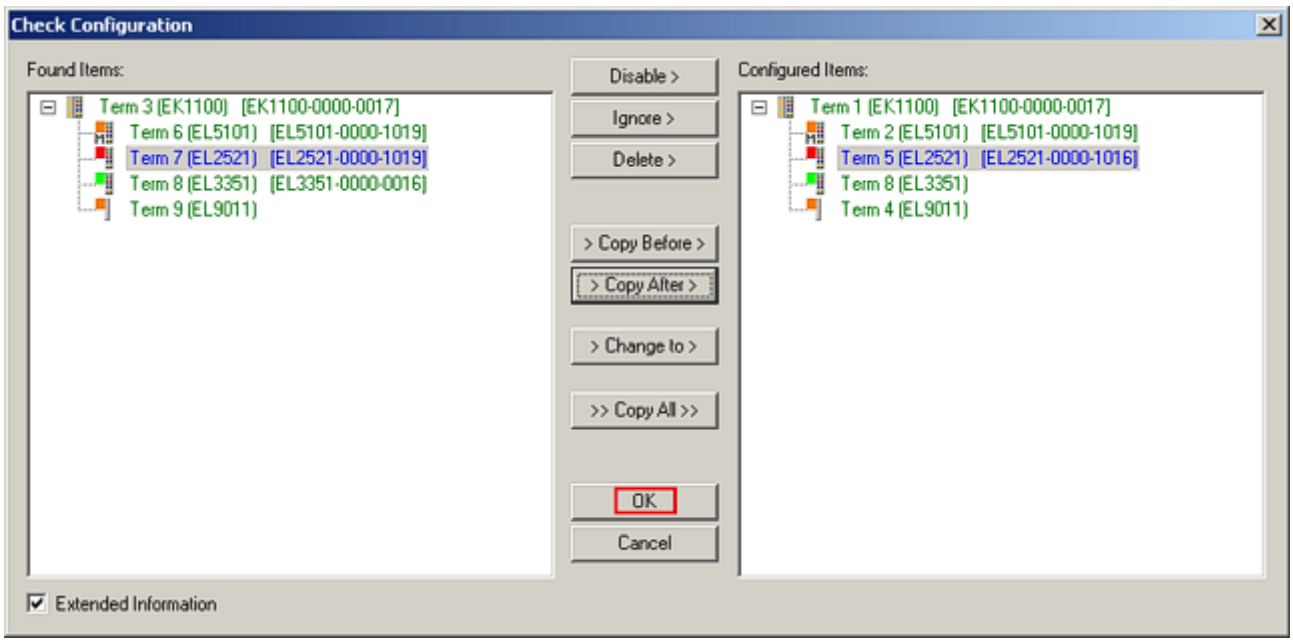


Fig. 108: Correction dialog with modifications

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.

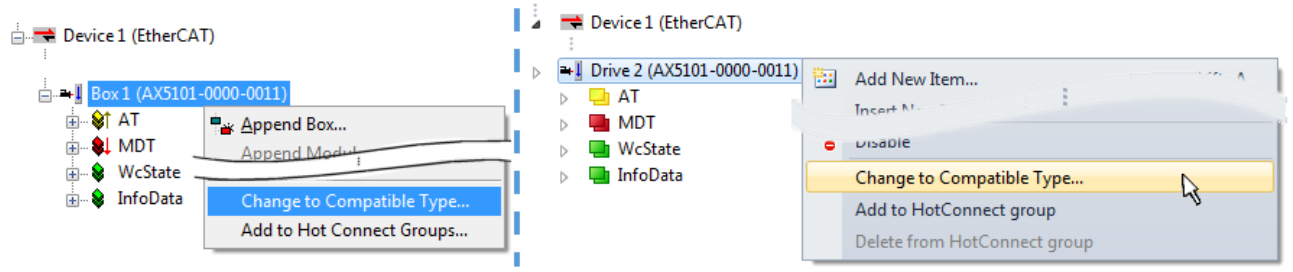


Fig. 109: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

The following elements in the ESI of an EtherCAT device are compared by TwinCAT and assumed to be the same in order to decide whether a device is indicated as "compatible":

- Physics (e.g. RJ45, Ebus...)

- FMMU (additional ones are allowed)

- SyncManager (SM, additional ones are allowed)

- EoE (attributes MAC, IP)

- CoE (attributes SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)

- FoE

- PDO (process data: Sequence, SyncUnit SU, SyncManager SM, EntryCount, Ent-ry.Datatype)

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

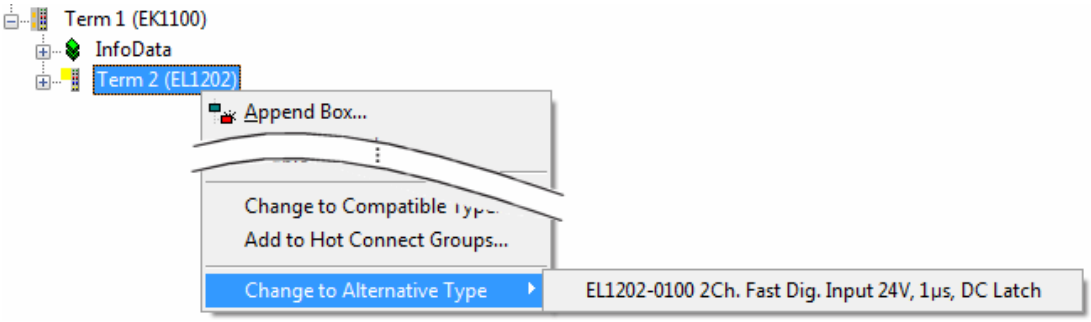


Fig. 110: TwinCAT 2 Dialog Change to Alternative Type

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).

5.3.7 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).

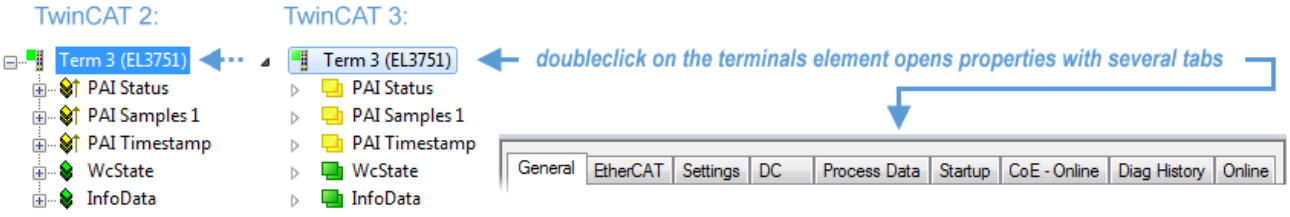


Fig. 111: Branch element as terminal EL3751

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

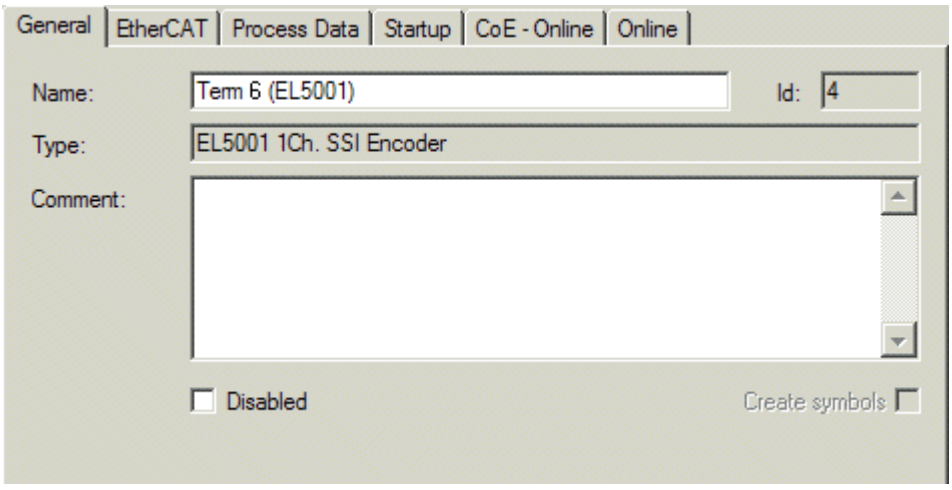


Fig. 112: “General” tab

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

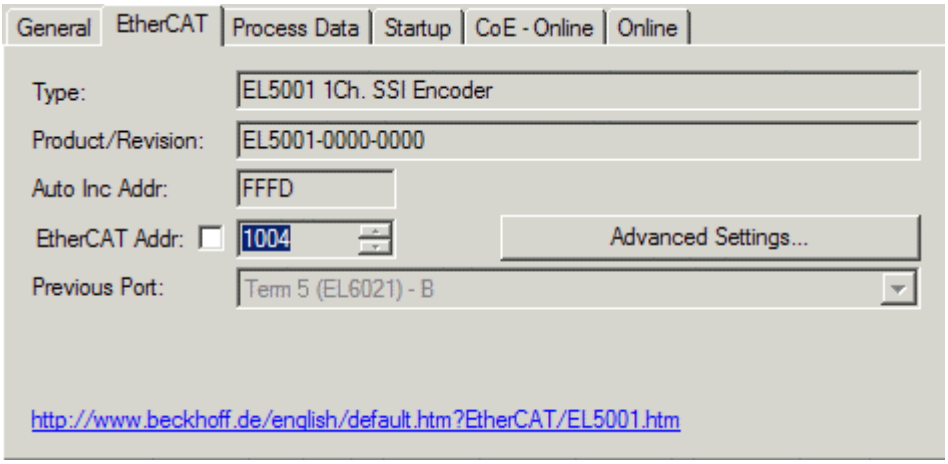


Fig. 113: "EtherCAT" tab

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.

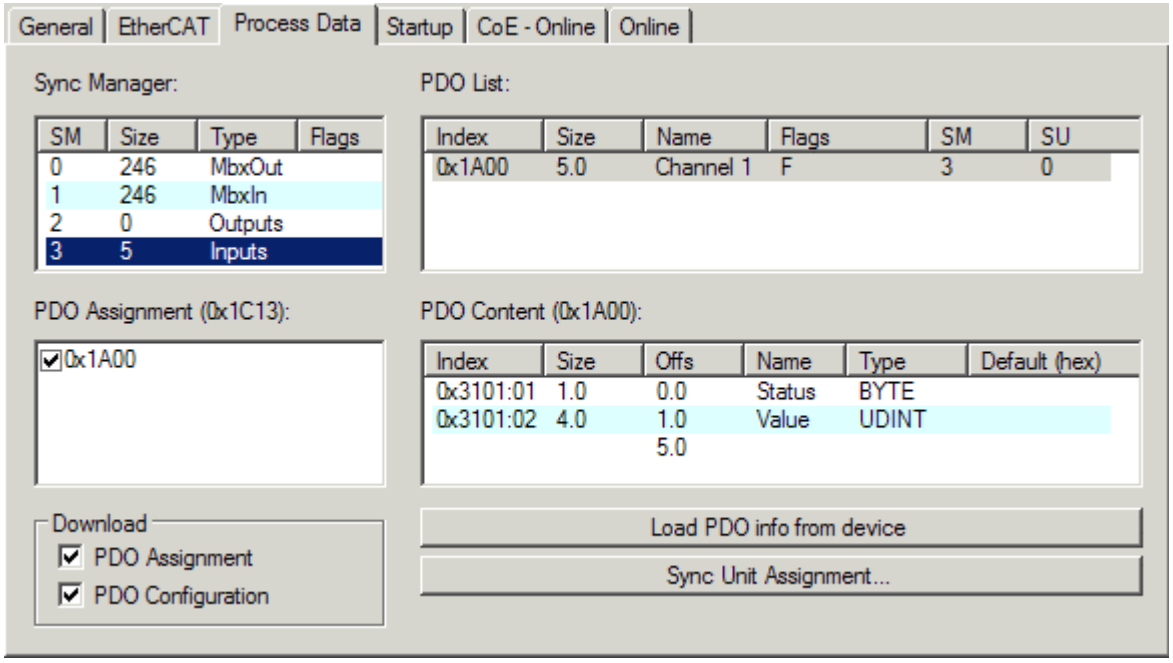


Fig. 114: “Process Data” tab

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”).

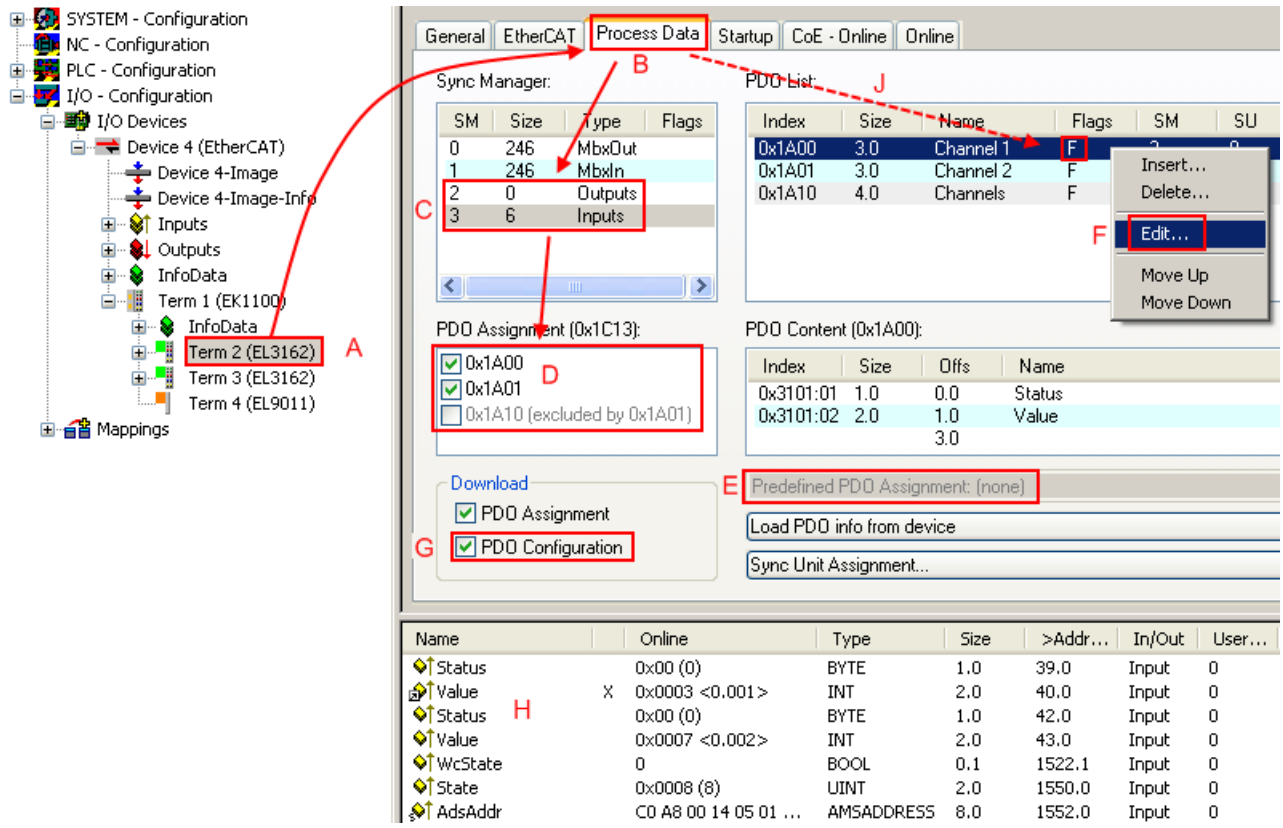


Fig. 115: Configuring the process data

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 [▶ 101] 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.

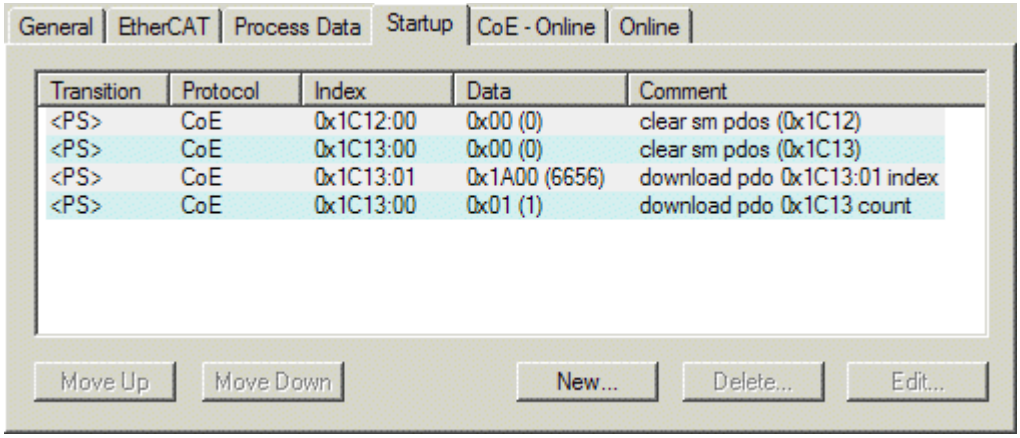


Fig. 116: “Startup” tab

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.

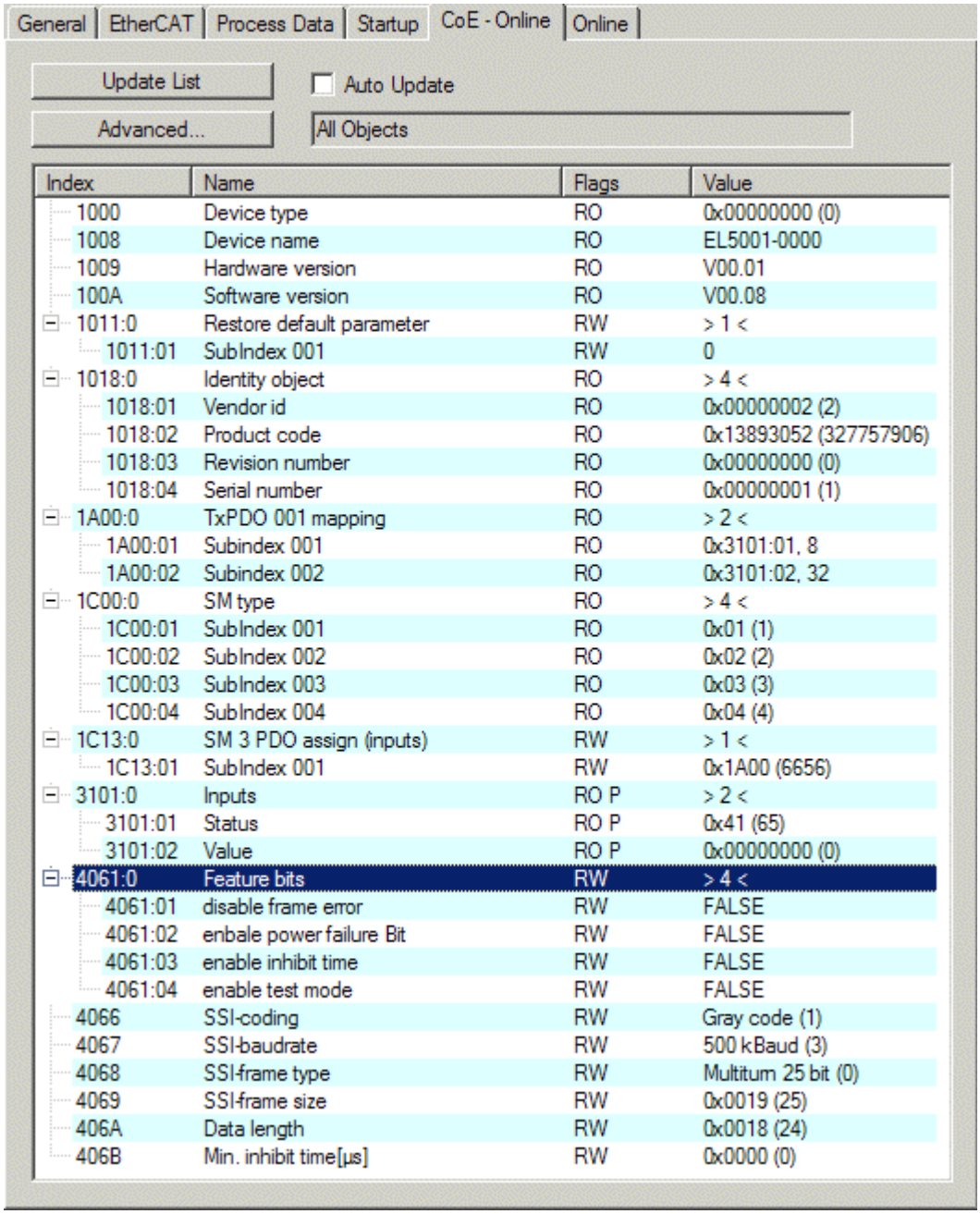


Fig. 117: “CoE - Online” tab

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.

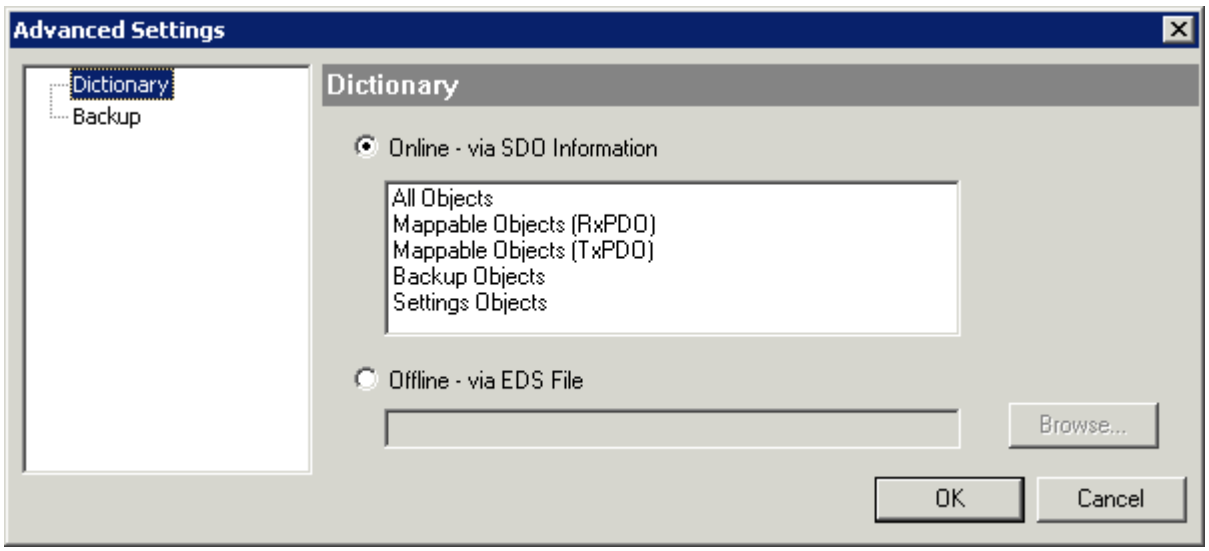


Fig. 118: Dialog “Advanced settings”

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

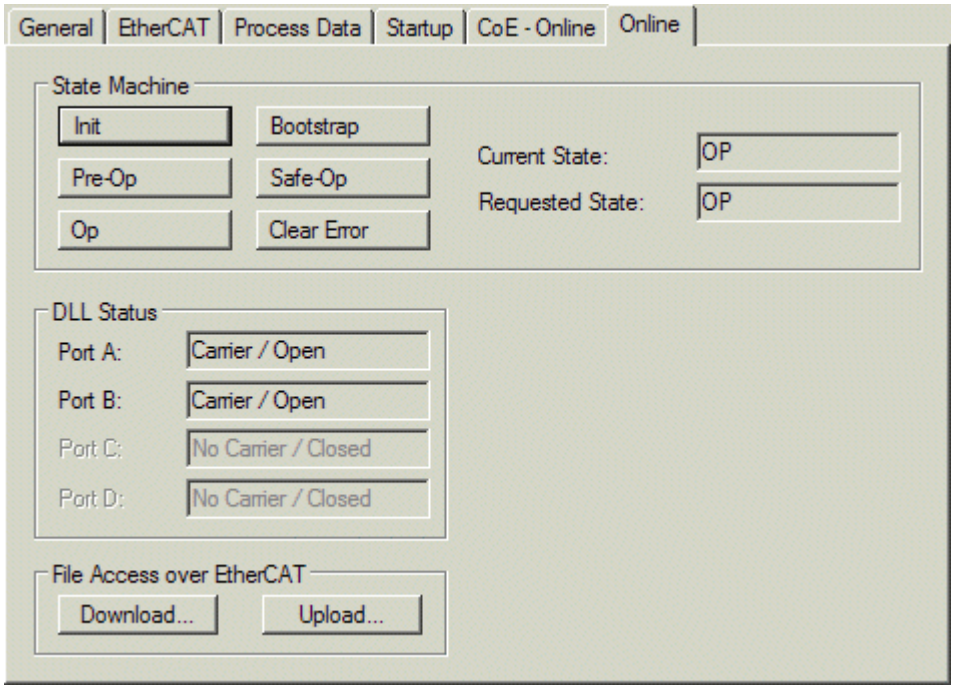


Fig. 119: “Online” tab

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)

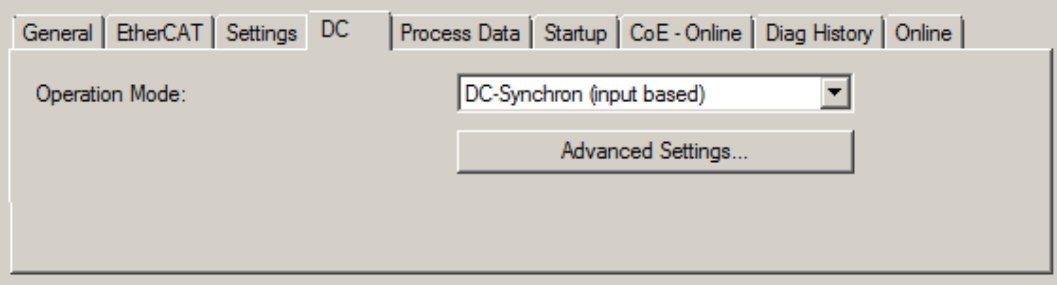


Fig. 120: "DC" tab (Distributed Clocks)

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

5.3.7.1 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 [▶ 99]),

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 [▶ 96] 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.

5.3.8 Import/Export of EtherCAT devices with SCI and XTI

SCI and XTI Export/Import – Handling of user-defined modified EtherCAT slaves

5.3.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

- Cyclic process data (PDO)
- Synchronization (Distributed Clocks, FreeRun, SM-Synchron)
- CoE parameters (acyclic object dictionary)

Note: Not all three elements may be present, depending on the slave.

For a better understanding of the export/import function, let's consider the usual procedure for IO configuration:

- The user/programmer processes the IO configuration in the TwinCAT system environment. This involves all input/output devices such as drives that are connected to the fieldbuses used.
  Note: In the following sections, only EtherCAT configurations in the TwinCAT system environment are considered.
- For example, the user manually adds devices to a configuration or performs a scan on the online system.
- This results in the IO system configuration.
- On insertion, the slave appears in the system configuration in the default configuration provided by the vendor, consisting of default PDO, default synchronization method and CoE StartUp parameter as defined in the ESI (XML device description).
- If necessary, elements of the slave configuration can be changed, e.g. the PDO configuration or the synchronization method, based on the respective device documentation.

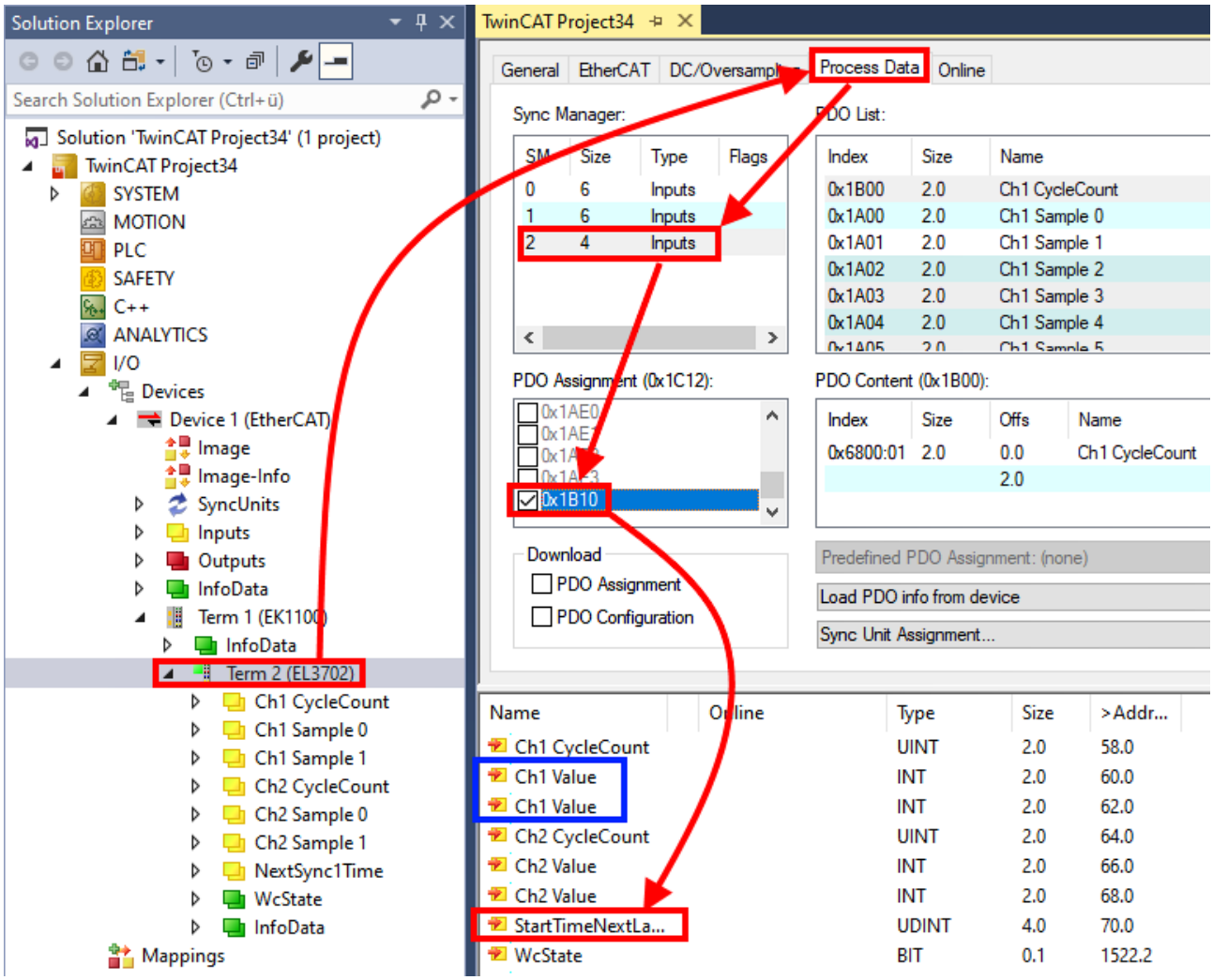
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

- Export the slave configuration from the project,
- Store and transport as a file,
- Import into another EtherCAT project.

TwinCAT offers two methods for this purpose:

- within the TwinCAT environment: Export/Import as **xti** file or
- outside, i.e. beyond the TwinCAT limits: Export/Import as **sci** file.

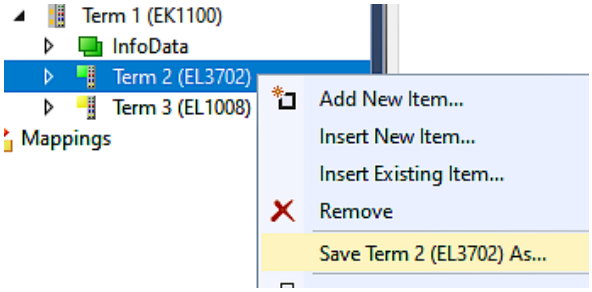
An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):



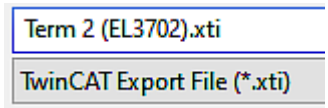
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.3.8.2 Procedure within TwinCAT with xti files

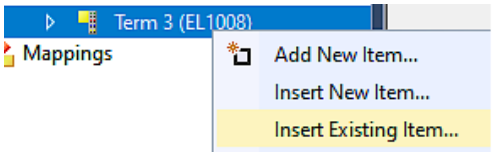
Each IO device can be exported/saved individually:



The xti file can be stored:



and imported again in another TwinCAT system via "Insert Existing item":



5.3.8.3 Procedure within and outside TwinCAT with sci file

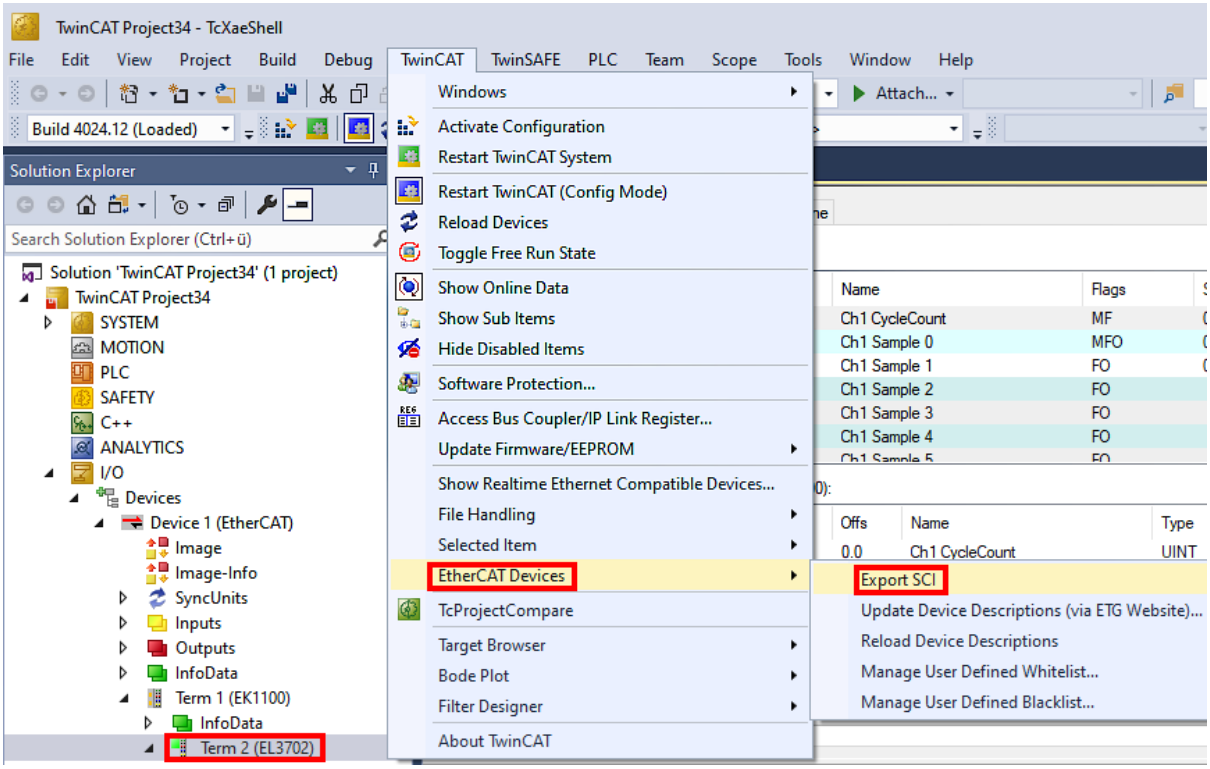
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

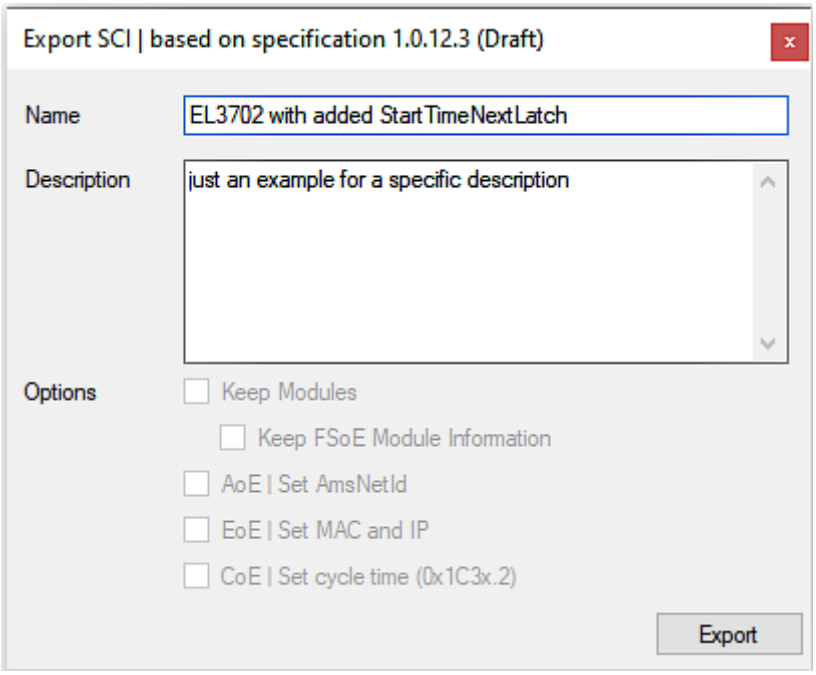
Export:

- select a single device via the menu (multiple selection is also possible):
  TwinCAT → EtherCAT Devices → Export SCI.



- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking OK:

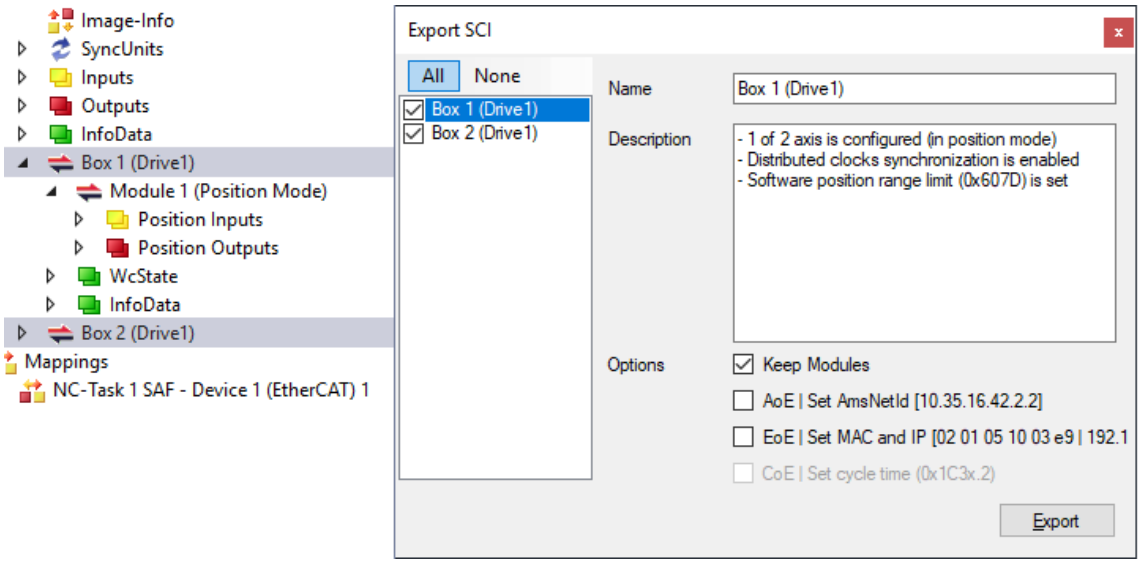
- A description may also be provided:



- Explanation of the dialog box:

Name		Name of the SCI, assigned by the user.
Description		Description of the slave configuration for the use case, assigned by the user.
Options	Keep modules	If a slave supports modules/slots, the user can decide whether these are to be exported or whether the module and device data are to be combined during export.
	AoE | Set AmsNetId	The configured AmsNetId is exported. Usually this is network-dependent and cannot always be determined in advance.
	EoE | Set MAC and IP	The configured virtual MAC and IP addresses are stored in the SCI. Usually these are network-dependent and cannot always be determined in advance.
	CoE | Set cycle time(0x1C3x.2)	The configured cycle time is exported. Usually this is network-dependent and cannot always be determined in advance.
ESI		Reference to the original ESI file.
Export		Save SCI file.

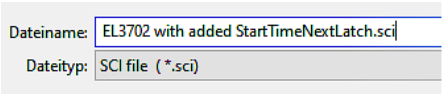
- A list view is available for multiple selections (*Export multiple SCI files)*:



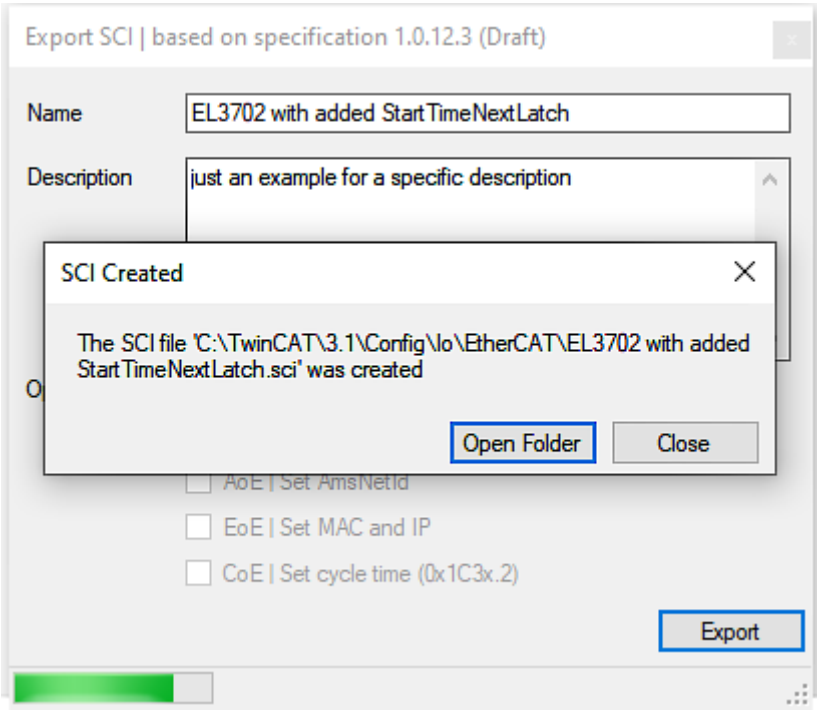
- Selection of the slaves to be exported:
  - All:
    All slaves are selected for export.

- None:
  All slaves are deselected.

- The sci file can be saved locally:

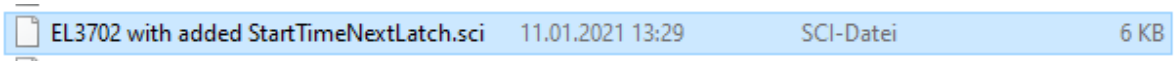


- The export takes place:

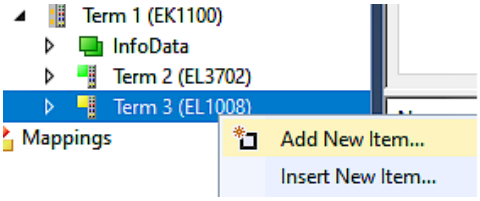


Import

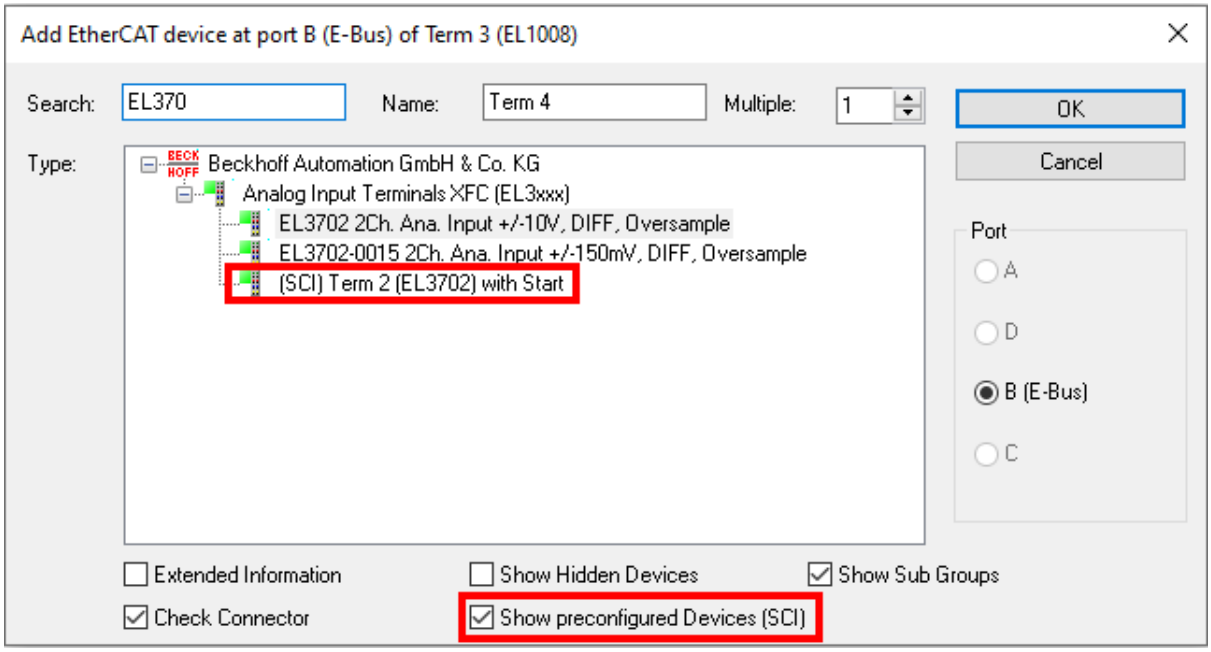
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under:
  C:\TwinCAT\3.1\Config\Io\EtherCAT



- Open the selection dialog:

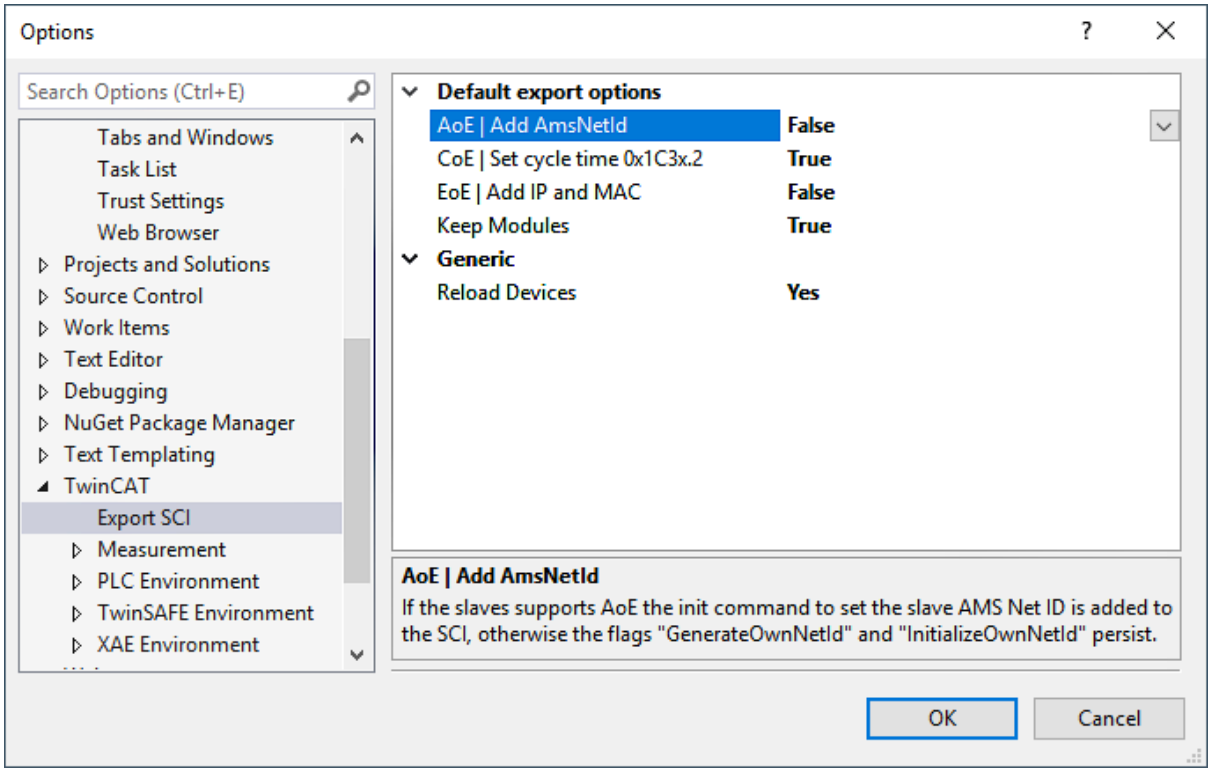


- Display SCI devices and select and insert the desired device:



Additional Notes

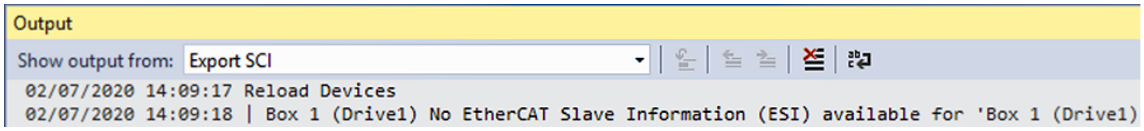
- Settings for the SCI function can be made via the general Options dialog (Tools → Options → TwinCAT → Export SCI):



Explanation of the settings:

Default export options	AoE | Set AmsNetId	Default setting whether the configured AmsNetId is exported.
	CoE | Set cycle time(0x1C3x.2)	Default setting whether the configured cycle time is exported.
	EoE | Set MAC and IP	Default setting whether the configured MAC and IP addresses are exported.
	Keep modules	Default setting whether the modules persist.
Generic	Reload Devices	Setting whether the Reload Devices command is executed before the SCI export. This is strongly recommended to ensure a consistent slave configuration.

SCI error messages are displayed in the TwinCAT logger output window if required:



5.4 EtherCAT basics

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

5.5 EtherCAT cabling – wire-bound

The cable length between two EtherCAT devices must not exceed 100 m. This results from the FastEthernet technology, which, above all for reasons of signal attenuation over the length of the cable, allows a maximum link length of 5 + 90 + 5 m if cables with appropriate properties are used. See also the Design recommendations for the infrastructure for EtherCAT/Ethernet.

Cables and connectors

For connecting EtherCAT devices only Ethernet connections (cables + plugs) that meet the requirements of at least category 5 (CAt5) according to EN 50173 or ISO/IEC 11801 should be used. EtherCAT uses 4 wires for signal transfer.

EtherCAT uses RJ45 plug connectors, for example. The pin assignment is compatible with the Ethernet standard (ISO/IEC 8802-3).

Pin	Color of conductor	Signal	Description
1	yellow	TD +	Transmission Data +
2	orange	TD -	Transmission Data -
3	white	RD +	Receiver Data +
6	blue	RD -	Receiver Data -

Due to automatic cable detection (auto-crossing) symmetric (1:1) or cross-over cables can be used between EtherCAT devices from Beckhoff.

Recommended cables

It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

Suitable cables for the connection of EtherCAT devices can be found on the Beckhoff website!

E-Bus supply

A bus coupler can supply the EL terminals added to it with the E-bus system voltage of 5 V; a coupler is thereby loadable up to 2 A as a rule (see details in respective device documentation).
Information on how much current each EL terminal requires from the E-bus supply is available online and in the catalogue. If the added terminals require more current than the coupler can supply, then power feed terminals (e.g. EL9410) must be inserted at appropriate places in the terminal strand.

The pre-calculated theoretical maximum E-Bus current is displayed in the TwinCAT System Manager. A shortfall is marked by a negative total amount and an exclamation mark; a power feed terminal is to be placed before such a position.

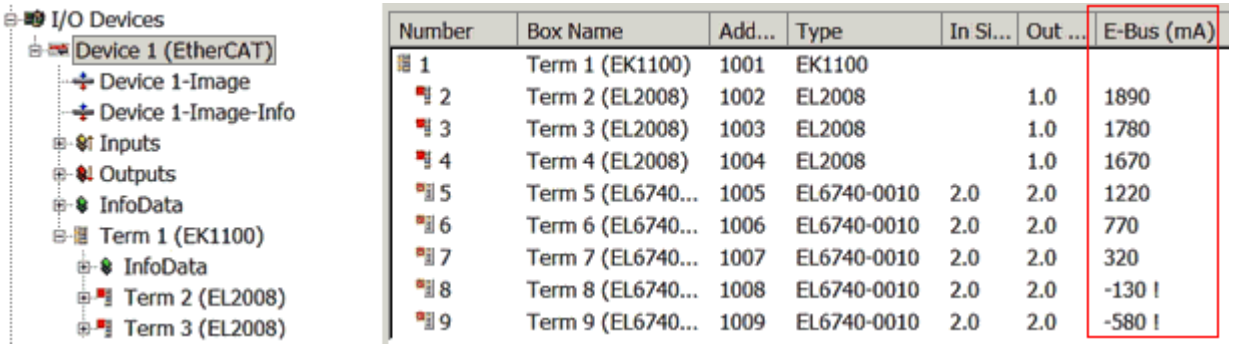


Fig. 121: System manager current calculation

NOTE

Malfunction possible!

The same ground potential must be used for the E-Bus supply of all EtherCAT terminals in a terminal block!

5.6 General notes for setting the watchdog

The ELxxxx terminals are equipped with a safety device (watchdog) which, e.g. in the event of interrupted process data traffic, switches the outputs (if present) to a presettable state after a presettable time, depending on the device and setting, e.g. to FALSE (off) or an output value.

The EtherCAT slave controller (ESC) features two watchdogs:

- SM watchdog (default: 100 ms)
- PDI watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:

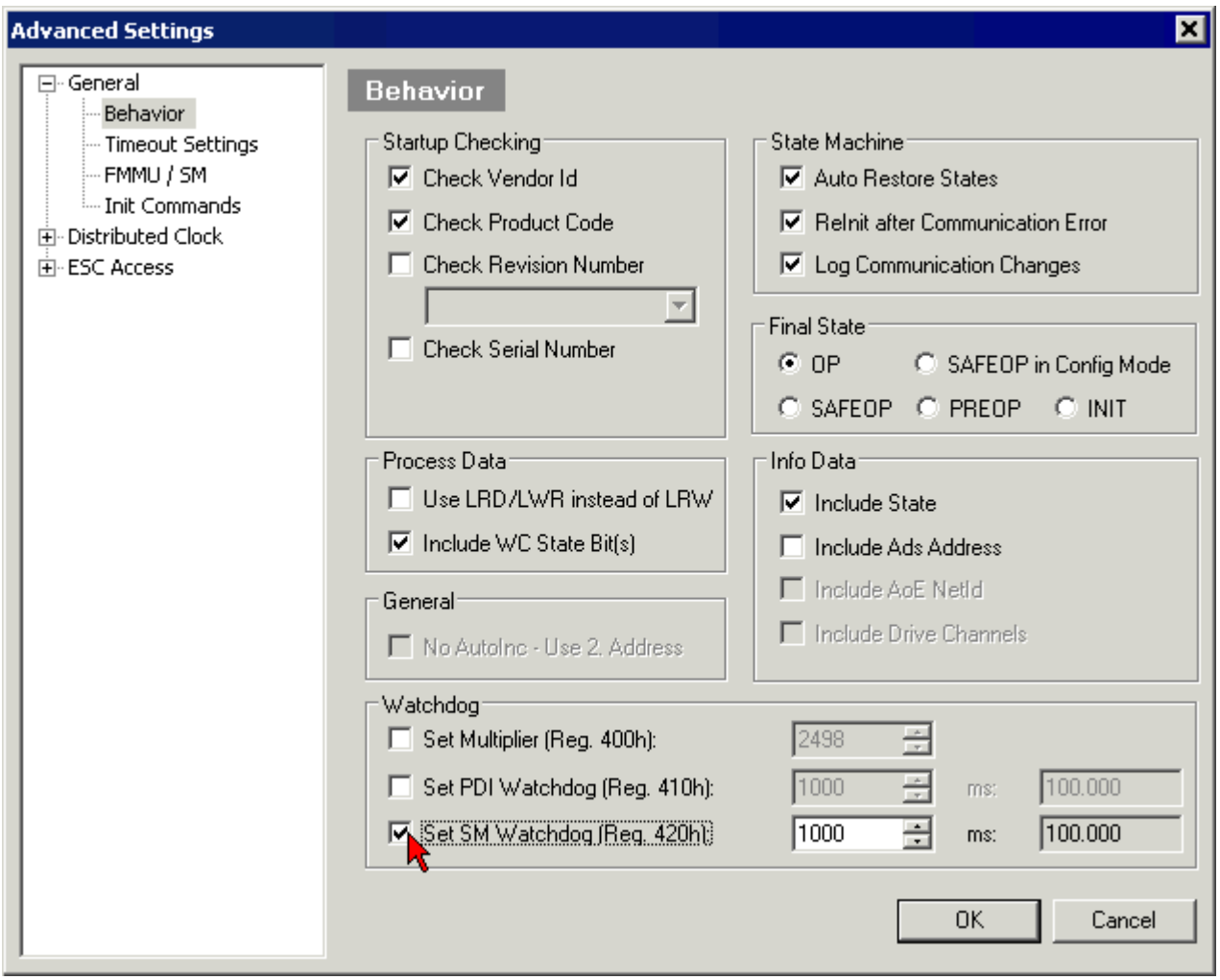


Fig. 122: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i.e. x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.
- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers x0400/0410/0420: ESC Access -> Memory

SM watchdog (SyncManager Watchdog)

The SyncManager watchdog is reset with each successful EtherCAT process data communication with the terminal. If, for example, no EtherCAT process data communication with the terminal takes place for longer than the set and activated SM watchdog time due to a line interruption, the watchdog is triggered. The status of the terminal (usually OP) remains unaffected. The watchdog is only reset again by a successful EtherCAT process data access.

The SyncManager watchdog is therefore a monitoring for correct and timely process data communication with the ESC from the EtherCAT side.

The maximum possible watchdog time depends on the device. For example, for "simple" EtherCAT slaves (without firmware) with watchdog execution in the ESC it is usually up to ~170 seconds. For "complex" EtherCAT slaves (with firmware) the SM watchdog function is usually parameterized via Reg. 400/420 but executed by the µC and can be significantly lower. In addition, the execution may then be subject to a certain time uncertainty. Since the TwinCAT dialog may allow inputs up to 65535, a test of the desired watchdog time is recommended.

PDI watchdog (Process Data Watchdog)

If there is no PDI communication with the EtherCAT slave controller (ESC) for longer than the set and activated PDI watchdog time, this watchdog is triggered.

PDI (Process Data Interface) is the internal interface of the ESC, e.g. to local processors in the EtherCAT slave. With the PDI watchdog this communication can be monitored for failure.

The PDI watchdog is therefore a monitoring for correct and timely process data communication with the ESC, but viewed from the application side.

Calculation

Watchdog time = [1/25 MHz * (Watchdog multiplier + 2)] * PDI/SM watchdog

Example: default setting Multiplier=2498, SM watchdog=1000 -> 100 ms

The value in Multiplier + 2 corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

The function for switching off of the SM watchdog via SM watchdog = 0 is only implemented in terminals from version -0016. In previous versions this operating mode should not be used.

⚠ CAUTION

Damage of devices and undefined state possible!

If the SM watchdog is activated and a value of 0 is entered the watchdog switches off completely. This is the deactivation of the watchdog! Set outputs are NOT set in a safe state if the communication is interrupted.

5.7 EtherCAT State Machine

The state of the EtherCAT slave is controlled via the EtherCAT State Machine (ESM). Depending upon the state, different functions are accessible or executable in the EtherCAT slave. Specific commands must be sent by the EtherCAT master to the device in each state, particularly during the bootup of the slave.

A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational and
- Operational
- Boot

The regular state of each EtherCAT slave after bootup is the OP state.

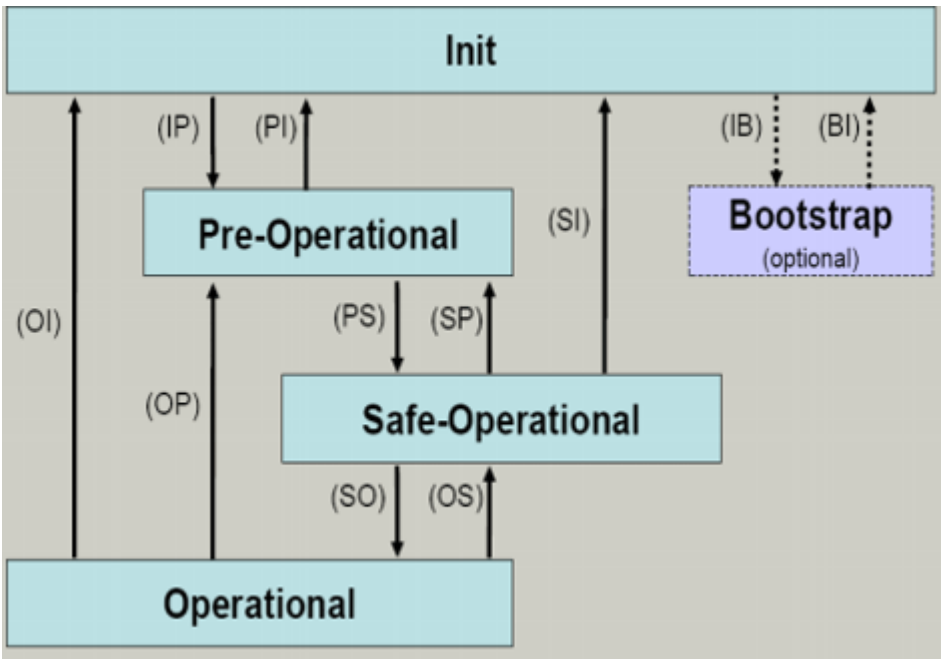


Fig. 123: States of the EtherCAT State Machine

Init

After switch-on the EtherCAT slave in the *Init* state. No mailbox or process data communication is possible. The EtherCAT master initializes sync manager channels 0 and 1 for mailbox communication.

Pre-Operational (Pre-Op)

During the transition between *Init* and *Pre-Op* the EtherCAT slave checks whether the mailbox was initialized correctly.

In *Pre-Op* state mailbox communication is possible, but not process data communication. The EtherCAT master initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the slave supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

Safe-Operational (Safe-Op)

During transition between *Pre-Op* and *Safe-Op* the EtherCAT slave checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT slave copies current input data into the associated DP-RAM areas of the EtherCAT slave controller (ECSC).

In *Safe-Op* state mailbox and process data communication is possible, although the slave keeps its outputs in a safe state, while the input data are updated cyclically.

Outputs in SAFEOP state

The default set watchdog [▶ 109] monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

Operational (Op)

Before the EtherCAT master switches the EtherCAT slave from *Safe-Op* to *Op* it must transfer valid output data.

In the *Op* state the slave copies the output data of the masters to its outputs. Process data and mailbox communication is possible.

Boot

In the *Boot* state the slave firmware can be updated. The *Boot* state can only be reached via the *Init* state.

In the *Boot* state mailbox communication via the *file access over EtherCAT* (FoE) protocol is possible, but no other mailbox communication and no process data communication.

5.8 CoE Interface

General description

The CoE interface (CAN application protocol over EtherCAT)) is used for parameter management of EtherCAT devices. EtherCAT slaves or the EtherCAT master manage fixed (read only) or variable parameters which they require for operation, diagnostics or commissioning.

CoE parameters are arranged in a table hierarchy. In principle, the user has read access via the fieldbus. The EtherCAT master (TwinCAT System Manager) can access the local CoE lists of the slaves via EtherCAT in read or write mode, depending on the attributes.

Different CoE parameter types are possible, including string (text), integer numbers, Boolean values or larger byte fields. They can be used to describe a wide range of features. Examples of such parameters include manufacturer ID, serial number, process data settings, device name, calibration values for analog measurement or passwords.

The order is specified in two levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

- Index: 0x0000 …0xFFFF ($0...65535_{dec}$)
- SubIndex: 0x00…0xFF ($0...255_{dec}$)

A parameter localized in this way is normally written as 0x8010:07, with preceding “0x” to identify the hexadecimal numerical range and a colon between index and subindex.

The relevant ranges for EtherCAT fieldbus users are:

- 0x1000: This is where fixed identity information for the device is stored, including name, manufacturer, serial number etc., plus information about the current and available process data configurations.
- 0x8000: This is where the operational and functional parameters for all channels are stored, such as filter settings or output frequency.

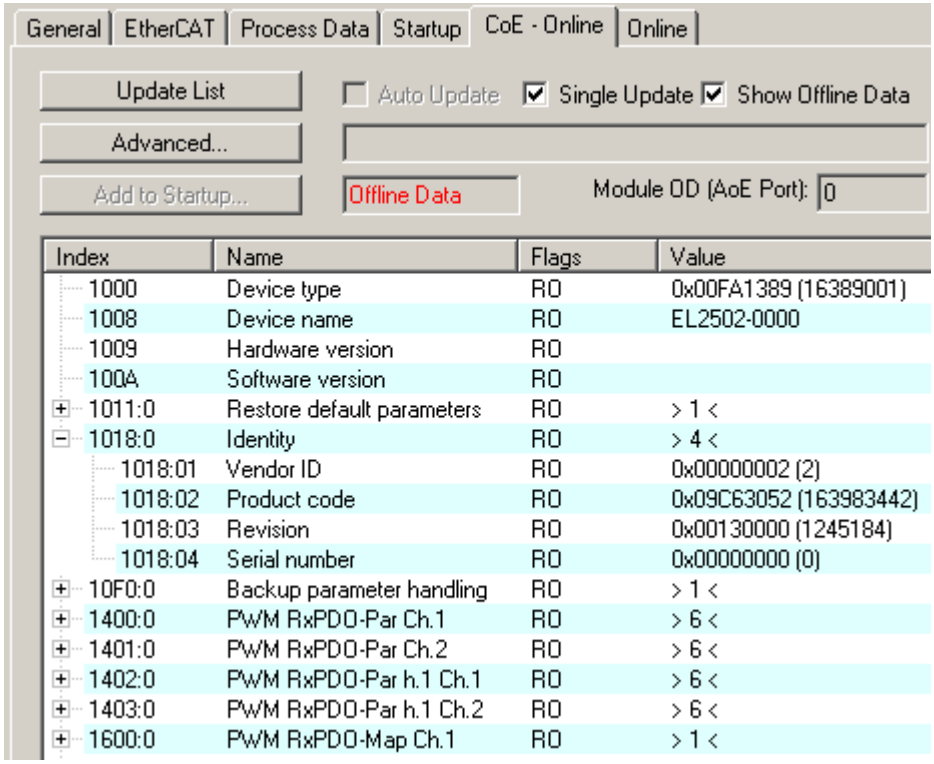
Other important ranges are:

- 0x4000: here are the channel parameters for some EtherCAT devices. Historically, this was the first parameter area before the 0x8000 area was introduced. EtherCAT devices that were previously equipped with parameters in 0x4000 and changed to 0x8000 support both ranges for compatibility reasons and mirror internally.
- 0x6000: Input PDOs (“input” from the perspective of the EtherCAT master)
- 0x7000: Output PDOs (“output” from the perspective of the EtherCAT master)

Availability

Not every EtherCAT device must have a CoE list. Simple I/O modules without dedicated processor usually have no variable parameters and therefore no CoE list.

If a device has a CoE list, it is shown in the TwinCAT System Manager as a separate tab with a listing of the elements:



Index	Name	Flags	Value
1000	Device type	RO	0x00FA1389 (16389001)
1008	Device name	RO	EL2502-0000
1009	Hardware version	RO	
100A	Software version	RO	
1011:0	Restore default parameters	RO	> 1 <
1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x09C63052 (163983442)
1018:03	Revision	RO	0x00130000 (1245184)
1018:04	Serial number	RO	0x00000000 (0)
10F0:0	Backup parameter handling	RO	> 1 <
1400:0	PWM RxPDO-Par Ch.1	RO	> 6 <
1401:0	PWM RxPDO-Par Ch.2	RO	> 6 <
1402:0	PWM RxPDO-Par h.1 Ch.1	RO	> 6 <
1403:0	PWM RxPDO-Par h.1 Ch.2	RO	> 6 <
1600:0	PWM RxPDO-Map Ch.1	RO	> 1 <

Fig. 124: “CoE Online” tab

The figure above shows the CoE objects available in device “EL2502”, ranging from 0x1000 to 0x1600. The subindices for 0x1018 are expanded.

NOTE

Changes in the CoE directory (CAN over EtherCAT), program access

When using/manipulating the CoE parameters observe the general CoE notes in chapter "CoE interface" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the Beckhoff website),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see TwinCAT3 | PLC Library: Tc2_EtherCAT and Example program R/W CoE)

Data management and function “NoCoeStorage”

Some parameters, particularly the setting parameters of the slave, are configurable and writeable. This can be done in write or read mode

- via the System Manager (Fig. “CoE Online” tab) by clicking
  This is useful for commissioning of the system/slaves. Click on the row of the index to be parameterized and enter a value in the “SetValue” dialog.
- from the control system/PLC via ADS, e.g. through blocks from the TcEtherCAT.lib library
  This is recommended for modifications while the system is running or if no System Manager or operating staff are available.

Data management

If slave CoE parameters are modified online, Beckhoff devices store any changes in a fail-safe manner in the EEPROM, i.e. the modified CoE parameters are still available after a restart.
The situation may be different with other manufacturers.

An EEPROM is subject to a limited lifetime with respect to write operations. From typically 100,000 write operations onwards it can no longer be guaranteed that new (changed) data are reliably saved or are still readable. This is irrelevant for normal commissioning. However, if CoE parameters are continuously changed via ADS at machine runtime, it is quite possible for the lifetime limit to be reached. Support for the NoCoeStorage function, which suppresses the saving of changed CoE values, depends on the firmware version.
Please refer to the technical data in this documentation as to whether this applies to the respective device.

- If the function is supported: the function is activated by entering the code word 0x12345678 once in CoE 0xF008 and remains active as long as the code word is not changed. After switching the device on it is then inactive. Changed CoE values are not saved in the EEPROM and can thus be changed any number of times.
- Function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

Startup list

Changes in the local CoE list of the terminal are lost if the terminal is replaced. If a terminal is replaced with a new Beckhoff terminal, it will have the default settings. It is therefore advisable to link all changes in the CoE list of an EtherCAT slave with the Startup list of the slave, which is processed whenever the EtherCAT fieldbus is started. In this way a replacement EtherCAT slave can automatically be parameterized with the specifications of the user.

If EtherCAT slaves are used which are unable to store local CoE values permanently, the Startup list must be used.

Recommended approach for manual modification of CoE parameters

- Make the required change in the System Manager
  The values are stored locally in the EtherCAT slave
- If the value is to be stored permanently, enter it in the Startup list.
  The order of the Startup entries is usually irrelevant.

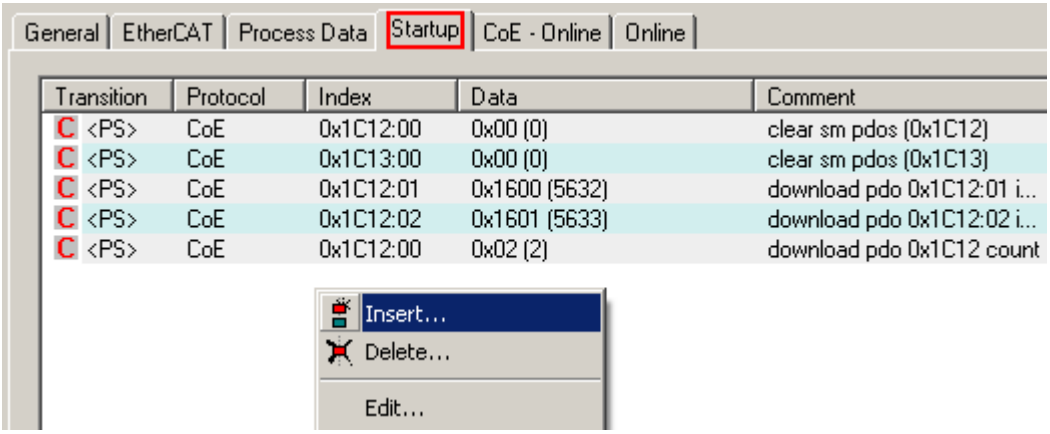


Fig. 125: Startup list in the TwinCAT System Manager

The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can be created.

Online/offline list

While working with the TwinCAT System Manager, a distinction has to be made whether the EtherCAT device is “available”, i.e. switched on and linked via EtherCAT and therefore **online**, or whether a configuration is created **offline** without connected slaves.

In both cases a CoE list as shown in Fig. “CoE online tab” is displayed. The connectivity is shown as offline/ online.

- If the slave is offline
  - The offline list from the ESI file is displayed. In this case modifications are not meaningful or possible.
  - The configured status is shown under Identity.
  - No firmware or hardware version is displayed, since these are features of the physical device.
  - **Offline** is shown in red.

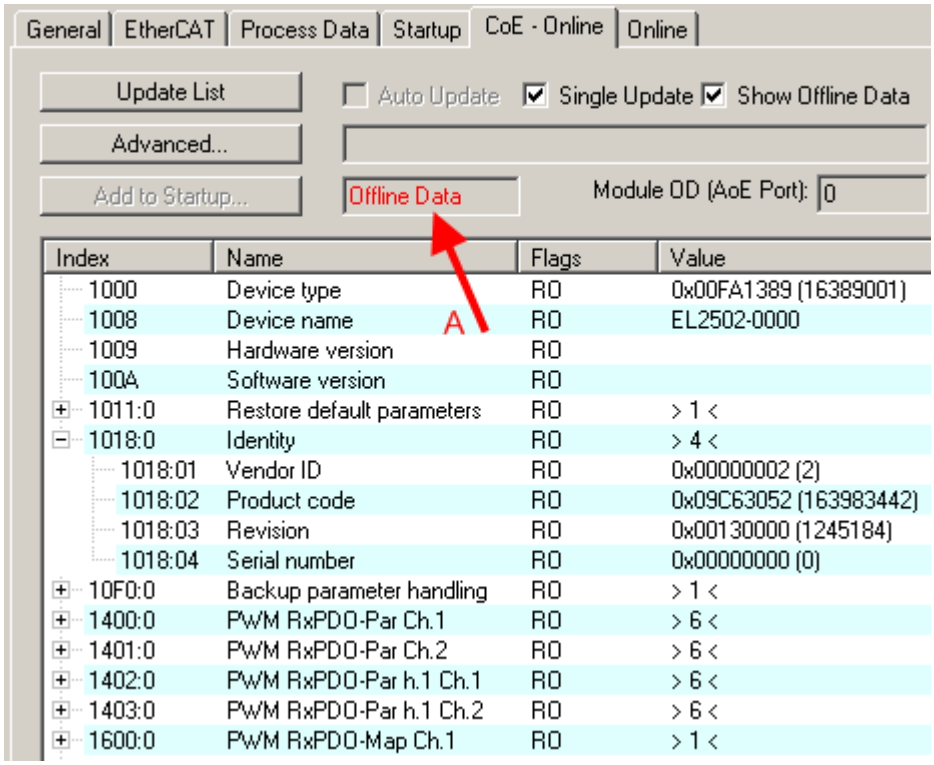


Fig. 126: Offline list

- If the slave is online
  - The actual current slave list is read. This may take several seconds, depending on the size and cycle time.
  - The actual identity is displayed
  - The firmware and hardware version of the equipment according to the electronic information is displayed
  - **Online** is shown in green.

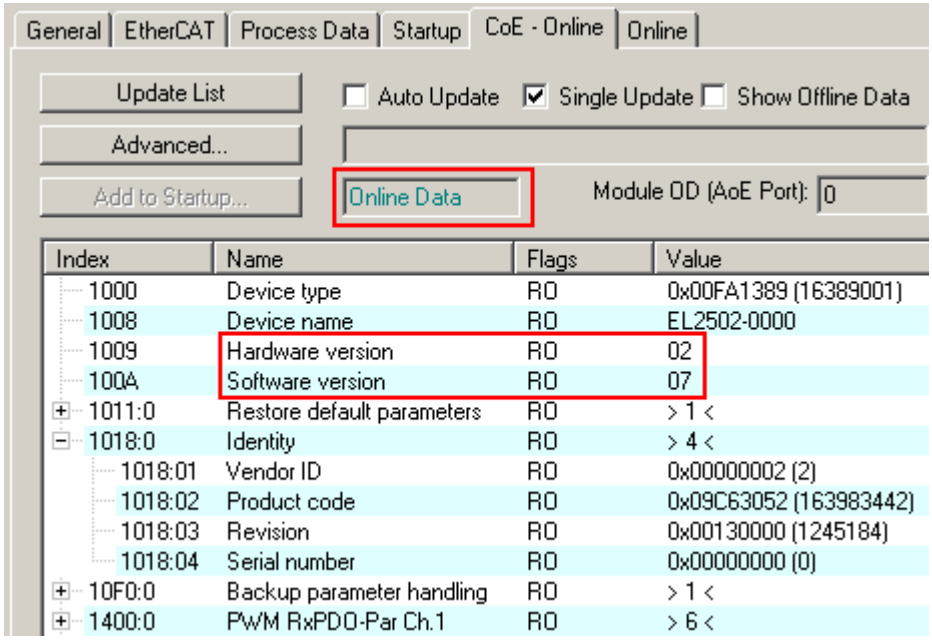


Fig. 127: Online list

Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels. For example, a 4-channel analog 0...10 V input terminal also has four logical channels and therefore four identical sets of parameter data for the channels. In order to avoid having to list each channel in the documentation, the placeholder “n” tends to be used for the individual channel numbers.

In the CoE system 16 indices, each with 255 subindices, are generally sufficient for representing all channel parameters. The channel-based order is therefore arranged in $16_{dec}/10_{hex}$ steps. The parameter range 0x8000 exemplifies this:

- Channel 0: parameter range 0x8000:00 ... 0x800F:255
- Channel 1: parameter range 0x8010:00 ... 0x801F:255
- Channel 2: parameter range 0x8020:00 ... 0x802F:255
- ...

This is generally written as 0x80n0.

Detailed information on the CoE interface can be found in the EtherCAT system documentation on the Beckhoff website.

5.9 Distributed Clock

The distributed clock represents a local clock in the EtherCAT slave controller (ESC) with the following characteristics:

- Unit *1 ns*
- Zero point *1.1.2000 00:00*
- Size *64 bit* (sufficient for the next 584 years; however, some EtherCAT slaves only offer 32-bit support, i.e. the variable overflows after approx. 4.2 seconds)
- The EtherCAT master automatically synchronizes the local clock with the master clock in the EtherCAT bus with a precision of < 100 ns.

For detailed information please refer to the EtherCAT system description.

6 Housing

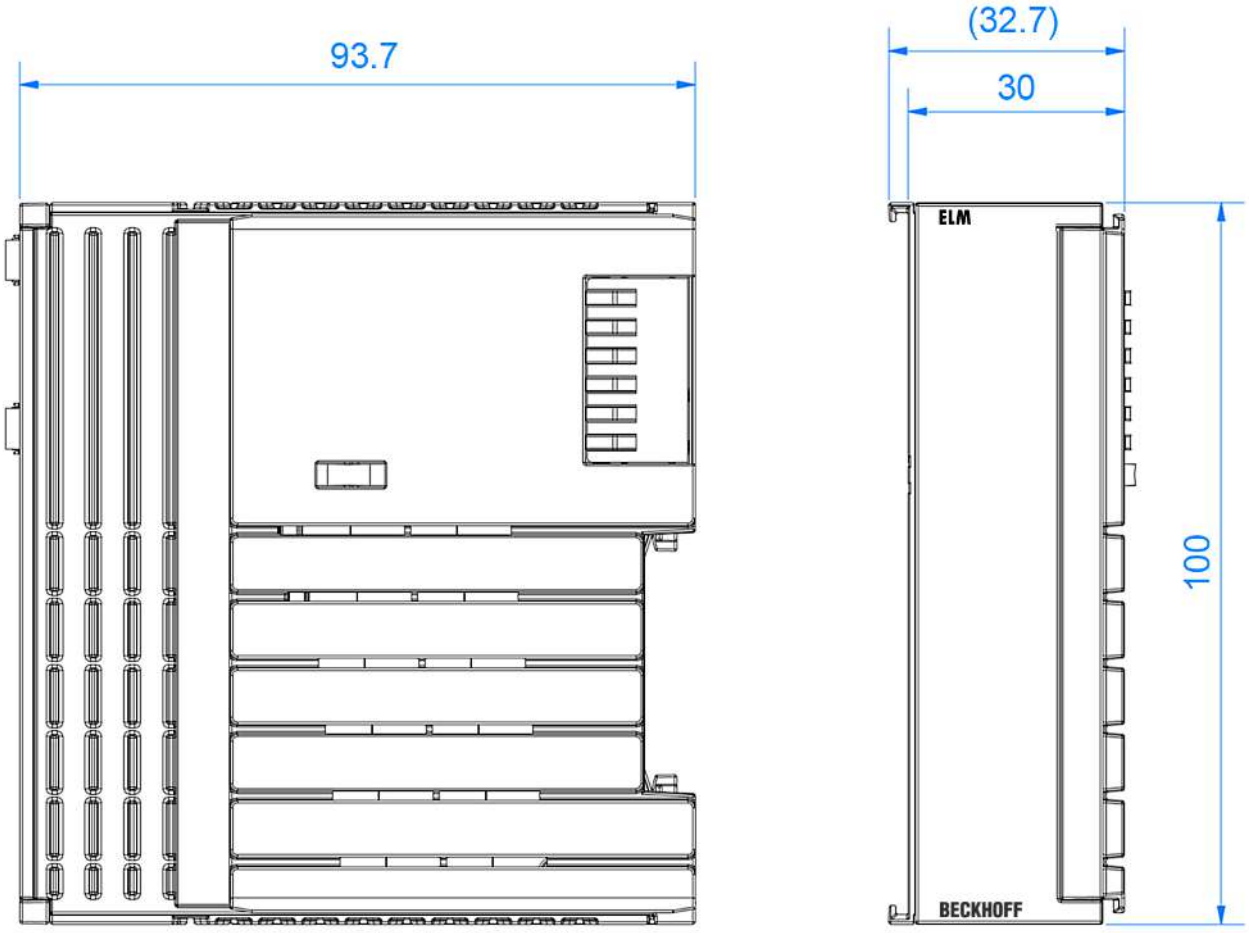


Fig. 128: Dimensions: ELM2xxx terminals

6.1 Housing data

Housing data

Terminal	Connection type	Width	Height	Depth
EL2642	Cage clamp	12 mm	100 mm	68 mm
ELM2642-0000 ELM2644-0000 ELM2742-0000 ELM2744-0000	push-in, for direct wiring, plug connector detachable for service	33 mm	100 mm	95 mm

7 Mounting and wiring

7.1 Notes regarding connectors and wiring

It is in the very nature of EtherCAT I/O modules/terminals/box modules that they have two connection sides: one to the fieldbus for communication with the module, which is obligatory, the other to the signal/sensor/actuator to facilitate proper use of the module. The “outer” connection side usually features contacting options for connecting outgoing wires.

Only few I/O devices do not have a second side. Examples include the EL6070 license key terminal and the EL6090 display terminal.

Notes and suggestions for dealing with the connection options are provided below

- **Manufacturer specifications**/notes for connection options must be followed. Any special tools that may have been provided must be used as intended, so that gas-tightness is ensured through the crimping pressure.
- Any detachable connection system is subject to a specified maximum number of **connection cycles**. Each connection/disconnection operation results in wear through friction, mechanical stretching/relaxation, possibly ingress of contaminants/gases/liquids/condensation, contact discharge, modification of the electrical properties and of the contact point (ohmic contact resistance). In other words, releasing/connecting a contact results in mechanical, chemical and therefore ultimately electrical changes.
  In terms of the application scenario it is therefore important to select suitable connection systems or devices with suitable connection systems:
  - For connections that are more or less permanent, it may make sense to use connectors/contacts with a maximum **number of mating cycles** (as specified by the manufacturer) of 10 to 100 cycles. This may be the case if devices are installed/wired only once, and over the entire lifetime rewiring is only expected to become necessary during maintenance work.
  - For connections that have to be detached on a regular basis, connectors/contacts with a maximum number of mating cycles of 1,000 or higher should be selected. Such connections can typically be found in laboratory environments, where the cabling may be changed several times each day but high-quality contact must nevertheless be ensured over many years.
- When handling and assembling connectors/contacts it is essential to avoid **contact with hand perspiration/liquids**, even for low-tech connections (open stranded wire, cage clamp/push-in). Acidic/alkaline liquids may have a very aggressive effect on the contact surface and quickly lead to structural changes and oxidation layers. These are very disruptive for analog measurements, particularly since they undermine the reproducibility of measurements and can therefore result (if known) in large systematic measurement uncertainty. It may be possible to rectify the problem by thorough follow-up cleaning.
- The actual/expected **load during operation** must be taken into account when selecting connectors.
- Abnormal vibrations can lead to microfriction/corrosion and change the electrical properties, potentially resulting in complete loss of contact.
- Temperature variations affect the mechanical strength of the connection and the spring forces in metallic components.
- Exposure to gas or liquid can damage the connection, particularly if the gas or liquid penetrates to the actual contact region and is unable to escape from there.
- Of high relevance for analog measurements is the **electrical quality** of the connection, both in the short term during commissioning and over the service life under external influences and perhaps repeated mating cycles. This is expressed in the repeatability of the transition. The influence should be checked against the expected accuracy. Of particular relevance is the (frequency dependent) contact resistance. Effects can be:
  - Increasing the contact resistance results in a voltage drop when power is transmitted, potentially leading to critical self-heating.
  - The internal voltage drop can distort corresponding measurements. In order to avoid negative effects, 4/5/6-wire connections should be used in SG/resistance measurements, since non-live contacts are no longer affected by a distorting voltage drop. The popular 3-wire connection for

resistance measurement (PT100, PT1000 etc.) does not provide absolute protection, since the singular line cannot be diagnosed. Current/voltage measurements in industrial environments are less sensitive to contact changes.

- A defective contact surface can lead to random resistance values, depending on the contact position and temperature. This makes reproducible measurements difficult.

- The **effort for establishing the connection**, including assembling the cables and connectors, generally increases with increasing transmission quality requirements. This applies to the tools, diligence and time required. Examples:
  - Cage clamp/push-in connections (e.g. Beckhoff EL terminals), which are common in automation applications, can be established or released in a few seconds with or without ferrule. A screwdriver or push pin is sufficient. On the other hand, in many cases the (ohmic) repeatability is insufficient for high-precision measurements in the SG/R range.
  - Some 10 minutes and costs of some 10 euros should be assumed for assembly a lab-standard LEMO/ODU connector (Beckhoff ELM3704-0001), depending on the number of poles. The result is a top-quality connection system with a high number of permissible mating cycles.
  - An intermediate solution can be field-configurable M8/M12 connections. For reasons of tightness, they are more elaborate to assemble (soldering or insulation displacement contact, if necessary), although the maximum number of mating cycles is similar to maintenance connectors.
- A pre-assembled connection should be subjected to electrical/mechanical testing before commissioning: visual inspection, pull-out test, crimp height measurement, resistance measurement etc.

7.2 Notes on connection technology

Connection type: Push-in with service plug

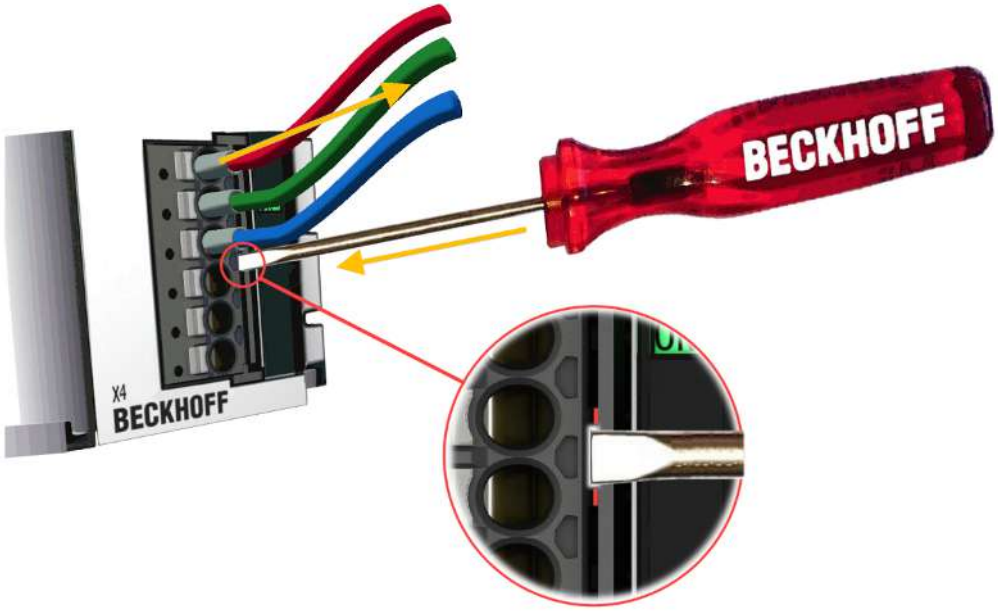
The wires are plugged in directly; for solid wires no tools are required, i.e. after the insulation has been stripped, the wire is simply pushed into the contact point. The same principle applies for the ferrule. Free stranded wire ends can also be connected in this way; in this case the wire clamping mechanism has to be opened by operating the pushing device.

Like in standard terminals, the wires are released via the contact release device, using a screwdriver or pushing device.

The cables must not be pulled/ pushed live or under load.

For maintenance purposes, e.g. during service, the entire plug-in body can be removed from the Beckhoff terminal without releasing the individual wires. Use a screwdriver (e.g. Beckhoff ZB8700) to release the central release device and pull the cables to release the connector body.

Additionally the service plug don’t have specified switching power, also it must not be pulled/ pushed live or under load, too.



The permitted conductor cross-sections and the strip length are shown in the following table.

Wire cross-section (solid wire)	0.2 ... 1.5 mm^2
Wire cross-section (stranded wire)	0.2 ... 1.5 mm^2
Wire cross-section (stranded)	0.25 ... 0.75 mm^2 (with ferrule with plastic collar)
Wire cross-section (stranded wire)	0.25 ... 1.5 mm^2 (with ferrule without plastic collar)
Current rating, permanent	5 A
Conductor (AWG)	24 – 14 | 14: THHN, THWN
Strip length	8 ... 9 mm / 0.31 – 0.35 in

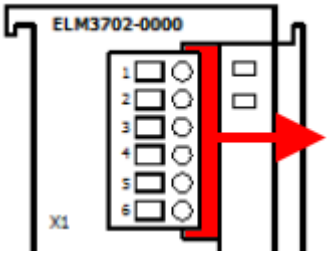
Releasing the contact

The push-in connector is supplied with the terminal.

The push-in connector is designed as a service plug.

Maximum number of mating cycles: 10

The connector with connected wires can be removed by pushing the unlocking tab (red) in the direction of the arrow, e.g. with a screwdriver, thereby releasing the unlocking device.



Meticulous cleanliness must be ensured when the connector is re-inserted. Do not touch the pins in the device tray. Push in the connector until it latches audibly and the front of the plug is flush with the ELM housing.

7.3 Note - Power supply

⚠ WARNING

Power supply from SELV/PELV power supply unit!

SELV/PELV circuits (Safety Extra Low Voltage, Protective Extra Low Voltage) according to IEC 61010-2-201 must be used to supply this device.

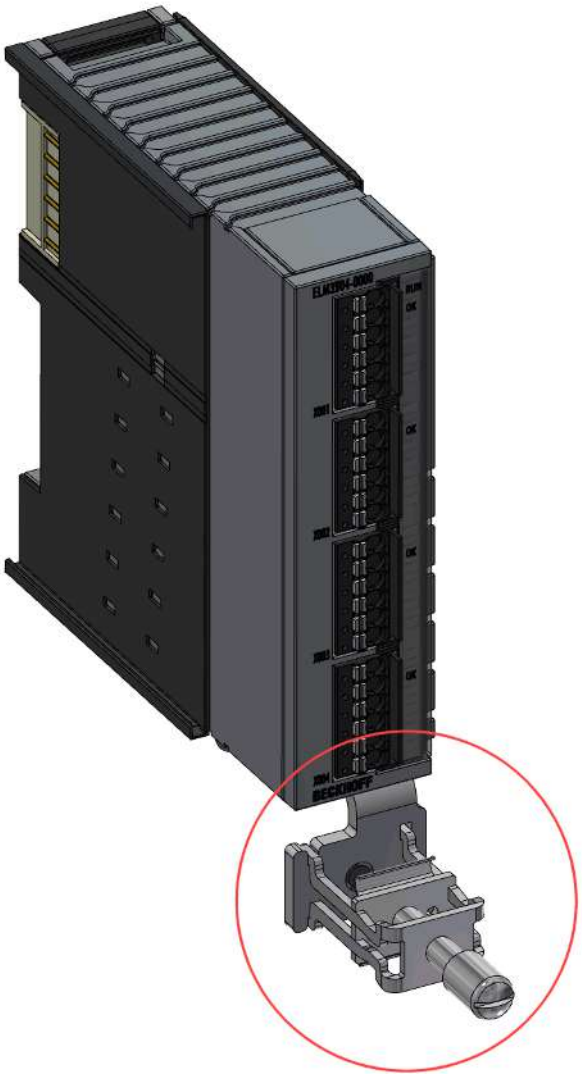
Notes:

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

7.4 Accessories

The following accessories are currently available for the analog input terminals of the ELM3xxx series

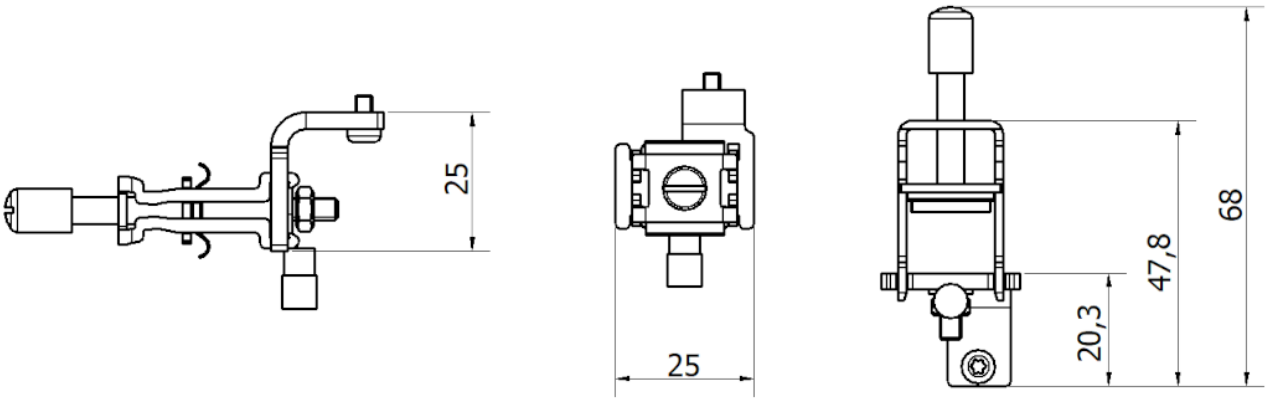
7.4.1 Shield connection



The shield connection is an optional component, which can be installed on the underside of the ELMxxxx housing. It must be ordered separately.

Available models:

- ZS9100-0002: Shield connection for ELM series
  - screw clamping, packaging unit = 1 piece



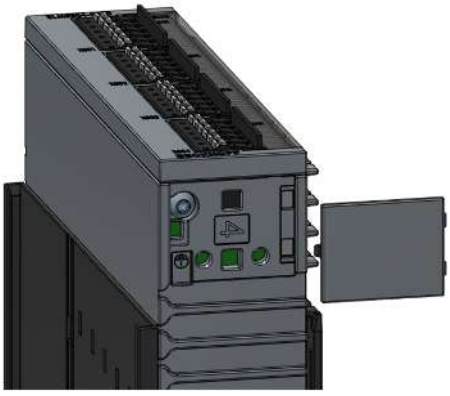
- ZS5300-0015: Shielding bracket for EtherCAT connection of EtherCAT Terminal ELM721x/ELM722x
  - clamping, 2 cables, packaging unit = 5 pieces
- ZS5300-0016: Shielding bracket for EtherCAT connection of EtherCAT Terminal ELM723x
  - clamping, 1 cable, packaging unit = 5 pieces

It is used as a low-resistance earthing connection at the housing, to deal with electrical interference signals arriving via the cable screen. The fault signals are then directed to the DIN rail via the metallic ELM housing and the integrated grounding springs. For this to work, the DIN rail/control cabinet also must have a low-resistance connection, of course.

Note: Electrical faults usually occur in the form of high-frequency signals. Therefore, it is important to not only ensure a good low-resistance connection for DC signals (continuity test with a multimeter), but also to ensure its effectiveness for high-frequency signals in the form of a low-impedance connection. This should be tested with special measuring devices unless the general installation instructions regarding EMC-compliant control cabinet construction are observed.

The shield connection should be used as follows:

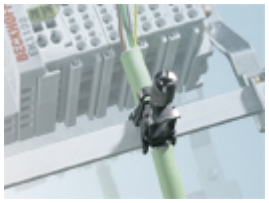
- Lever off the plastic cover from the ELM housing and retain if for later reuse, if required
- Attach the shield connection with the screw provided. Clean the contact surfaces, as appropriate. The second screw hole remains free in case a PE connection is required.
- Strip the signal cable, feed it through the shield clamp and hand-tighten the clamp (recommended screw tightening torque: 0.5 Nm)
- Apply the signal cable wires at the plug connector.
- For disassembly, proceed in reverse order.



Note: the shield connection does not act as strain relief!

Alternative shield connection methods for analog signal lines:

- Beckhoff shielding connection system ZB8500 https://www.beckhoff.com/zb8500/



- Separate shield connection depending on requirements

7.4.2 Shielding hood ZS9100-0003

The shielding hood is an optional component for the ELMxxxx housing series. It has to be ordered separately.

It does not affect the visibility of the LED displays of the terminal.

The shielding hood has two purposes

- Electromagnetic shielding of faults
  If push-in connectors are used, they represent a gateway for faults in the terminal, due to the fact that they are made of plastic. The shielding hood can be installed (either right away or retrospectively) in order to form an enclosed metallic cage around the terminal and the signal cable.
  Alternatively, ELMxxxx terminals with shielded plug connectors can be used (e.g. LEMO, BNC), in which case the shielding hood is not required.
- Thermal shielding for thermocouple measurements
  If the ELM3xxx terminal is used for measuring temperatures with thermocouples, the **integrated** cold junction measurement contributes significantly to the overall uncertainty. Thermal turbulence caused by

air flowing past and radiant heat can lead to large temperature gradients around the plug, resulting in fluctuating temperature measurements. The shielding hood facilitates a thermally stabilized environment around the plug, which helps to increase the measuring accuracy.

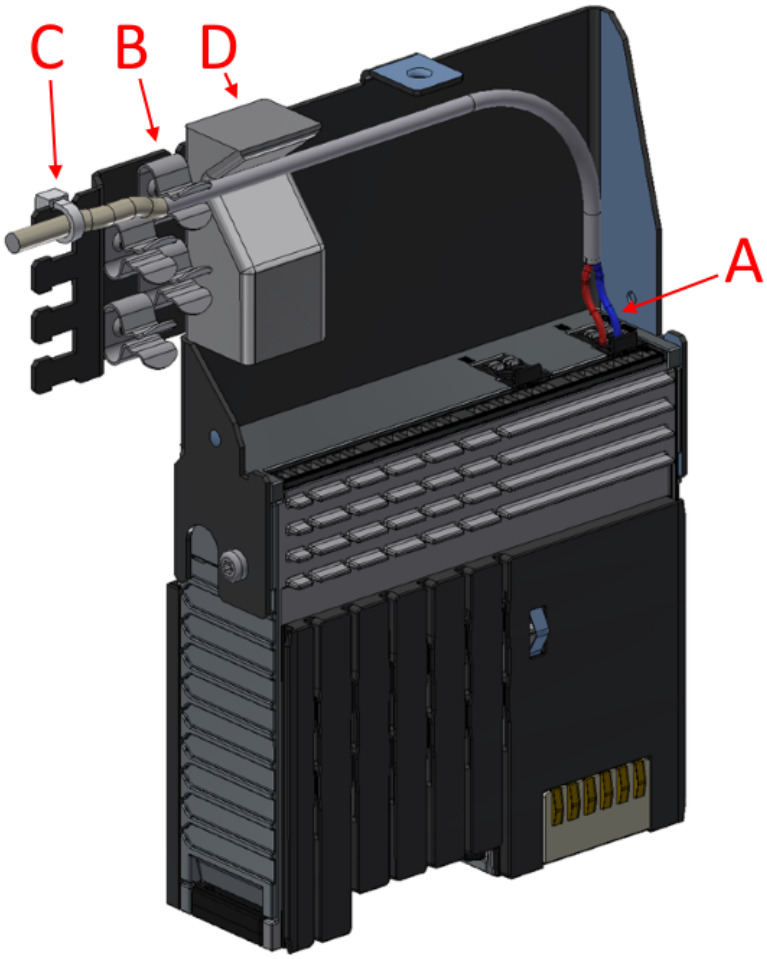
Between one and four commercially available signal lines up to approx. 7 mm shield diameter (usually corresponds to approx. 9 mm outer diameter) can be connected.

Technical data	ZS9100-0003
Weight	approx. 190 g
Dimensions (W x H x D)	26 x 145 x 93 mm effective extended width after mounting: 74 mm
Permissible ambient temperature range during operation and storage	-40…+85 °C
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27 Usage restriction: see below
Protection class	IP 20
Installation position	variable
Approval	CE

The shielding hood should be installed as follows:

- Use a screwdriver to lever off the two painted plastic covers on the top and bottom of the ELM housing; retain the covers for later reuse
- Slide on the shield connection and fasten it with the three screws provided. The fourth screw hole is intended for a PE connection, if required.

- Remove the sheathing from the signal cables and insert the wires into the connectors (A). Then push the shield braid into the EMC clamp (B) and fasten the cable to the strain relief clip (C) using the cable tie provided. Follow the cable manufacturer's recommendations for the bending radius.



- The shield braid should rest on the conductive foam block (D). This block ensures EMC-compliant sealing when the hood is closed.
- Position the hood and hand-tighten it with the knurled screw. Ensure that the unpainted sections and the foam block are in close contact.

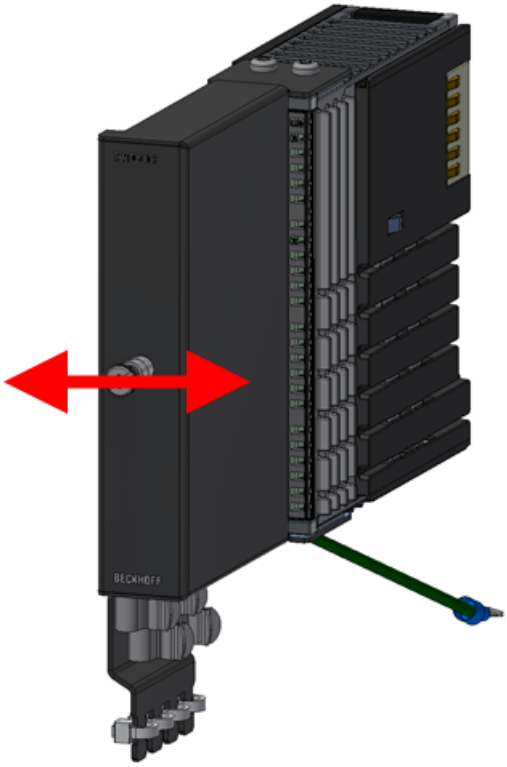
- For disassembly, proceed in reverse order.

Any component identification should be replicated on the hood.

NOTE

Note for use under vibration load

An application of the ELM terminal with mounted shielding hood ZS9100-0003 under vibration and shock effect in the direction of DIN rail track (red arrow) is, regardless of the installation position, not allowed.



If vibration / shock inevitably occurs during operation, an installation position must be selected which does not load the ELM terminal and accordingly the shielding hood in the indicated direction of the arrow. Basically, an additional mechanical support of the shielding hood and cables respectively is recommended for vibration / shock.

7.4.3 Replacement push-in ZS2001-000x

The black push-in service plugs for ELM/EKM terminals can be ordered separately as spare parts. Per unit 10 pieces are included.

ZS2001-000x

Number of poles	Designation
2	ZS2001-0006
4	ZS2001-0007
6	ZS2001-0008
10	ZS2001-0009


ELM3002-0000
X1
X2

7.5 Common notes to the power contacts

If the ELM terminal doesn't have own wheeling of electricity or supply of the power contacts, the terminal on its right mustn't have sticking out power contacts on the left side. They would be free accessible if the ELM terminal would be pulled out from the DIN rail.

Also see about this

ELM/EKM terminal mounting on DIN rail [▶ 135]

7.6 Installation positions

NOTE

Constraints regarding installation position and operating temperature range

Please refer to the technical data for a terminal to ascertain whether any restrictions regarding the installation position and/or the operating temperature range have been specified. When installing high power dissipation terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

Optimum installation position (standard)

The optimum installation position requires the mounting rail to be installed horizontally and the connection surfaces of the EL/KL terminals to face forward (see Fig. *"Recommended distances for standard installation position"*). The terminals are ventilated from below, which enables optimum cooling of the electronics through convection. "From below" is relative to the acceleration of gravity.

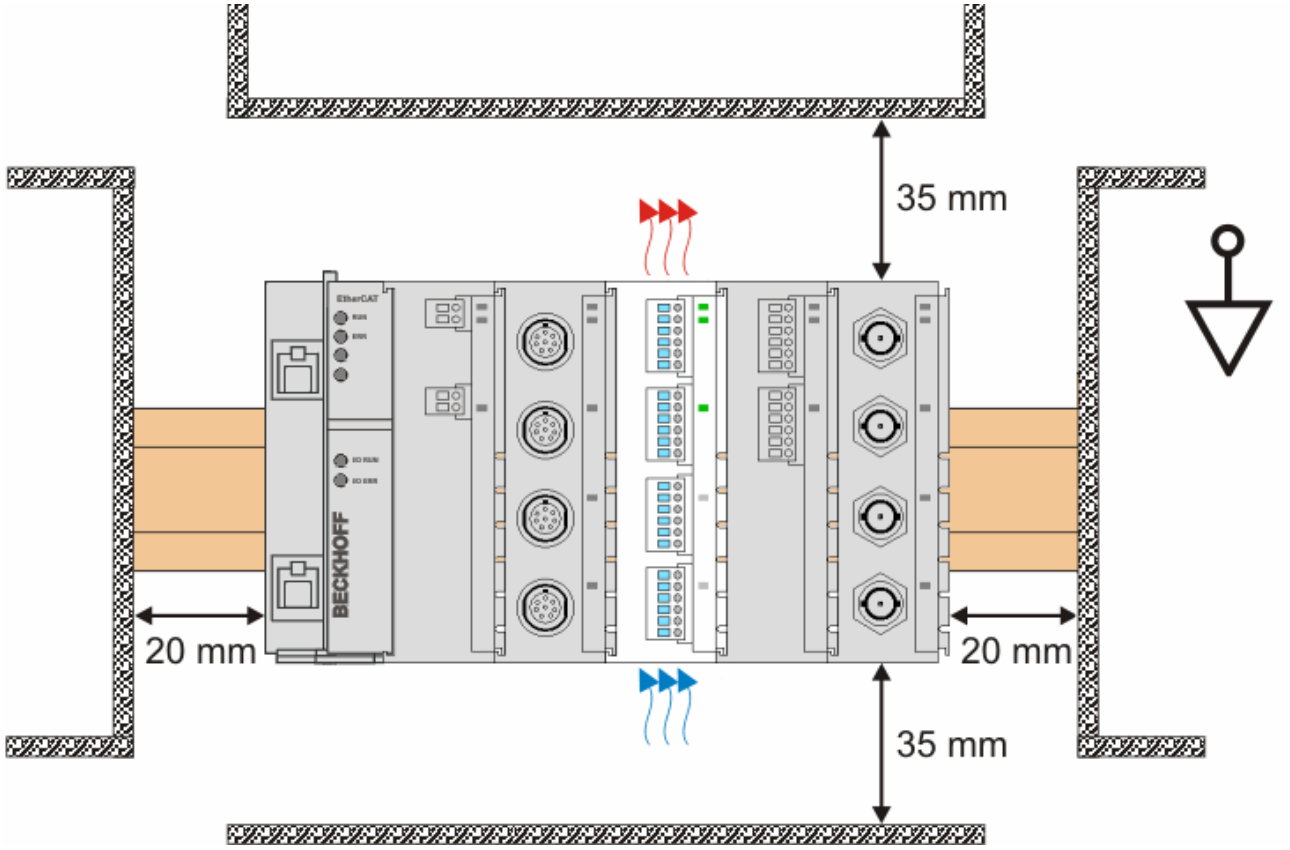


Fig. 129: *Recommended distances for standard installation position*

Compliance with the distances shown in Fig. *"Recommended distances for standard installation position"* is recommended.

Other installation positions

All other installation positions are characterized by different spatial arrangement of the mounting rail - see Fig *“Other installation positions”*.

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

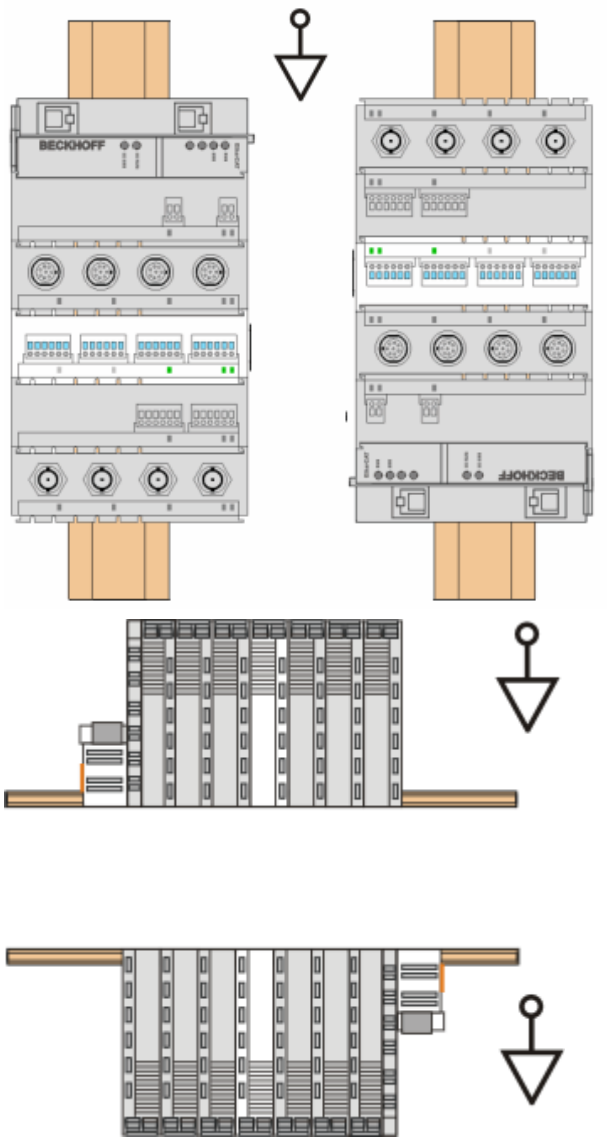


Fig. 130: *Other installation positions*

7.7 Mounting of Passive Terminals

Hint for mounting passive terminals

EtherCAT Bus Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called Passive Terminals. The Passive Terminals have no current consumption out of the E-Bus To ensure an optimal data transfer, you must not directly string together more than 2 Passive Terminals!

Examples for mounting passive terminals (highlighted)

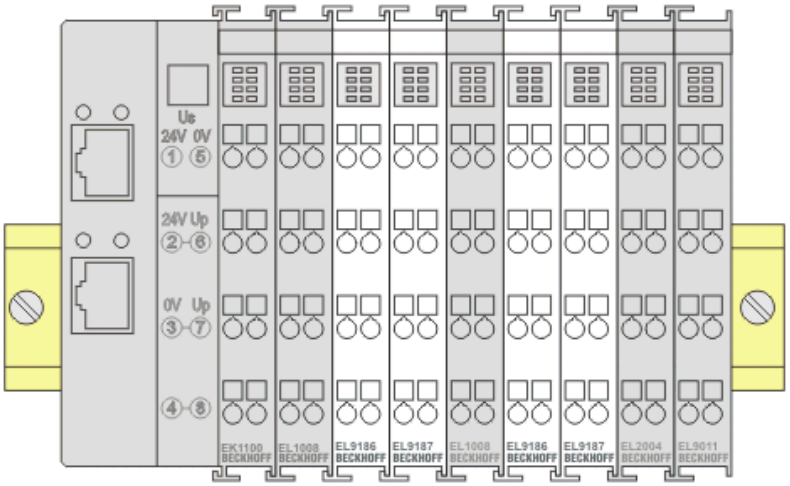


Fig. 131: *Correct configuration*

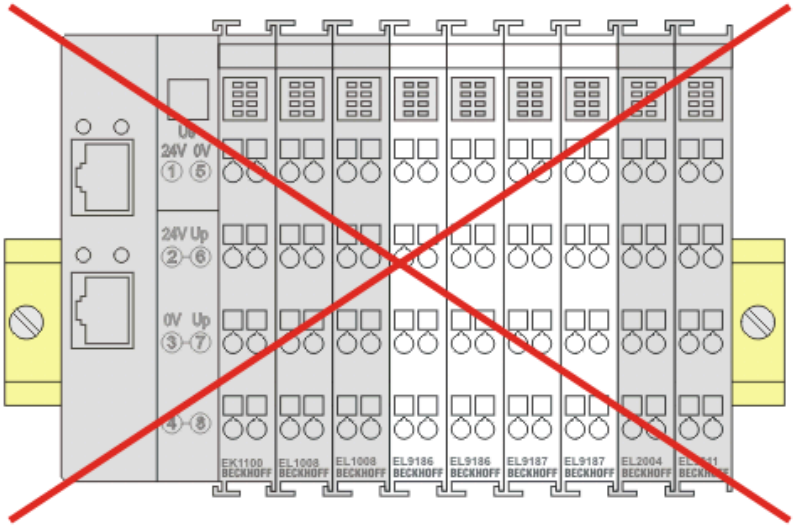


Fig. 132: *Incorrect configuration*

7.8 Shielding concept

Together with the shield busbar, the prefabricated cables from Beckhoff Automation offer optimum protection against electromagnetic interference.
It is highly recommended to apply the shield as close as possible to the terminal, in order to minimize operational disturbances.

Connection of the motor cable to the shield busbar

Fasten the shield busbar supports 1 to the DIN rail 2. The mounting rail 2 must be in contact with the metallic rear wall of the control cabinet over a wide area. Install the shield busbar 3 as shown below.
As an alternative, a shield busbar clamp 3a can be screwed directly to the metallic rear wall of the control cabinet (fig. “shield busbar clamp”)

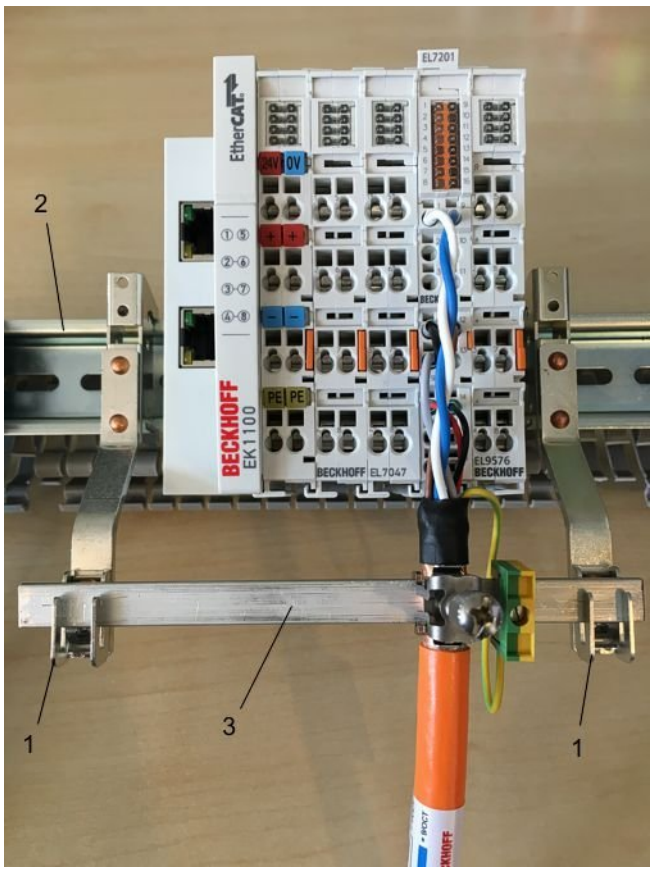


Fig. 133: Shield busbar

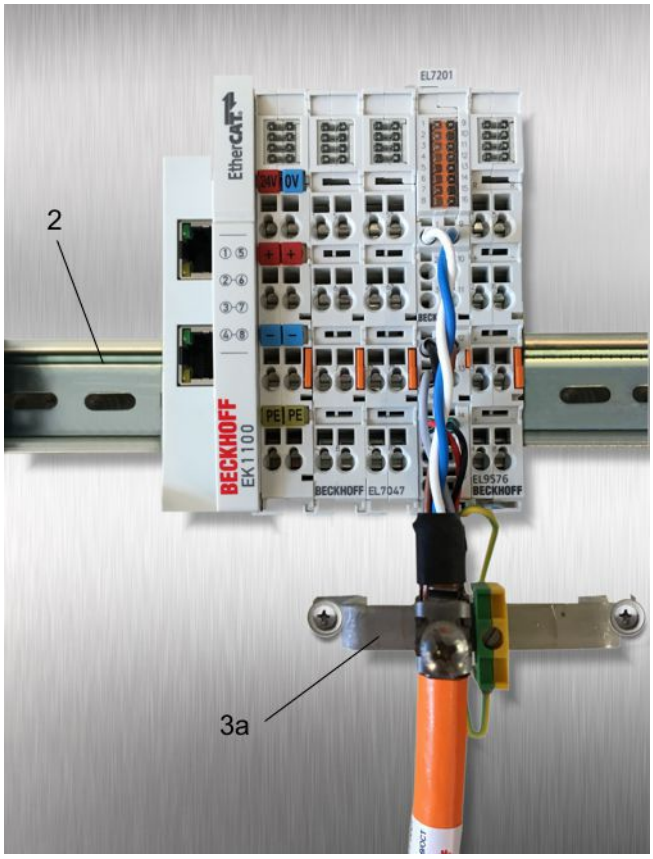


Fig. 134: Shield busbar clamp

Connect the cores 4 of the motor cable 5, then attach the copper-sheathed end 6 of the motor cable 5 with the shield clamp 7 to the shield busbar 3 or shield busbar clamp 3a. Tighten the screw 8 to the stop.
Fasten the PE clamp 9 to the shield busbar 3 or shield busbar clamp 3a. Clamp the PE core 10 of the motor cable 5 under the PE clamp 9.

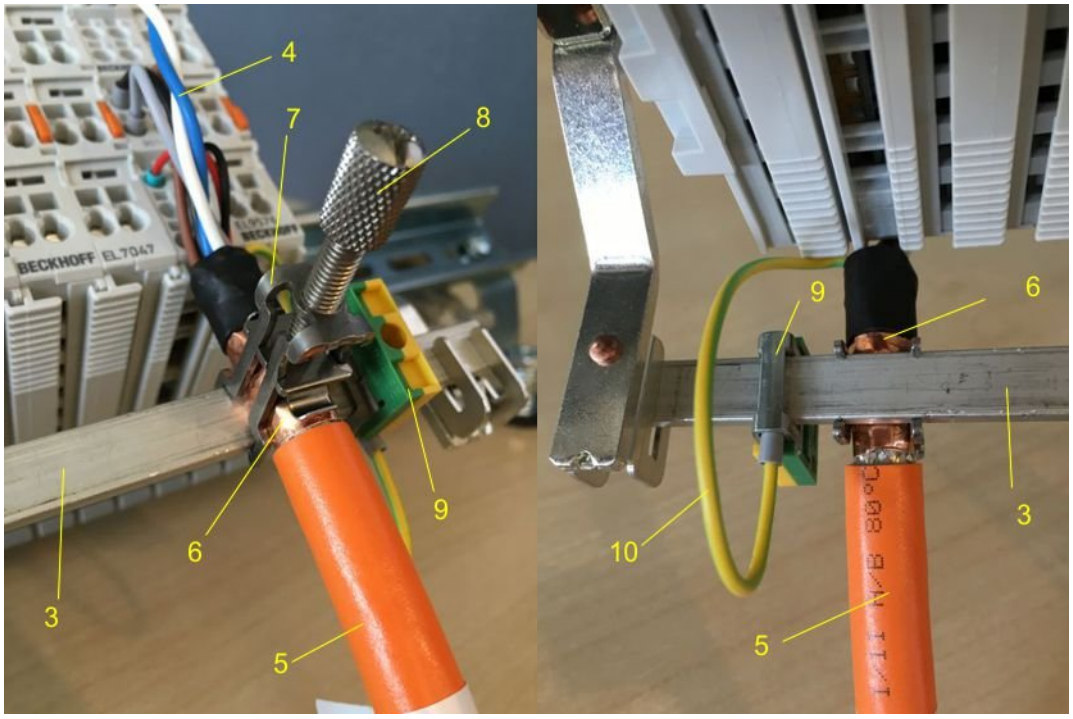


Fig. 135: Shield connection

Connection of the feedback cable to the motor

Twisting of the feedback cable cores

The feedback cable cores should be twisted, in order to avoid operational disturbances.

When screwing the feedback plug to the motor, the shield of the feedback cable is connected via the metallic plug fastener.
On the terminal side the shield can also be connected. Connect the cores of the feedback cable and attach the copper-sheathed end of the feedback cable to the shield busbar 3 or shield busbar clamp 3a with the shield clamp 7. The motor cable and the feedback cable can be connected to the shield clamp 7 with the screw 8.

7.9 Power supply, potential groups

Electrical isolations for EL2642, ELM264x, ELM274x

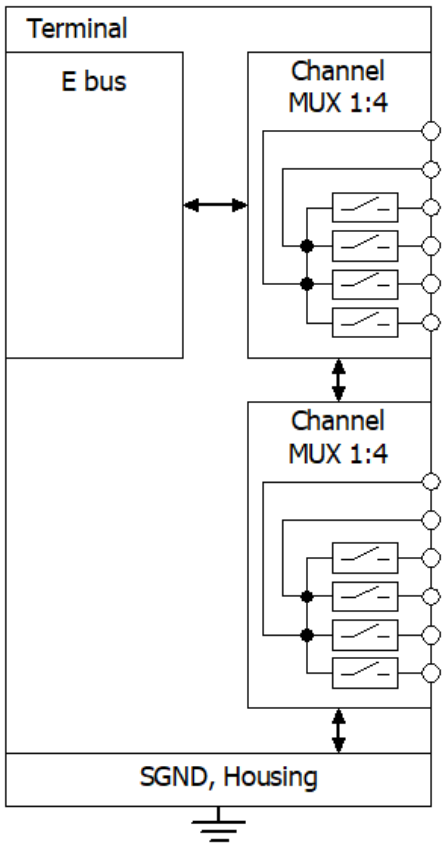


Fig. 136: Representation of the electrical isolations I/O to supply for EL2642, ELM264x, ELM274x

7.10 ELM/EKM terminal mounting on DIN rail

⚠ 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!

Assembly

The ELM terminals are locked to commercially available 35 mm mounting rails (DIN rails according to EN 60715) as following described:

- The ELM terminal can easily be latched onto the DIN rail. Therefore the clips of the terminal on top and down side have to be opened first:

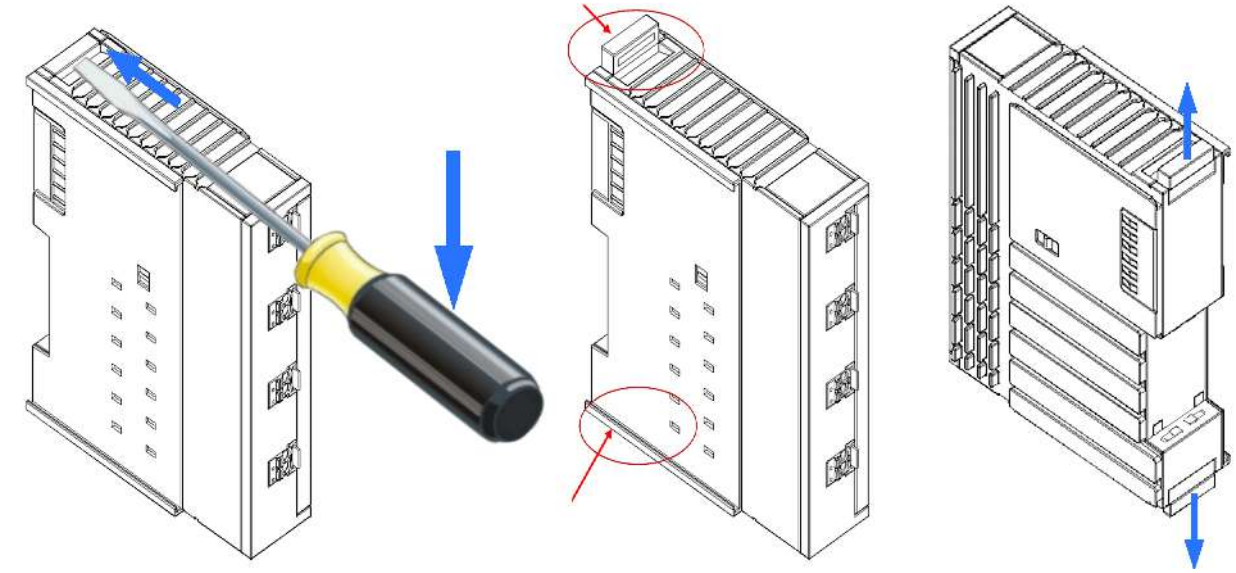


Fig. 137: Opening the clips on top and down side by lifting them e.g. with a screw driver

- Insert the ELM terminal to other already on the DIN rail arranged moduls together with tongue and groove and push the terminals against the mounting rail, until it clicks onto the touchdown point of the mounting rail. Then close the both clips on top and down side of the terminal respectively:

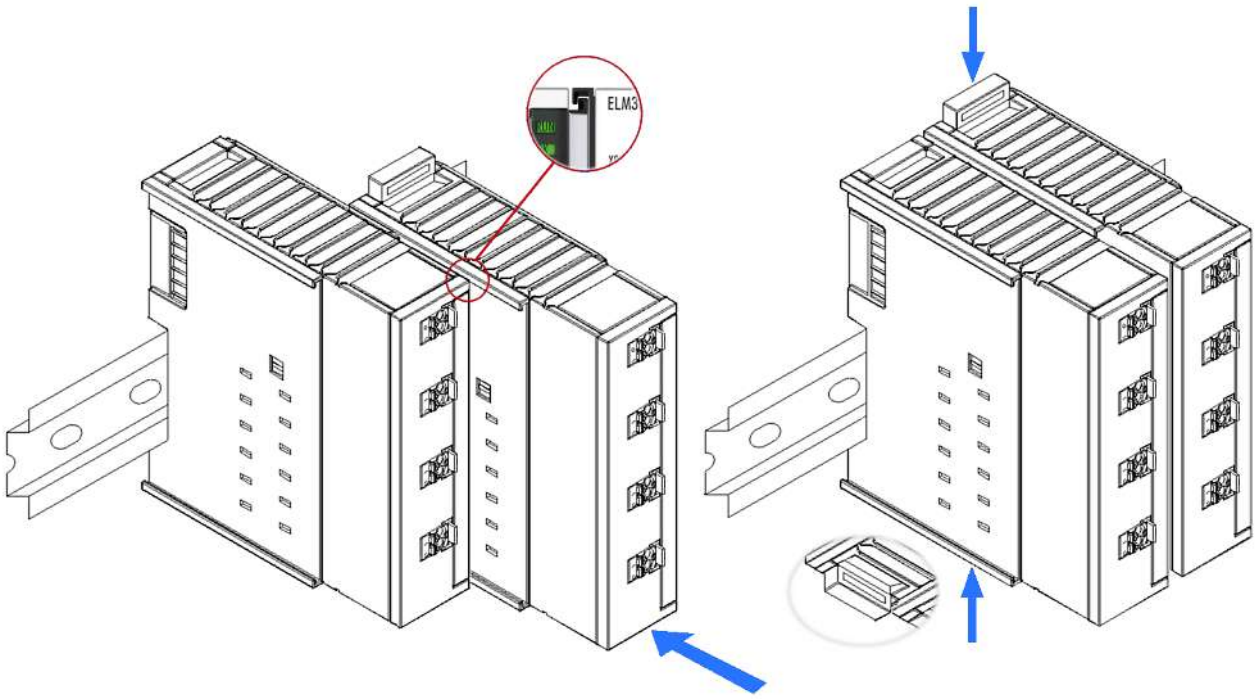
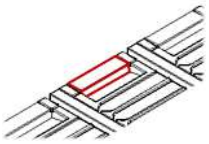


Fig. 138: Push-in of the ELM terminal and closing the mounting rail clips top and down

- During closing of the both clips there mustn't be a disruptive mechanical resistance being noticeable. The clips have to be snapped so that they're ending flat with the housing:



Attention: If the ELM terminal is 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.

Disassembly

Each ELM terminal is secured by a lock on the mounting rail, which must be released for disassembly. The procedure for demounting have to be done in *reverse* order as described in Assembly [▶ 135]:

1. Release the mounting rail lock of the ELM terminal on the top and down side 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.

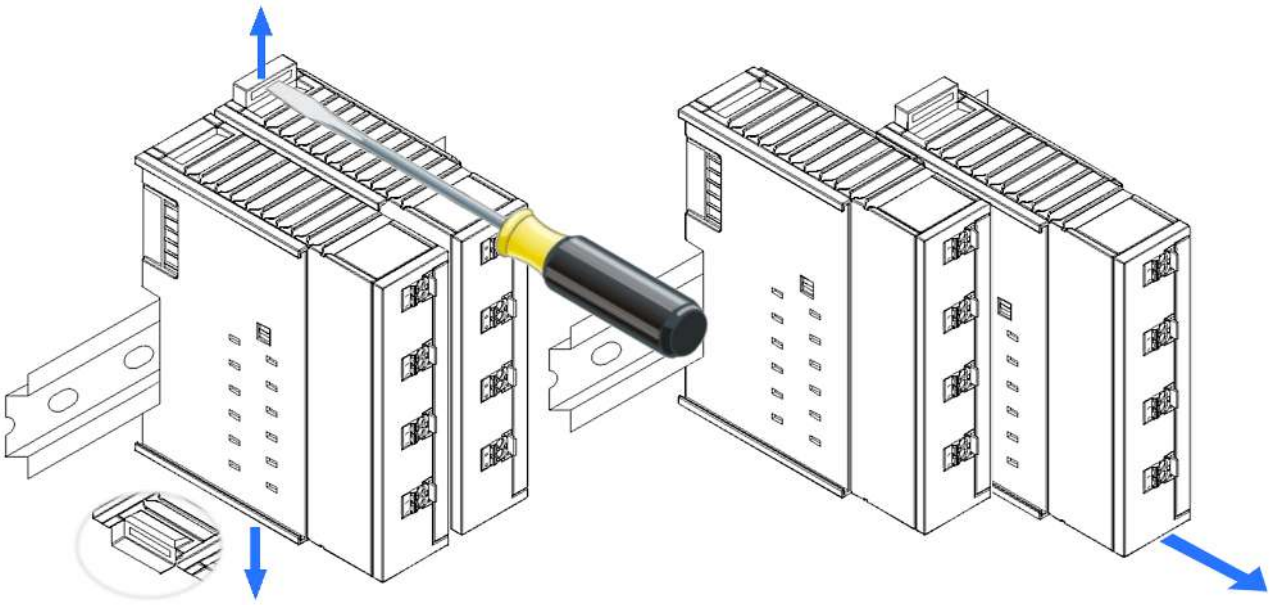


Fig. 139: Opening of the upper and lower mounting rail lock and pull out the ELM terminal module

Connections within a bus terminal block

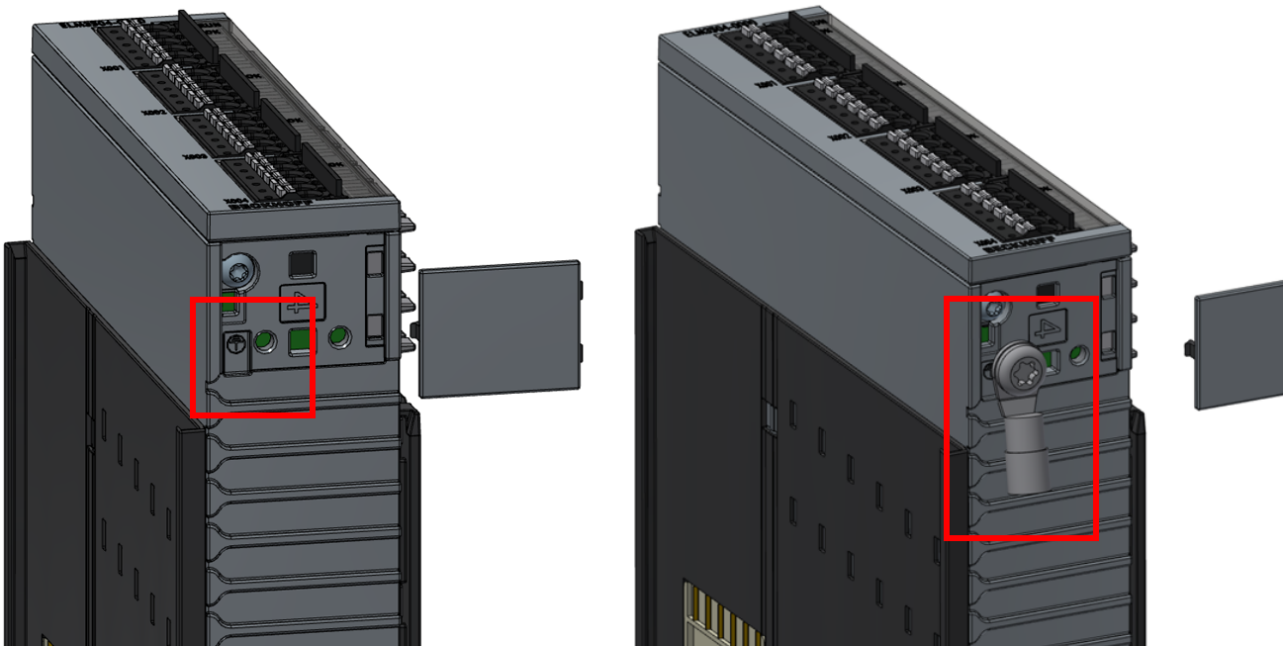
The electric connections between the Bus Coupler and the Bus Terminals are automatically realized by joining the components: The six spring contacts of the K-Bus/E-Bus deal with the transfer of the data and the supply of the Bus Terminal electronics.

7.11 Protective earth (PE)

The housings of the ELM/EKM series are made of die-cast zinc and are thus metallic. This results in a need for clarification regarding the use of protective earthing against the risk of electric shock.

Attention: The relevant application standards refer to the surrounding control cabinet/control box as "housing", whereas this documentation refers to the Beckhoff terminal as "housing".

See also chapter "Notes regarding analog equipment - shielding and earth" in this documentation.



The housing offers the option of an M3 bolted connection for connecting a ring terminal to PE.

The procedure for this is as follows:

- Lever off the plastic cover from the ELM housing and retain if for later reuse, if required
- Secure the previously prepared ring terminal, which was crimped to the protective conductor, using an M3x4 screw; max. torque 0.5 Nm. Use a suitable tool.
  ATTENTION: The screw must not be longer than specified, in order to avoid it protruding into the interior, where it could cause damage. This would be evident if the unit is sent in for repair.
- Connect the PE cable to the protective conductor system.

Notes on whether a PE connection is necessary in the specific application

- A PE connection is required if the terminal could pose a risk of electric shock due to an inadmissible contact voltage. A distinction is made between two causes:
  - if the terminal is subjected to high internal voltages (not SELV/PELV), this high voltage may reach the housing in the event of a fault. For such terminals, a PE connection is essential. See the corresponding mechanical options at the module. For background information please refer to product and device standards such as EN 61010.
    Note: The terminals of type ELM3004, ELM3002, ELM3104, ELM3102, ELM3504, ELM3502, ELM3604, ELM3602, ELM3704, ELM3702 operate with low voltage SELV/PELV, so that there is usually no potential risk.
  - A connection to the protective earth conductor system must nevertheless be provided if the terminal operates with protective extra-low voltage (SELV/PELV), but there is a risk that a live conductor may come into contact with the housing in the event of a fault, resulting in unacceptable touch voltage. This is stipulated by application standards such as EN60204-1 or EN61439-1 relating to control cabinet design.
- It is therefore always necessary to check in which environment the application is used to ascertain whether a PE connection is required.

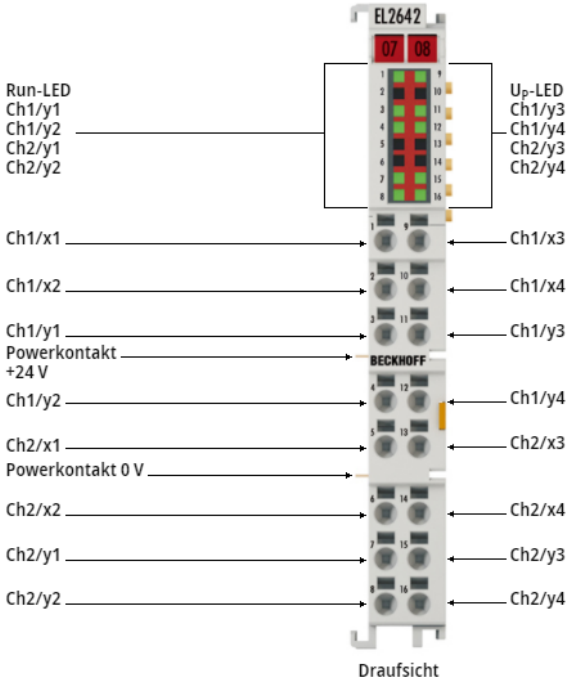
Note on protective earth (PE) with regard to analog measurements

The protective earth conductor system is specifically designed for discharging high currents. This may result in significant high-frequency interference, which could adversely affect an analog measuring device if it is/has to be connected to the protective conductor system. In such cases, a strictly star-shaped configuration of the FE and PE systems may be advisable, in order to have as few interference sources as possible on the

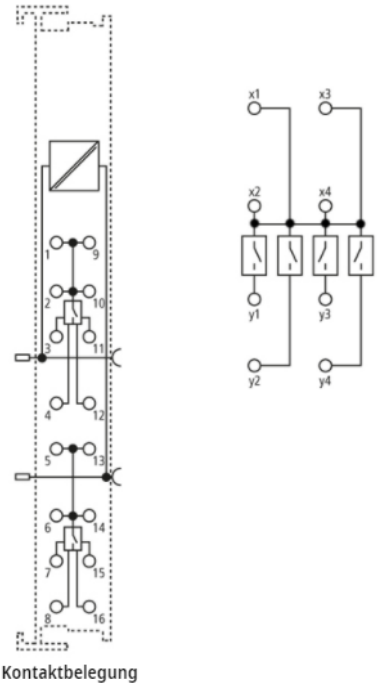
PE system that are close to the analog measuring system. Ideally, no PE connection should be used at all. However, in this case the installation must comply with the two conditions referred to above, which may necessitate splitting the system into a high-voltage and a low-voltage control cabinet, so that no PE would be required for the latter.

7.12 LEDs and connection

7.12.1 EL2642



Draufsicht



Kontaktbelegung

LEDs EL2642

LED	Color	Meaning
UP-LED	green	24V power contact voltage is present
	off	No power supply present
Ch(n) / y(m), n=1, 2 m=1..4	green	*Signal passing through (switch closed)*
	off	*Signal not passing (switch open)*

LED	Color	Meaning	
RUN	green	off	State of the EtherCAT State Machine [▶ 111]: **INIT** = initialization 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** = check the channels of the Sync Manager [▶ 101] and the Distributed Clocks [▶ 118] (if supported)
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
		flickering	State of the EtherCAT State Machine: **BOOTSTRAP** = function for firmware updates [▶ 144] of the terminal

Pin assignment EL2642

Terminal point		Description
Name	No.	
Ch1 / x1	1	Channel 1, input x1
Ch1 / x2	2	Channel 1, input x2
Ch1 / y1	3	Channel 1, output y1

Terminal point		Description
Name	No.	
Ch1 / y2	4	Channel 1, output y2
Ch2 / x1	5	Channel 2, input x1
Ch2 / x2	6	Channel 2, input x2
Ch2 / y1	7	Channel 2, output y1
Ch2 / y2	8	Channel 2, output y2
Ch1 / x3	9	Channel 1, input x3
Ch1 / x4	10	Channel 1, input x4
Ch1 / y3	11	Channel 1, output y3
Ch1 / y4	12	Channel 1, output y4
Ch2 / x3	13	Channel 2, input x3
Ch2 / x4	14	Channel 2, input x4
Ch2 / y3	15	Channel 2, output y3
Ch2 / y4	16	Channel 2, output y4

7.12.2 ELM264x/ ELM274x

Fig. 140: ELM264x/ ELM274x Connections and LEDs

LED	Color	Meaning
UP (1…n)	green	24V power contact voltage is present
	off	No power supply present
ON (1..n, m)	green	*Signal passing through (switch closed)*
	off	*Signal not passing (switch open)*

LED	Color	Meaning	
RUN	green	off	State of the EtherCAT State Machine [▶ 111]: **INIT** = initialization 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** = check the channels of the Sync Manager [▶ 101] and the Distributed Clocks [▶ 118] (if supported)
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
		flickering	State of the EtherCAT State Machine: **BOOTSTRAP** = function for firmware updates [▶ 144] of the terminal

Pin assignment ELM264x / ELM274x

Connector X1, X2 (ELM2642, ELM2742) X1...X4 (ELM2644, ELM2744)		Description
Name	Terminal point No.	
Z1.1	1	Input Z1.1
Z1.2	2	Input Z1.2
Y1	3	Output Y1
Y2	4	Output Y2
Y3	5	Output Y3
Y4	6	Output Y4

7.13 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.

8 Appendix

8.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

8.2 Firmware Update EL/ES/EM/ELM/EPxxxx

This section describes the device update for Beckhoff EtherCAT slaves from the EL/ES, ELM, EM, EK and EP series. A firmware update should only be carried out after consultation with Beckhoff support.

NOTE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website https://www.beckhoff.com/en-us/.

To update the firmware, TwinCAT can be operated in the so-called FreeRun mode, a paid license is not required.

The device to be updated can usually remain in the installation location, but TwinCAT has to be operated in the FreeRun. Please make sure that EtherCAT communication is trouble-free (no LostFrames etc.).

Other EtherCAT master software, such as the EtherCAT Configurator, should not be used, as they may not support the complexities of updating firmware, EEPROM and other device components.

Storage locations

An EtherCAT slave stores operating data in up to three locations:

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.
- In addition, each EtherCAT slave has a memory chip, a so-called **ESI-EEPROM**, for storing its own device description (ESI: EtherCAT Slave Information). On power-up this description is loaded and the EtherCAT communication is set up accordingly. The device description is available from the download area of the Beckhoff website at (https://www.beckhoff.com). All ESI files are accessible there as zip files.

Customers can access the data via the EtherCAT fieldbus and its communication mechanisms. Acyclic mailbox communication or register access to the ESC is used for updating or reading of these data.

The TwinCAT System Manager offers mechanisms for programming all three parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

Simplified update by bundle firmware

The update using so-called **bundle firmware** is more convenient: in this case the controller firmware and the ESI description are combined in a *.efw file; during the update both the firmware and the ESI are changed in the terminal. For this to happen it is necessary

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_REV0016_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

- ESI/Revision: e.g. by means of an online scan in TwinCAT ConfigMode/FreeRun – this is a convenient way to determine the revision
- Firmware: e.g. by looking in the online CoE of the device

NOTE

Risk of damage to the device!

✓ Note the following when downloading new device files

a) Firmware downloads to an EtherCAT device must not be interrupted

b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.

c) The power supply must adequately dimensioned. The signal level must meet the specification.

⇨ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

8.2.1 Device description ESI file/XML

NOTE

Attention regarding update of the ESI description/EEPROM

Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

The ESI device description is stored locally on the slave and loaded on start-up. Each device description has a unique identifier consisting of slave name (9 characters/digits) and a revision number (4 digits). Each slave configured in the System Manager shows its identifier in the EtherCAT tab:

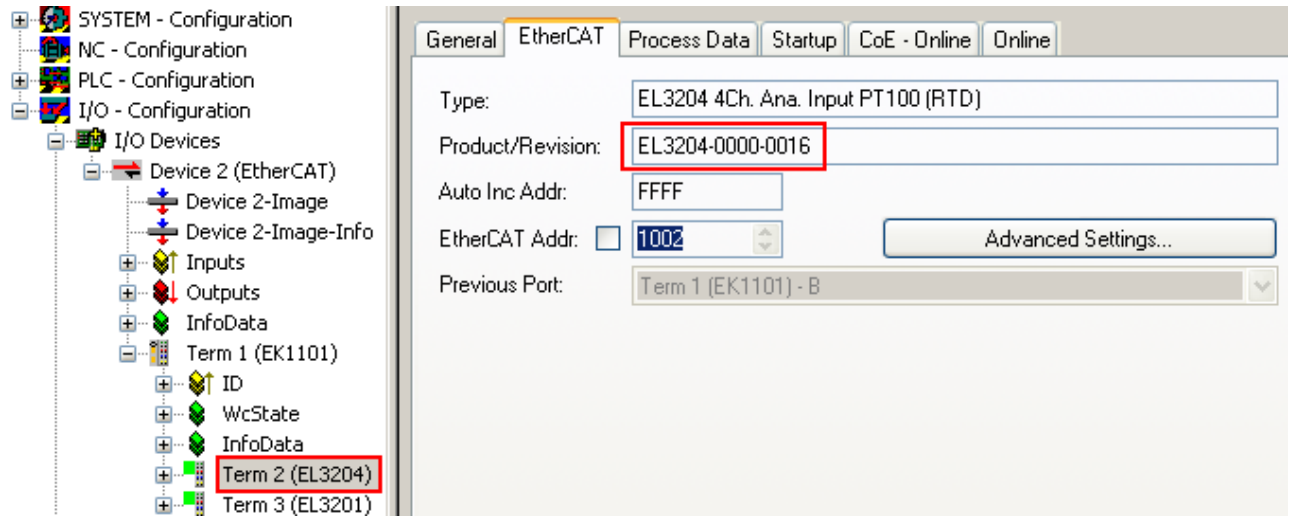


Fig. 141: Device identifier consisting of name EL3204-0000 and revision -0016

The configured identifier must be compatible with the actual device description used as hardware, i.e. the description which the slave has loaded on start-up (in this case EL3204). Normally the configured revision must be the same or lower than that actually present in the terminal network.

For further information on this, please refer to the EtherCAT system documentation.

Update of XML/ESI description

The device revision is closely linked to the firmware and hardware used. Incompatible combinations lead to malfunctions or even final shutdown of the device. Corresponding updates should only be carried out in consultation with Beckhoff support.

Display of ESI slave identifier

The simplest way to ascertain compliance of configured and actual device description is to scan the EtherCAT boxes in TwinCAT mode Config/FreeRun:

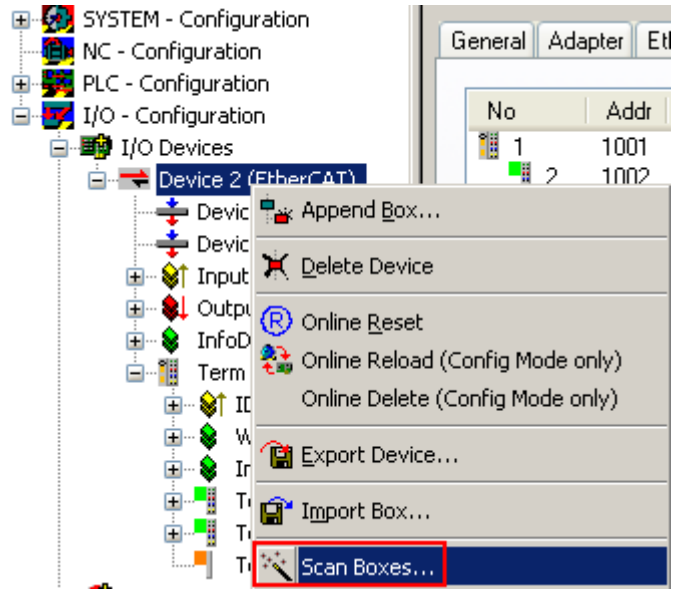


Fig. 142: Scan the subordinate field by right-clicking on the EtherCAT device

If the found field matches the configured field, the display shows

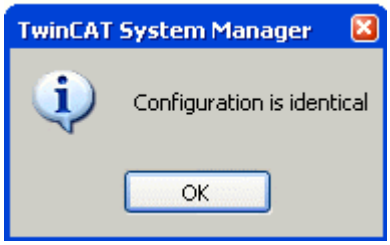


Fig. 143: Configuration is identical

otherwise a change dialog appears for entering the actual data in the configuration.

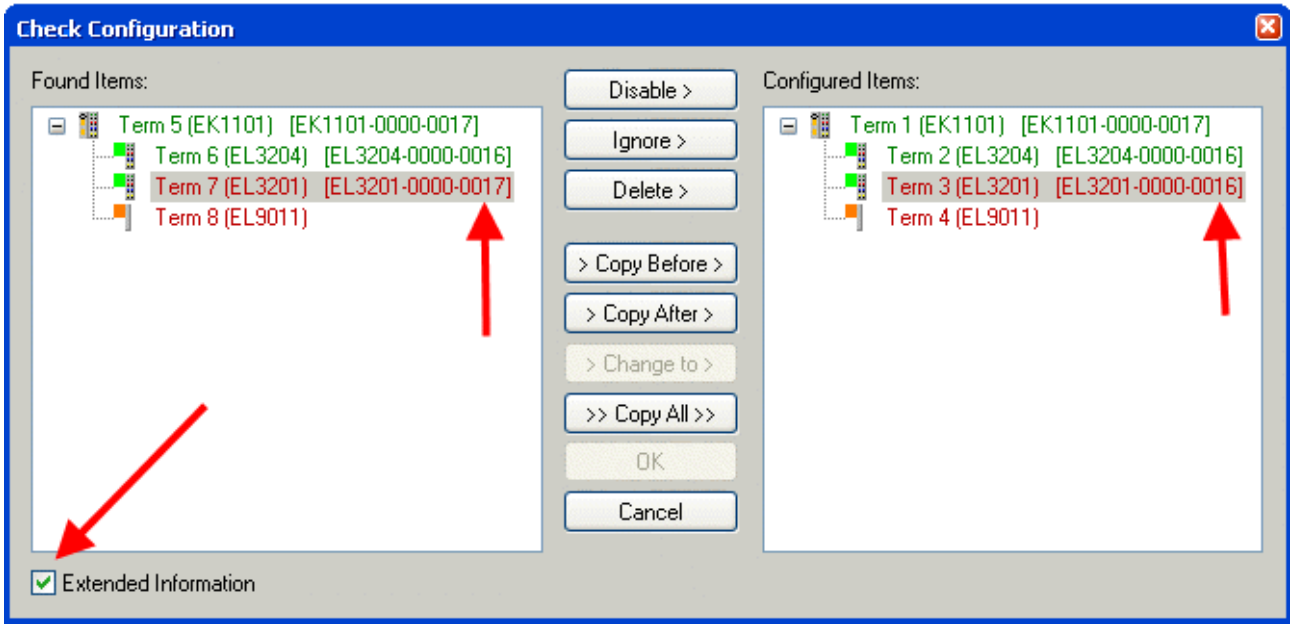


Fig. 144: Change dialog

In this example in Fig. *Change dialog*, an EL3201-0000-**0017** was found, while an EL3201-0000-**0016** was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*

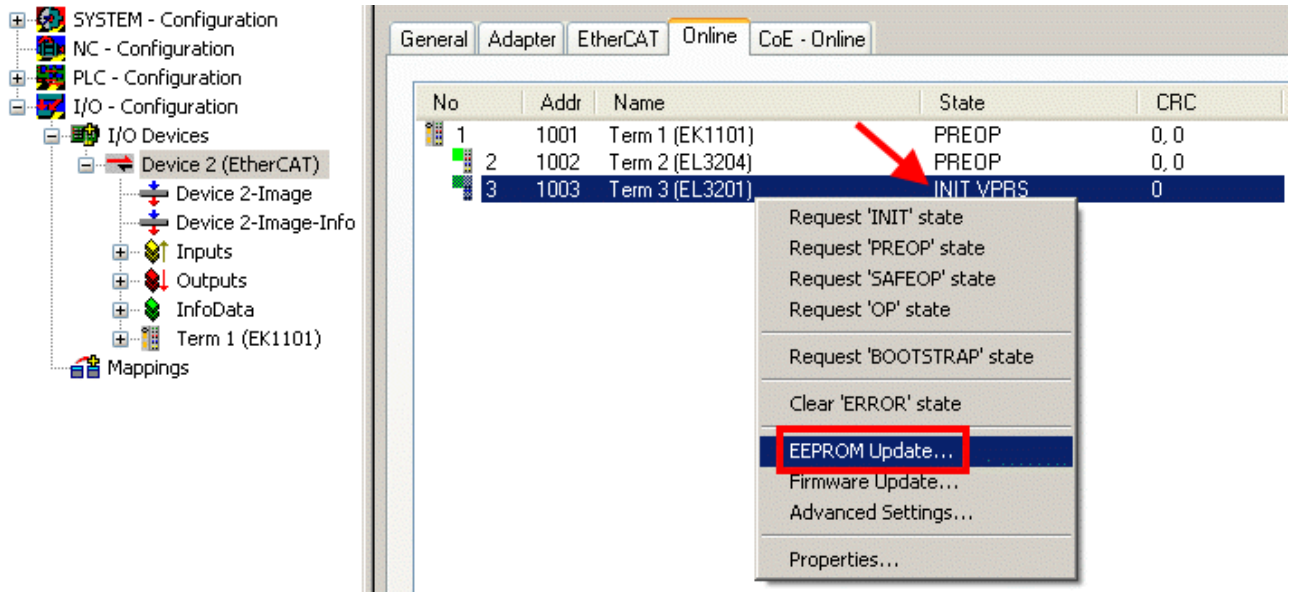


Fig. 145: EEPROM Update

The new ESI description is selected in the following dialog, see Fig. *Selecting the new ESI.* The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.

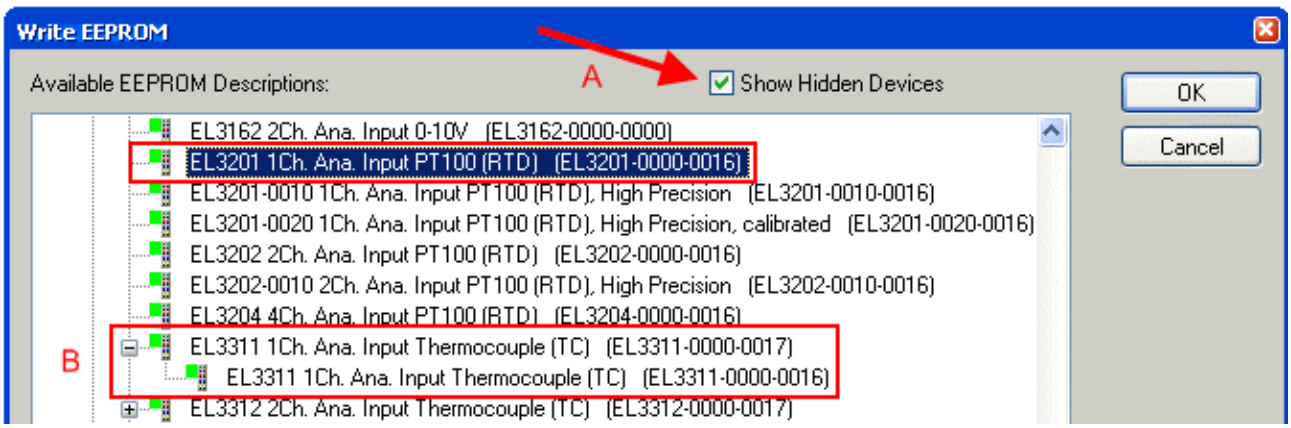


Fig. 146: Selecting the new ESI

A progress bar in the System Manager shows the progress. Data are first written, then verified.

The change only takes effect after a restart.

Most EtherCAT devices read a modified ESI description immediately or after startup from the INIT. Some communication settings such as distributed clocks are only read during power-on. The EtherCAT slave therefore has to be switched off briefly in order for the change to take effect.

8.2.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

CoE Online and Offline CoE

Two CoE directories are available:
• **online**: This is offered in the EtherCAT slave by the controller, if the EtherCAT slave supports this. This CoE directory can only be displayed if a slave is connected and operational.
• **offline**: The EtherCAT Slave Information ESI/XML may contain the default content of the CoE. This CoE directory can only be displayed if it is included in the ESI (e.g. “Beckhoff EL5xxx.xml”).

The Advanced button must be used for switching between the two views.

In Fig. *Display of EL3204 firmware version* the firmware version of the selected EL3204 is shown as 03 in CoE entry 0x100A.

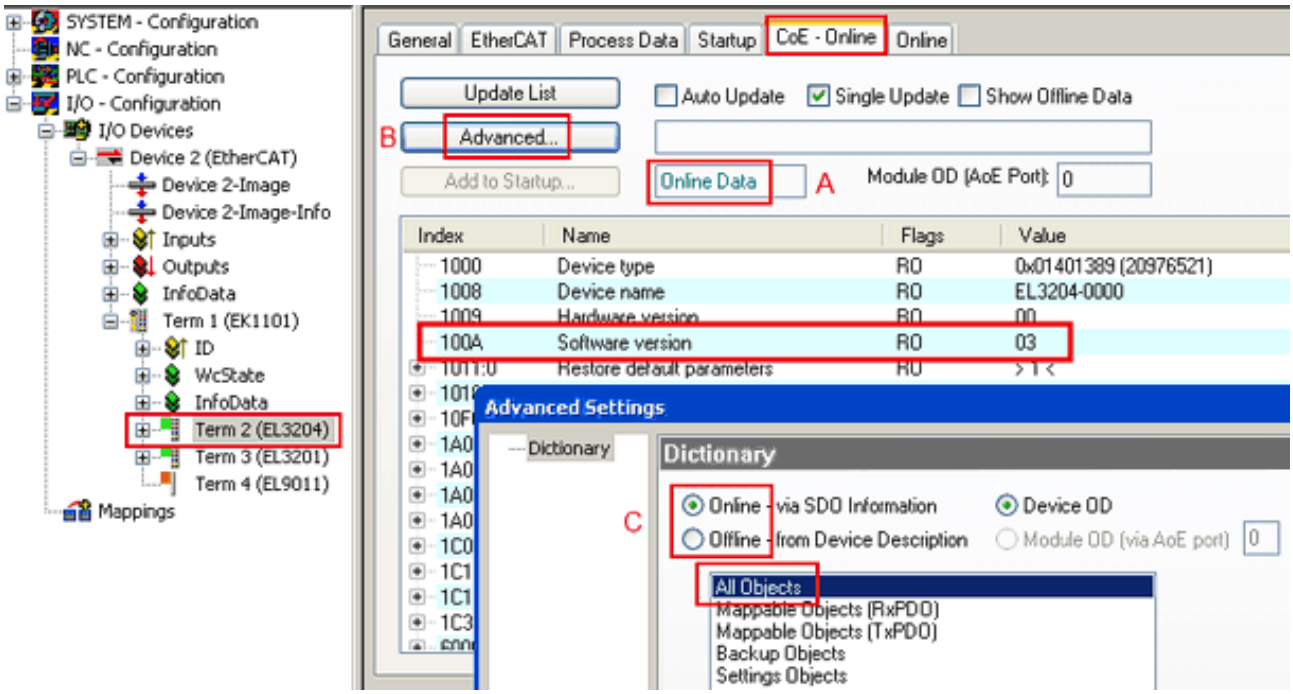


Fig. 147: Display of EL3204 firmware version

In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

8.2.3 Updating controller firmware *.efw

CoE directory

The Online CoE directory is managed by the controller and stored in a dedicated EEPROM, which is generally not changed during a firmware update.

Switch to the *Online* tab to update the controller firmware of a slave, see Fig. *Firmware Update*.

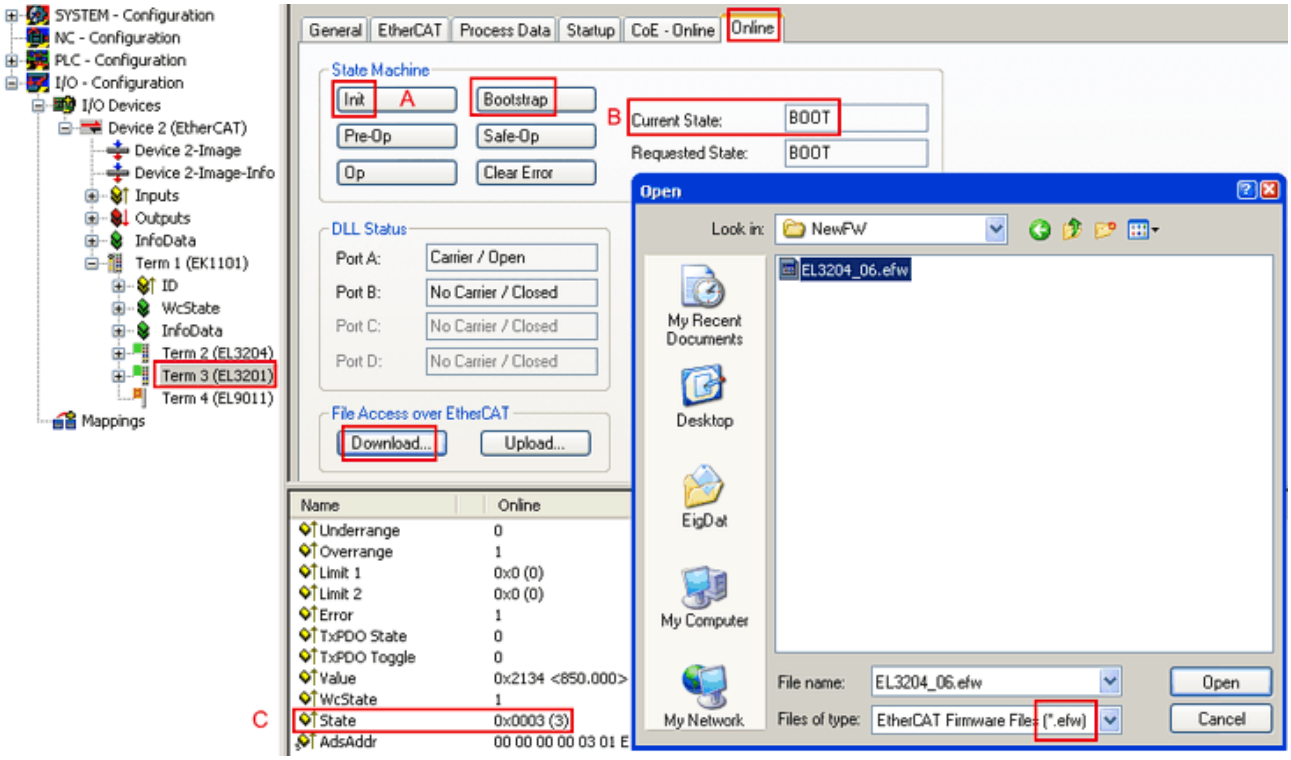
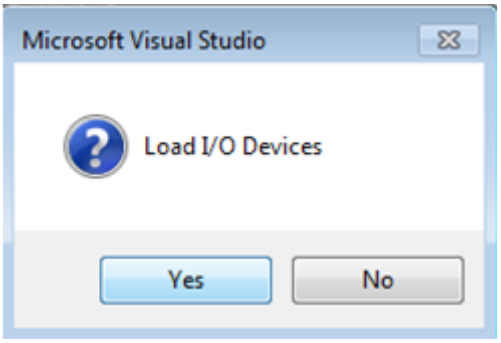
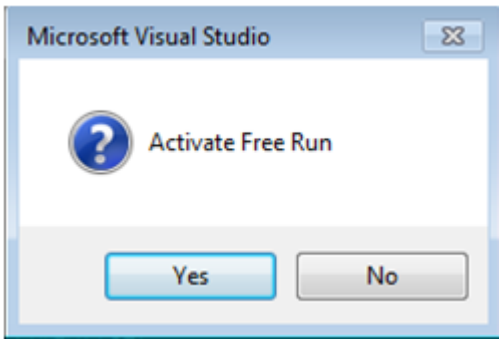


Fig. 148: Firmware Update

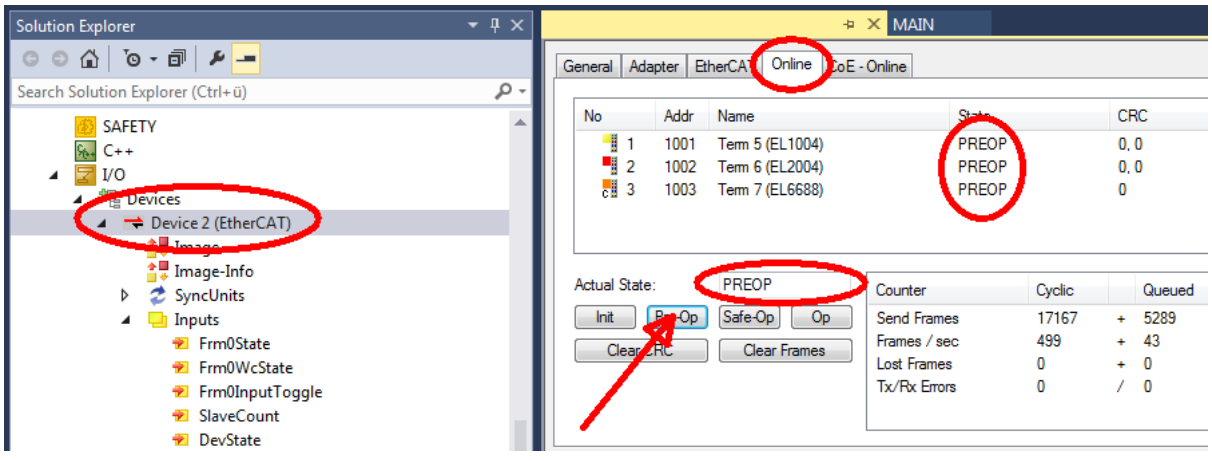
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time >= 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.



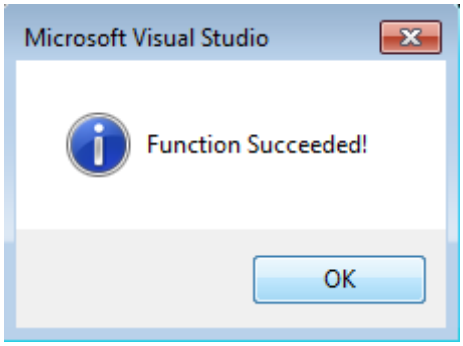


- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A password will not be necessary usually.



- After the download switch to INIT, then PreOP
- Switch off the slave briefly (don't pull under voltage!)
- Check within CoE 0x100A, if the FW status was correctly overtaken.

8.2.4 FPGA firmware *.rbf

If an FPGA chip deals with the EtherCAT communication an update may be accomplished via an *.rbf file.

- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

The firmware version number included in the terminal serial number contains both firmware components. If one of these firmware components is modified this version number is updated.

Determining the version via the System Manager

The TwinCAT System Manager indicates the FPGA firmware version. Click on the Ethernet card of your EtherCAT strand (Device 2 in the example) and select the *Online* tab.

The *Reg:0002* column indicates the firmware version of the individual EtherCAT devices in hexadecimal and decimal representation.

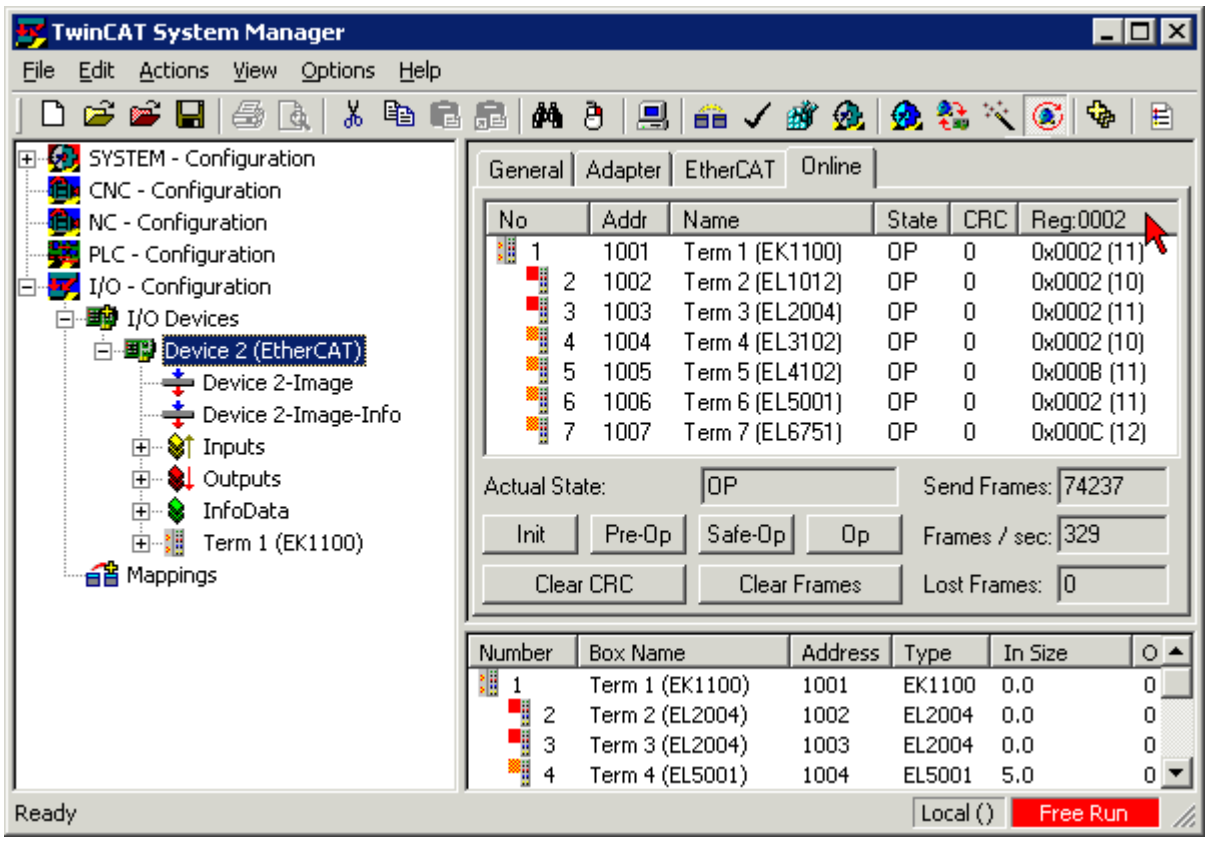


Fig. 149: FPGA firmware version definition

If the column *Reg:0002* is not displayed, right-click the table header and select *Properties* in the context menu.

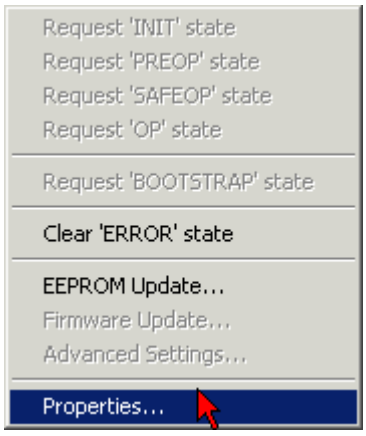


Fig. 150: Context menu *Properties*

The *Advanced Settings* dialog appears where the columns to be displayed can be selected. Under *Diagnosis/***Online View** select the *'0002 ETxxxx Build'* check box in order to activate the FPGA firmware version display.

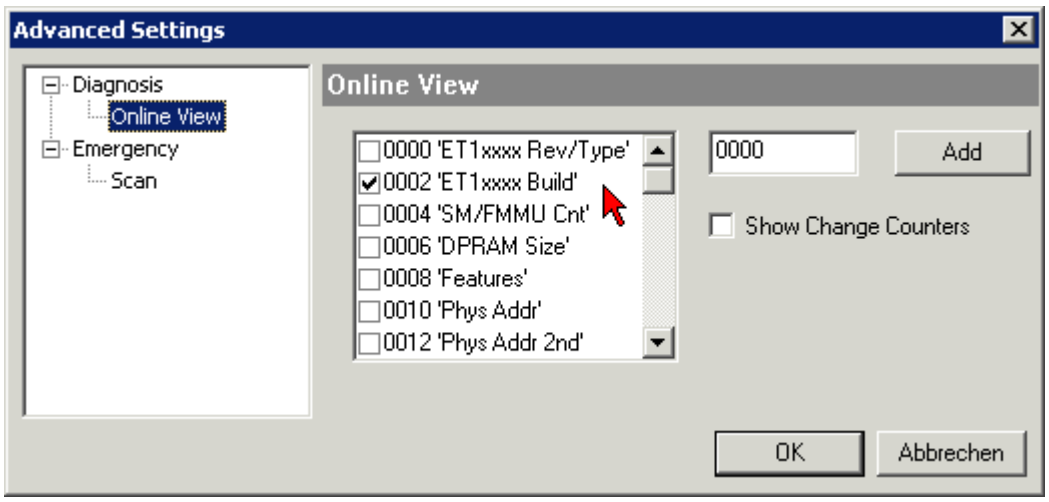


Fig. 151: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

- of an EtherCAT coupler the coupler must have FPGA firmware version 11 or higher;
- of an E-Bus Terminal the terminal must have FPGA firmware version 10 or higher.

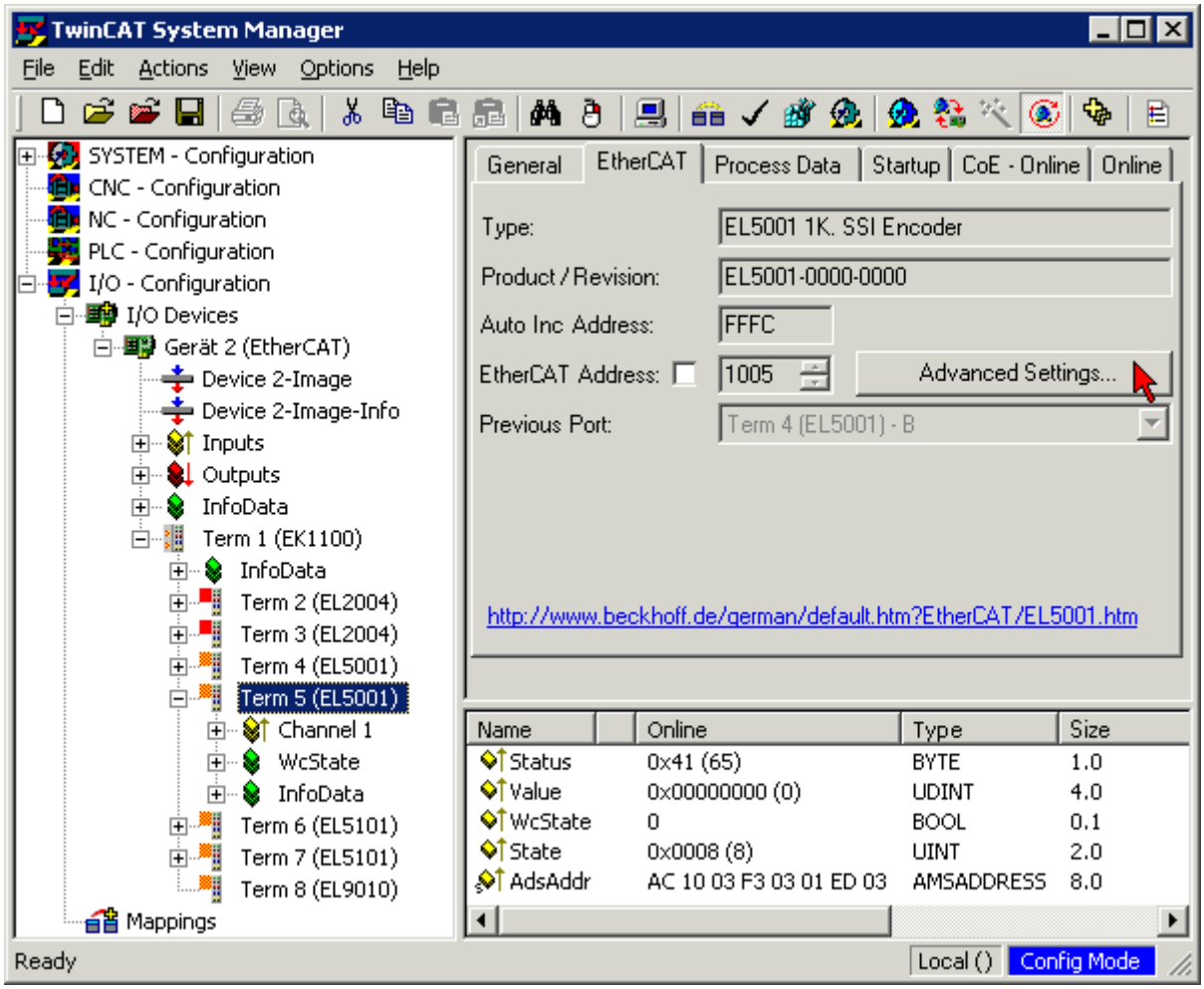
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

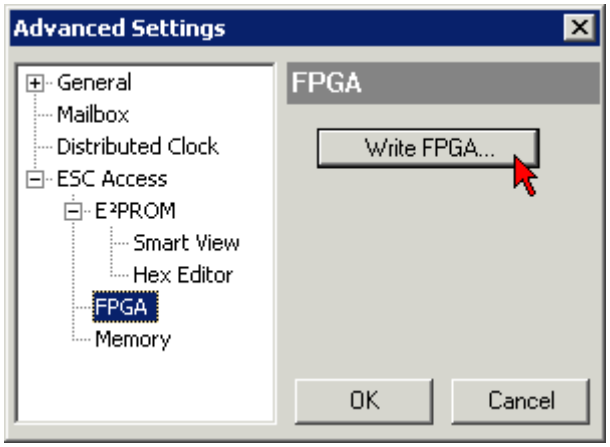
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- Switch TwinCAT system to ConfigMode/FreeRun with cycle time >= 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.

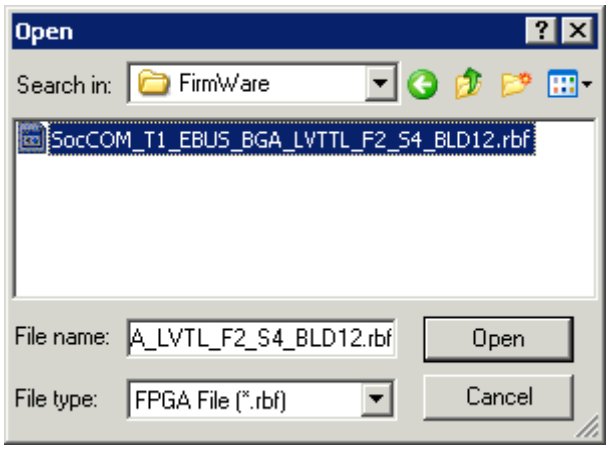
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
  click the *Advanced Settings* button in the *EtherCAT* tab:



- The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM*/FPGA click on *Write FPGA* button:



- Select the file (*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device:



- Wait until download ends
- Switch slave current less for a short time (don't pull under voltage!). In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.
- Check the new FPGA status

NOTE

Risk of damage to the device!

A download of firmware to an EtherCAT device must not be interrupted in any case! If you interrupt this process by switching off power supply or disconnecting the Ethernet link, the EtherCAT device can only be recommissioned by the manufacturer!

8.2.5 Simultaneous updating of several EtherCAT devices

The firmware and ESI descriptions of several devices can be updated simultaneously, provided the devices have the same firmware file/ESI.

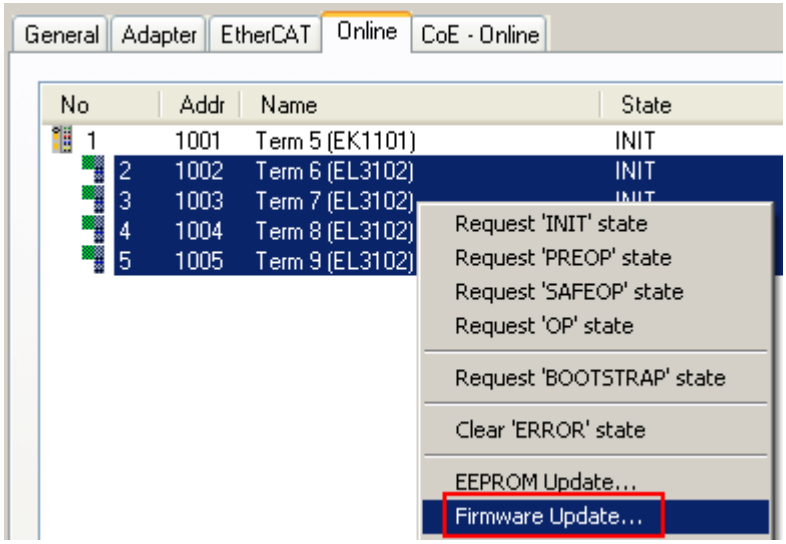


Fig. 152: Multiple selection and firmware update

Select the required slaves and carry out the firmware update in BOOTSTRAP mode as described above.

8.3 Firmware compatibility

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

Note

- It is recommended to use the newest possible firmware for the respective hardware.
- Beckhoff is not under any obligation to provide customers with free firmware updates for delivered products.

NOTE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 144]. If a device is placed in BOOTSTRAP mode for a firmware update, it does not check when downloading whether the new firmware is suitable. This can result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL2642			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00*	01*	0016	03/2021

ELM2642			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
01*	01*	0016	12/2020

ELM2644			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
01*	01*	0016	12/2020

ELM2742			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
01*	01*	0016	12/2020

ELM2744			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
01*	01*	0016	12/2020

[1]) This is the current compatible firmware/hardware version at the time of the preparing this documentation. Check on the Beckhoff web page whether more up-to-date documentation is available.

8.4 Firmware compatibility - passive terminals

The passive terminals ELxxxx terminal series have no firmware to update.

8.5 Restoring the delivery state

To restore the delivery state (factory settings) for backup objects in ELxxxx terminals, the CoE object Restore default parameters, *SubIndex 001* can be selected in the TwinCAT System Manager (Config mode) (see Fig. *Selecting the Restore default parameters PDO*)

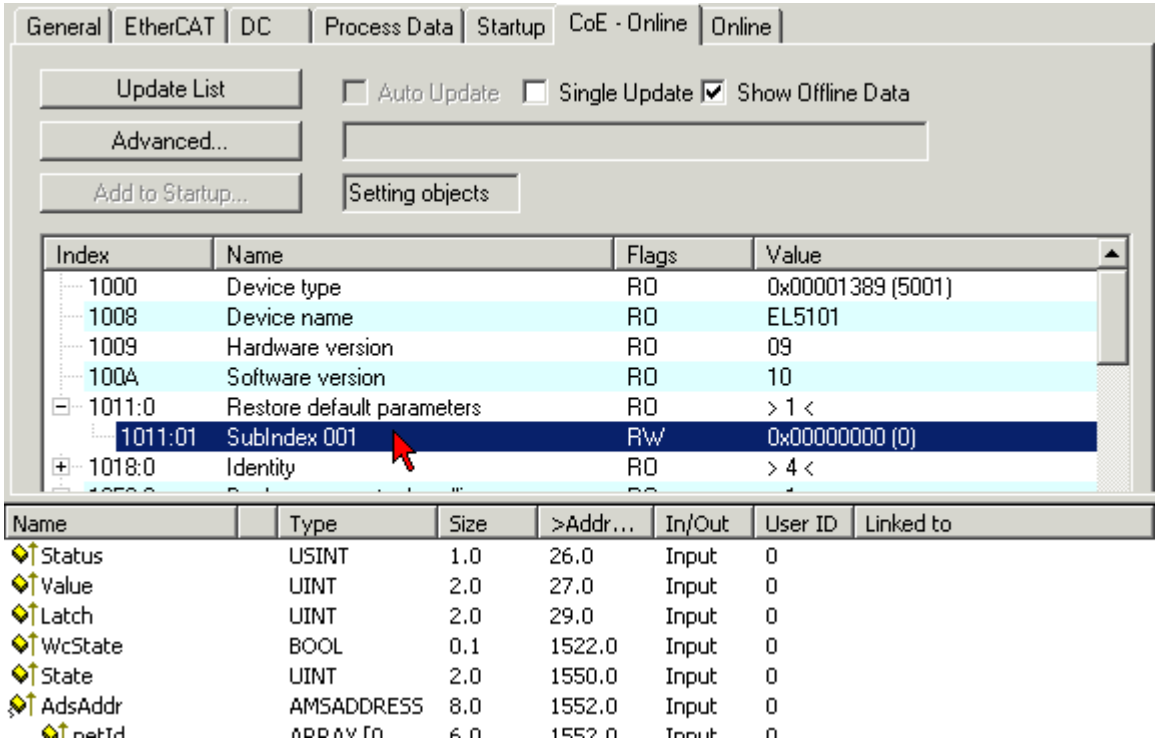


Fig. 153: Selecting the *Restore default parameters* PDO

Double-click on SubIndex 001 to enter the Set Value dialog. Enter the value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* and confirm with *OK* (Fig. *Entering a restore value in the Set Value dialog*). All backup objects are reset to the delivery state.

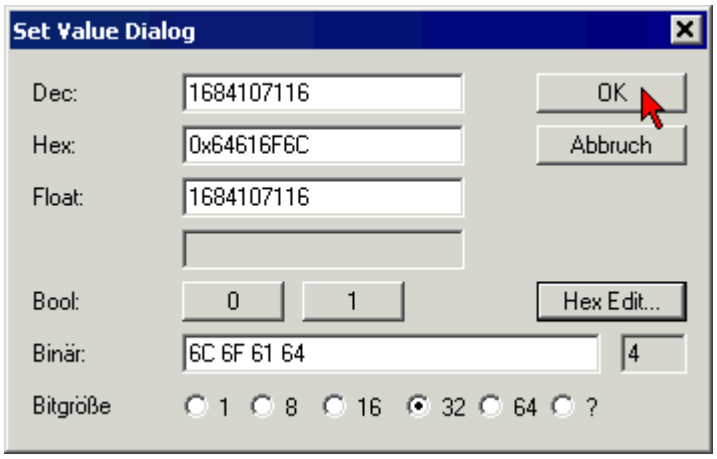


Fig. 154: Entering a restore value in the Set Value dialog

Alternative restore value

In some older terminals the backup objects can be switched with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164An incorrect entry for the restore value has no effect.

8.6 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

8.7 Reshipment and return

This product is individually packed and sealed. Unless otherwise agreed, Beckhoff can only accept returns in unopened original packaging with the seal intact.

List of illustrations

More Information:
www.beckhoff.com/en-us/products/i-o/ethercat-terminals/el2xxx-digital-output/

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