BECKHOFF New Automation Technology

Documentation | EN

EL126x

Digital Input Terminals with oversampling

2025-04-23 | Version: 2.9.0

1 Product overview - Digital input/output terminals with oversampling

EL1262 [▶ 16]
2-channel digital input terminal (24 V DC, 1 µs, with oversampling)

EL1262-0010 [▶ 22]
2-channel digital input/output terminal (5 V DC, 100 ns, 0.1 A, RS422/RS485, with oversampling)

EL1262-0050 [▶ 16]
2-channel digital input terminal (5 V DC, 1 µs, with oversampling)

EL1264 [▶ 16]
4-channel digital input terminal (24 V DC, 1 µs, with oversampling)

Table of contents

2 Foreword

2.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

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

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

Copyright

Third-party brands

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

2.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

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

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

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

Warning of damage to property or environment

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

Information on handling the product

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

2.3 Documentation Issue Status

Version	Comment
2.9.0	• Update chapter “Termination and topology” • Update structure
2.8.2	• Documentation for EL1264 supplemented • Additional documentation completed for EL1262-0010 • Chapter "Product overview" renamed to chapter "Product description" and subchapters "Introduction", "Technical data", "LEDs and connection" restructured • Subchapters "Using the SENT protocol" and "Access to TEDS" moved from the chapter "Commissioning" to the chapter "Sample programs" and summarized with the corresponding samples • Chapter "Mounting and wiring" supplemented with subchapters "Note on power supply" and "Disposal"; subchapter "Connection technology" updated, renamed subchapter "Connection" • All subchapters "Technical data" in the chapter "Product description" updated • "Guide through the documentation" supplemented in the foreword • Subchapter "Version identification of EtherCAT devices" (foreword) updated, addition of subchapters ◦ "General notes on marking" ◦ "Version identification of EL terminals" ◦ "Electronic access to the BIC (eBIC)" • Additions to the chapter "Commissioning" with subchapters: ◦ "Notes on EL1262-0010" ◦ "Sensitivity of the input" ◦ "Explanation of CycleCounter monitoring" ◦ "Explanation of the output behavior in case of error (Watchdog, CycleCounter)" • Addition to chapter "Firmware update EL1262-0010" (Appendix)
2.7	• Addition to chapter “Sample programs” • Structure update
2.6	• Addition to chapter "Sensitivity of the input" • Structure update
2.5	• Update chapter "UL notes" • Update chapter "Technical data" • Structure update
2.4	• Chapter Commissioning/Sample programs: "Sample 3: Reading and writing TEDS data" supplemented
2.3	• Chapter "Using the SENT protocol with EL1262-0050" incl. sample program (2) supplemented in the chapter "Commissioning"
2.2	• Update chapter "Notes on the documentation" • Update Technical data • Chapter "Installation instructions for enhanced mechanical load capacity" added • Update chapter "TwinCAT 2.1x" -> chapter "TwinCAT development environment" and chapter "TwinCAT Quick Start"
2.1	• Chapter "Oversampling terminals and TwinCAT Scope" supplemented
2.0	• Migration
1.5	• Structure update • Chapter "Technical data" updated • Chapter "LEDs and connection" updated
1.4	• Structure update • EL1262-0050 supplemented
1.3	• Technical data supplemented
1.2	• Notes on device description update supplemented; note on trademarks inserted
1.1	• Sample program supplemented
1.0	• Technical description added, first publication
0.1	• Provisional documentation for EL1262

2.4 Guide through documentation

NOTICE

Further components of documentation

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

Title	Description
EtherCAT System Documentation (PDF)	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
Explosion Protection for Terminal Systems (PDF)	Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx
Control Drawing I/O, CX, CPX (PDF)	Connection diagrams and Ex markings (conform to cFMus)
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

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

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

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

2.5 Version identification of EtherCAT devices

2.5.1 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. *"EL2872 with revision 0022 and serial number 01200815"*.
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

2.5.2 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

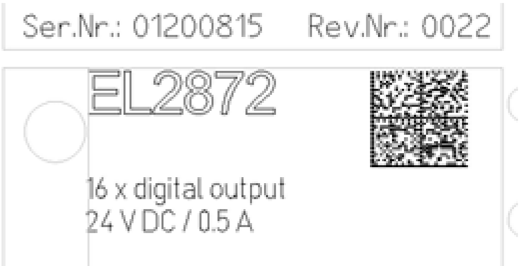


Fig. 1: EL2872 with revision 0022 and serial number 01200815

2.5.3 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.

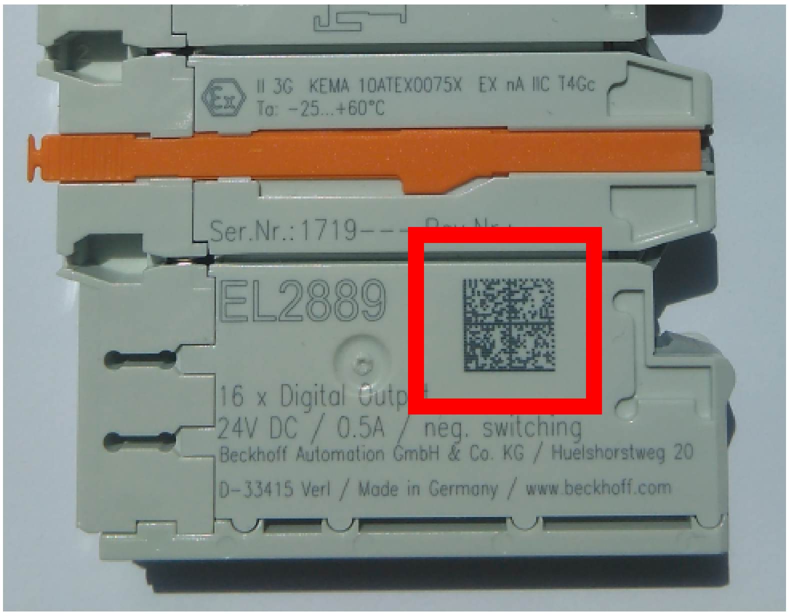


Fig. 2: BIC as data matrix code (DMC, code scheme ECC200)

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:

Posi-tion	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	**1P**072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	SBTN	12	**SBTN**k4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	**1K**EL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	**Q**1
5	Batch number	Optional: Year and week of production	2P	14	**2P**401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	**51S**678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	12	**30P**F971, 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**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

Fig. 3: Example DMC **1P**072222**SBTN**k4p562d7**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.

NOTICE

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 documentation.

2.5.4 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.

The interface that the product can be electronically addressed by is crucial for the electronic readout.

K-bus devices (IP20, IP67)

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

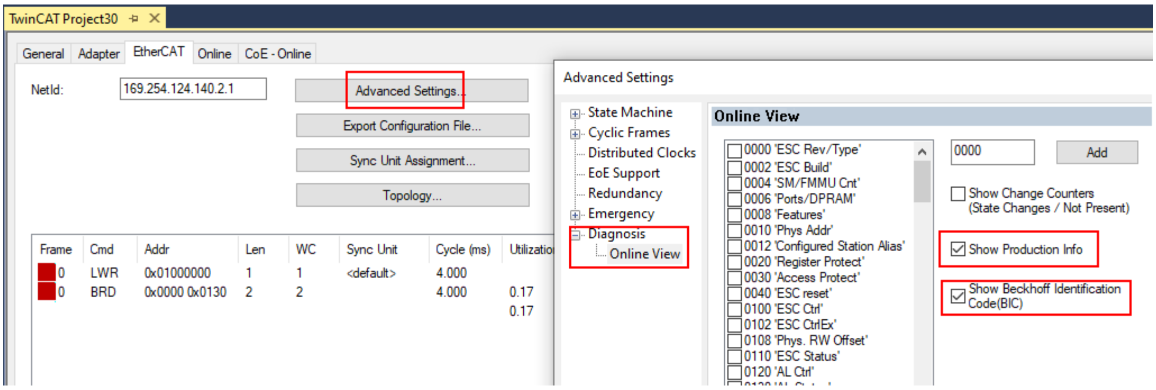
EtherCAT devices (IP20, IP67)

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

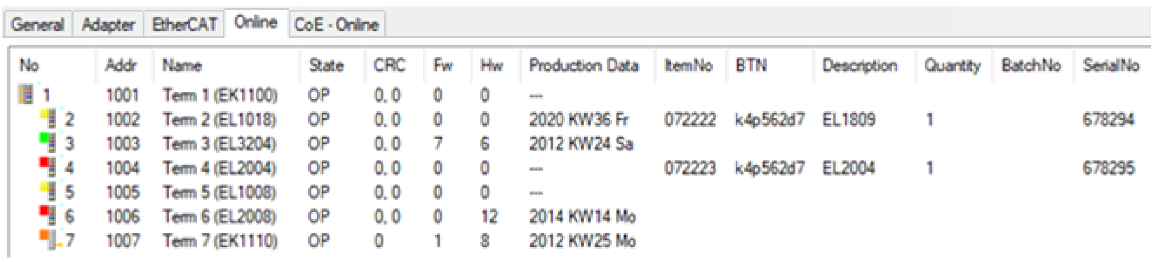
Beckhoff also stores the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

The user can electronically access the eBIC (if present) 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 "Show Beckhoff Identification Code (BIC)" checkbox under EtherCAT → Advanced Settings → Diagnostics:



    - The BTN and its contents are then displayed:



    - Note: As shown in the figure, the production data HW version, FW version, and production date, which have been programmed since 2012, can also be displayed with "Show production info".
    - Access from the PLC: From TwinCAT 3.1. build 4024.24, the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0.
- EtherCAT devices with a CoE directory may also have the object 0x10E2:01 to display their own eBIC, which can also be easily accessed by the PLC:

- 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 preferentially introduced into stock products in the course of necessary firmware revision.
  - From TwinCAT 3.1. build 4024.24, the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0
- The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2_Utilities* as of TwinCAT 3.1 build 4024.24
  - F_SplitBIC: The function splits the Beckhoff Identification Code (BIC) sBICValue into its components using known identifiers and returns the recognized substrings in the ST_SplittedBIC structure as a return value
  - BIC_TO_BTN: The function extracts the BTN from the BIC and returns it as a return value
- Note: If there is further electronic 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 written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50..200 bytes of memory.
- 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 which each have 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, and DeviceNet devices

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

3 Product description

3.1 EL1262, EL1262-0050, EL1264

3.1.1 Introduction

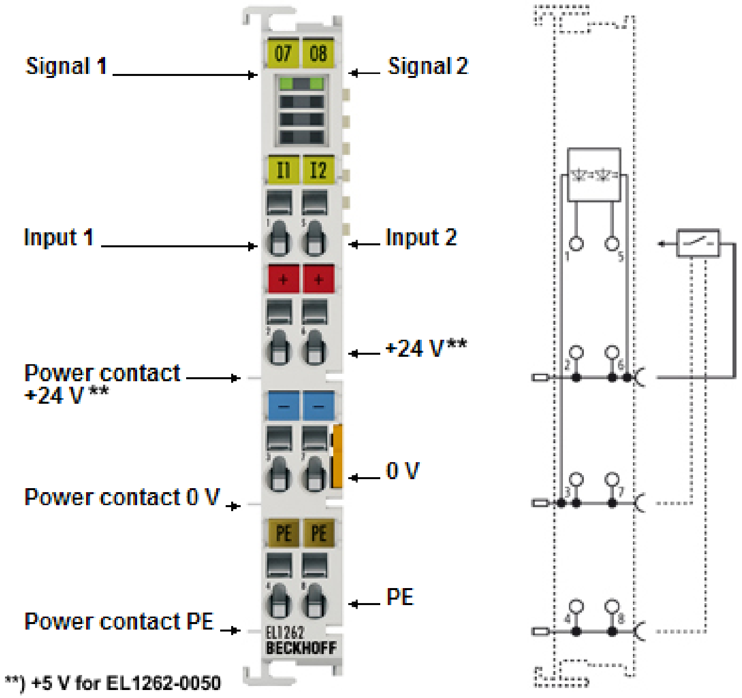


Fig. 4: EL1262-00x0

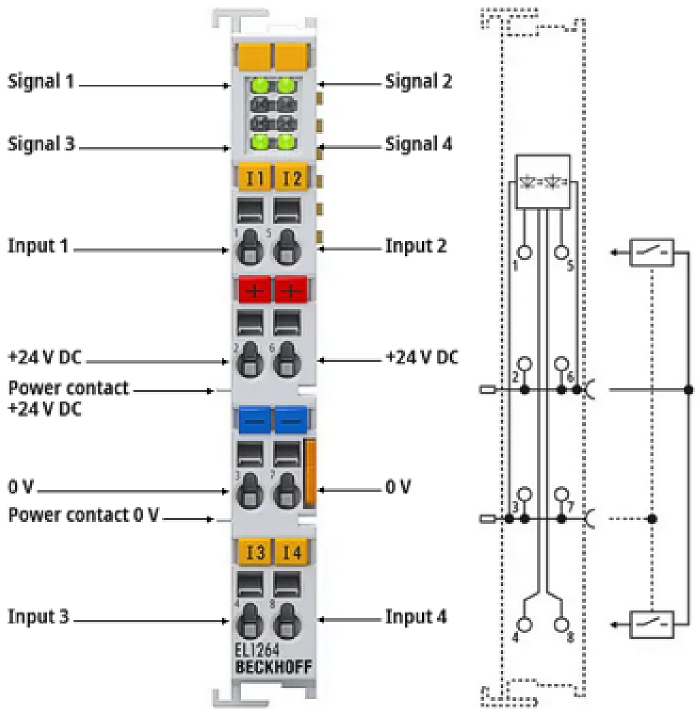


Fig. 5: EL1264

Digital input terminals (5/ 24 V DC, 1 µs, with oversampling)

The EL1262 digital input terminal acquires the fast binary control signals from the process level and transmits them, in an electrically isolated form, to the controller. The signals are sampled with a configurable, integer multiple (oversampling factor: n) of the bus cycle time (n microcycles per bus cycle). For each microcycle, the EtherCAT Terminal generates a process data block that is transferred collectively during the next bus cycle. The time base of the terminal can be synchronized precisely with other EtherCAT devices via distributed clocks. This procedure enables the temporal resolution of the acquisition of the digital input signals to be increased to n times the bus cycle time.

The EL1262-0050 offers a version with 5 V input voltage (TTL level) and 5 V supply voltage.

The EL1264 is the 4-channel version of the terminal.

Quick links

- EtherCAT basics
- LEDs and connection [▸ 19]
- Commissioning [▸ 60]
- Basic function principles [▸ 155]

3.1.2 Technical data

Technical data	EL1262	EL1264	EL1262-0050
Digital inputs	2	4	2
Nominal voltage of the inputs	24 V_{DC} (-15 %/+20 %)		5 V_{DC} (-15 %/+20 %)
Signal voltage "0"	-3 V … +5 V (based on EN 61131-2, type 3)		< 0.8 V
Signal voltage "1"	+11 V … +30 V (based on EN 61131-2, type 3)		>2.4 V
Input current	typ. 3 mA (based on EN 61131-2, type 3)		typ. 50 µA
Input filter delay	< 1 µs typ.		
Oversampling factor	n = integer multiple of the cycle time, 1...1000		
Accuracy Distributed Clocks (DC)	<< 1 µs		
Sampling rate	max. 1 Msamples/s		
Power supply for the electronics	via the power contacts (24 V)		via the power contacts (5 V) (observe the instructions!)
Current consumption via E-bus	typ. 70 mA	typ. 80 mA	typ. 70 mA
Electrical isolation	500 V (E-bus/field potential)		
Bit width in process image	n x 2 Inputs + 64-Bit-CycleCounter/Latch	4 x (n Inputs + 2 Byte CycleCounter)	n x 2 Inputs + 64-Bit-CycleCounter/Latch
Configuration	via TwinCAT System Manager		
Weight	approx. 55 g		
Permissible ambient temperature range during operation	0 °C ... + 55 °C		
Permissible ambient temperature range during storage	-25 °C ... + 85 °C		
Permissible relative air humidity	95 %, no condensation		
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)		
Installation [▸ 47]	on 35 mm mounting rail, conforms to EN 60715		
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27 see also Installation instructions for enhanced mechanical load capacity [▸ 50]		
EMC immunity / emission	in accordance with EN 61000-6-2 / EN 61000-6-4		
Protection rating	IP20		
Installation position	variable		
Approvals *)	CE, EAC, UKCA, CCC cFMus [▸ 44], ATEX [▸ 41], IECEx [▸ 42] cULus [▸ 46]	CE, EAC, UKCA	CE, EAC, UKCA, CCC ATEX [▸ 41], IECEx [▸ 42] cULus [▸ 46]

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

Ex markings

Standard	Marking
ATEX	II 3 G Ex nA IIC T4 Gc
IECEx	Ex nA IIC T4 Gc
cFMus	Class I, Division 2, Groups A, B, C, D Class I, Zone 2, AEx/Ex ec IIC T4 Gc

Electrical supply

The fast input circuits of the EL1262/EL1264 are supplied via the power contacts. Please note that the EL1262-0050 can only be operated with a 5 V supply voltage! If necessary, use a EL9505 power supply terminal for supplying the EL1262-0050.

3.1.3 LEDs and connection

EL1262

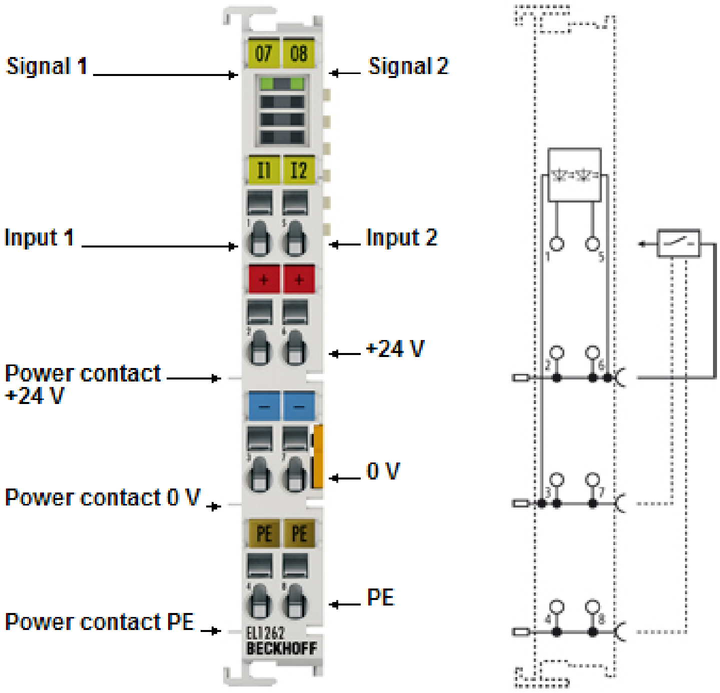


Fig. 6: EL1262

EL1262 LEDs

LED	Color	Meaning	
INPUT 1 - 2	green	off	There is no input signal at the respective input
		on	+24 V input signal at the respective input

EL1262 connection

Terminal point		Description
Name	No.	
Input 1	1	Input 1
+ 24 V	2	+24 V (internally connected to terminal point 6 and positive power contact)
0 V	3	0 V (internally connected to terminal point 7 and negative power contact)
PE	4	PE contact
Input 2	5	Input 2
+ 24 V	6	24 V (internally connected to terminal point 2 and positive power contact)
0 V	7	0 V (internally connected to terminal point 3 and negative power contact)
PE	8	PE contact

EL1262-0050

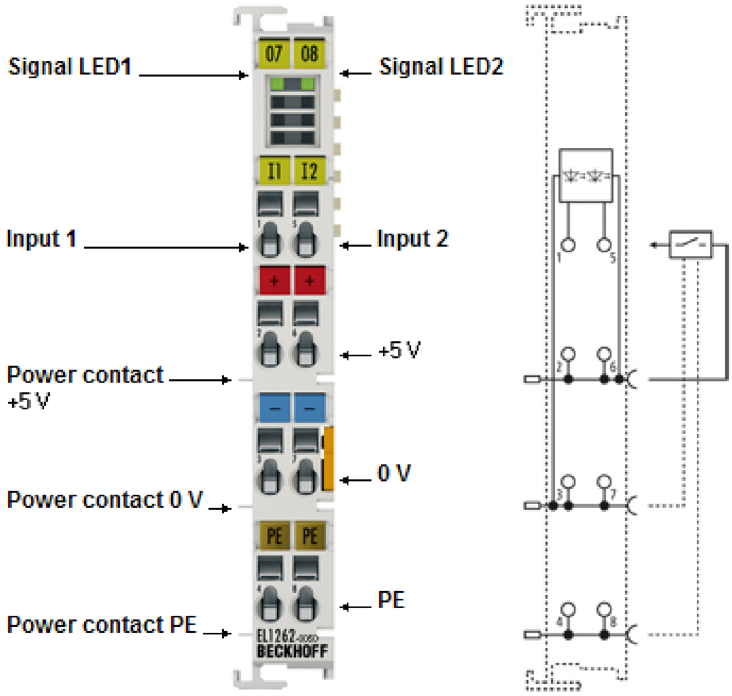


Fig. 7: EL1262-0050

NOTICE

Be sure the correct supply voltage is used!

The EL1262-0050 can only be operated with a 5 V supply voltage! The terminal will not work in a terminal network with a 24 V supply voltage at the power contacts! If necessary, use a EL9505 power supply terminal for supplying the EL1262-0050.

EL1262-0050 LEDs

LED	Color	Meaning	
INPUT 1 - 2	green	off	There is no input signal at the respective input
		on	+5 V input signal at the respective input

EL1262-0050 connection

Terminal point		Description
Name	No.	
Input 1	1	Input 1
+ 5 V	2	+5 V (internally connected to terminal point 6 and positive power contact)
0 V	3	0 V (internally connected to terminal point 7 and negative power contact)
PE	4	PE contact
Input 2	5	Input 2
+ 5 V	6	+ 5 V (internally connected to terminal point 2 and positive power contact)
0 V	7	0 V (internally connected to terminal point 3 and negative power contact)
PE	8	PE contact

EL1264

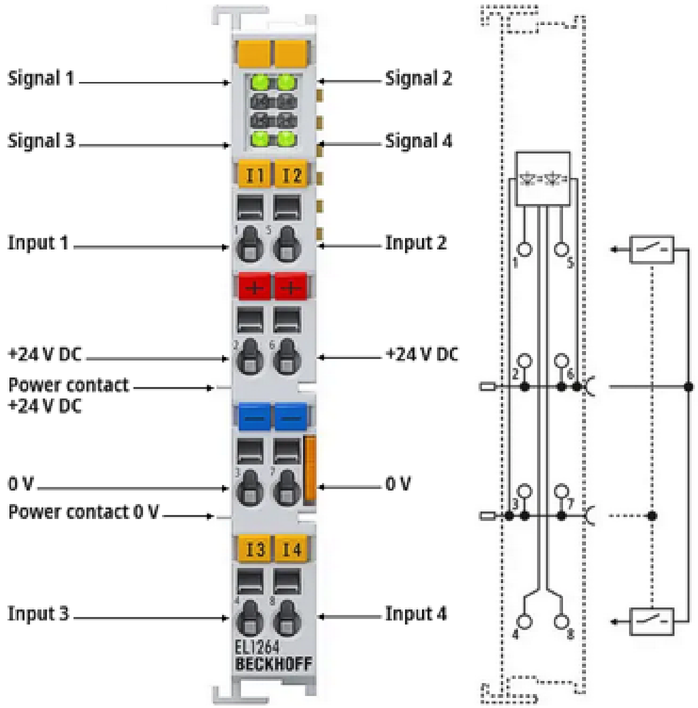


Fig. 8: EL1264

EL1264 LEDs

LED	Color	Meaning	
INPUT 1-4	green	off	There is no input signal at the respective input
		on	+24 V input signal at the respective input

EL1264 connection

Terminal point		Description
Name	No.	
Input 1	1	Input 1
+ 24 V	2	+24 V (internally connected to terminal point 6 and positive power contact)
0 V	3	0 V (internally connected to terminal point 7 and negative power contact)
Input 3	4	Input 3
Input 2	5	Input 2
+ 24 V	6	24 V (internally connected to terminal point 2 and positive power contact)
0 V	7	0 V (internally connected to terminal point 3 and negative power contact)
Input 4	8	Input 4

3.2 EL1262-0010

3.2.1 Introduction

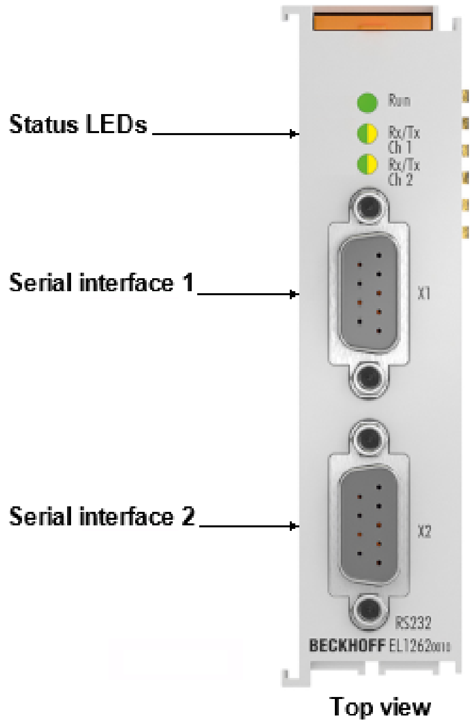


Fig. 9: EL1262-0010

2-channel digital input/output terminal (5 V DC, 100 ns, 0.1 A, RS422/RS485, with oversampling)

The EL1262-0010 is a universal, high performance oversampling terminal for digital input and output signals. As an EtherCAT oversampling terminal, it increases with this XFC process the temporal resolution of the digital signals many times compared to the EtherCAT bus cycle. The time base of the terminal can be synchronized precisely with other EtherCAT devices via distributed clocks. For the input data it generates a set of process data in each EtherCAT bus cycle, which is collected and transmitted in the next bus cycle. Conversely, it expects a set of output data for the outputs in each EtherCAT bus cycle. With the EtherCAT oversampling principle the output/read time can not be set freely, it results from the continuous rhythm. This distinguishes "oversampling" from the "EtherCAT timestamp principle" where the output/read time can be freely selected (compare EL1252, EL2252).

The terminal contains 2 input and 2 output channels. All 4 channels can be used simultaneously at up to 10 Mbit/s. This allows it to read and output binary states with up to 100 ns fineness. It thus extends the application range of the previous EL1262/EL2262 with its 1 Mbit/s by a factor of 10.

Electrically, it offers both differential RS422 (full duplex)/RS485 (half duplex) and 5 V single-ended in both directions. Termination and bias is electronically switchable for serial operation.

This makes it suitable for the most demanding high-speed applications such as

- laser control
- reading or generating serial protocols (emulating protocols)
- PWM control of converters

Quick links

- EtherCAT basics
- LEDs and connection [▶ 24]
- Commissioning [▶ 60]
- Basic function principles [▶ 155]

3.2.2 Technical data

Technical data		EL1262-0010
Connection technology		2x D-sub, 9-pin, female
Number of inputs		2
Sampling rate		max. 10 Msamples/s, min. clock rate 0.1 µs
Number of outputs		2
Output rate		max. 10 Msamples/s, min. clock rate 0.1 µs
Oversampling/multi-timestamping factor		n = integer multiple of the cycle time, 8...10,000, see documentation
Precision of timestamp in the terminal		10 ns
Distributed clocks		yes
Distributed clock precision		<< 1 µs
RS operation	Specification	RS422/RS485
5 V operation	Specification	5 V TTL
	Max. output current	0.1 A on each channel, individually short-circuit proof
	Output stage	push-pull
	Load type	ohmic
Electrical isolation		500 V (E-bus/field potential)
Current consumption via E-bus		typ. 540 mA
Current consumption power contacts		-
Power supply for the electronics		via E-bus
Configuration		via TwinCAT System Manager
Special features		Oversampling, termination/bias switchable in RS mode, 5 V supply permanently available
Weight		approx. 55 g
Permissible ambient temperature range during operation		0 °C... + 55 °C
Permissible ambient temperature range during storage		-25 °C... + 85 °C
Permissible relative air humidity		95%, no condensation
Dimensions (W x H x D)		approx. 24 mm x 100 mm x 52 mm
Installation [▶ 47]		on 35 mm DIN rail, conforming to EN 60715
Vibration / shock resistance		conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity / emission		conforms to EN 61000-6-2 / EN 61000-6-4
Protection rating		IP20
Installation position		variable
Approvals *)		CE, EAC

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

3.2.3 LEDs and connection

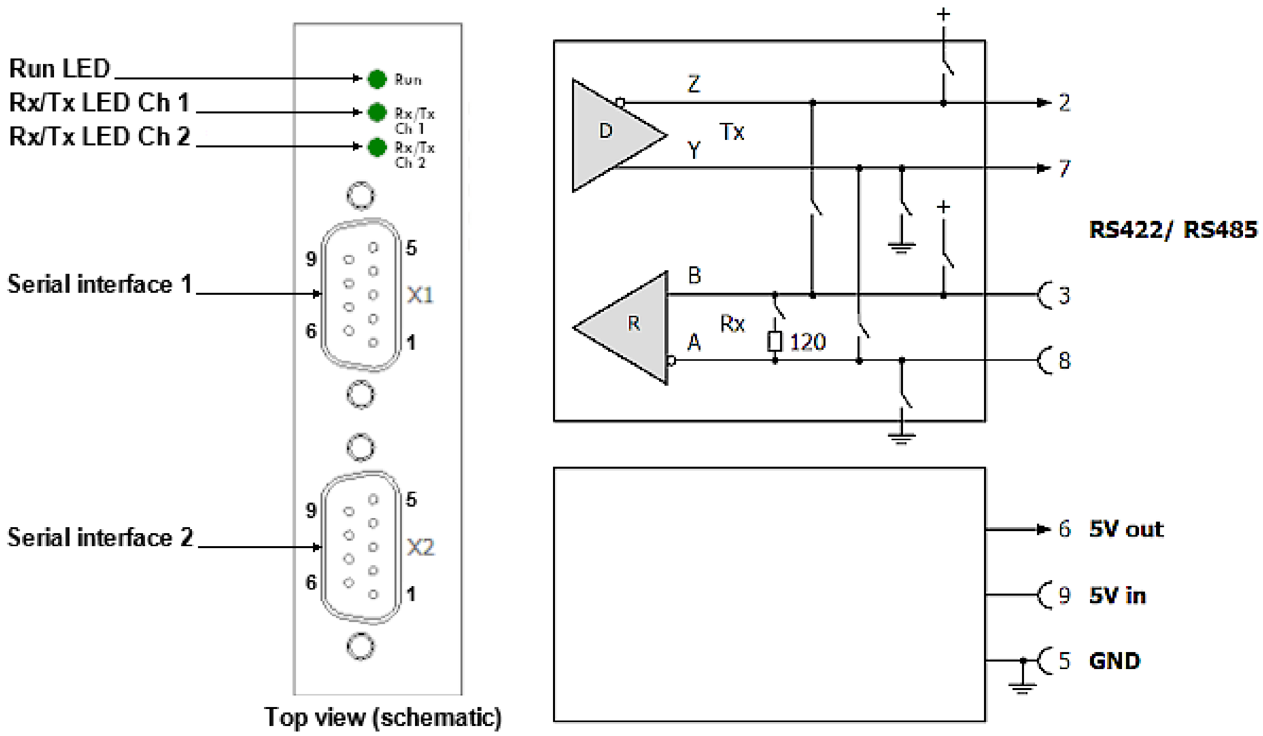


Fig. 10: EL1262-0010 LEDs and connectors

EL1262-0010 LEDs

LED	Color	Meaning	
RUN	green	off	State of the EtherCAT State Machine [▶ 32]: **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 [▶ 119] and the Distributed Clocks [▶ 39] (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 [▶ 196] of the terminal

LED	Color	Meaning	
Signal Rx/ Tx Ch(n)	Green	Off	No signal inputs activity
		On	Signal inputs activity
	Yellow	Off	No signal output activity
		On	Signal outputs activity

EL1262-0010 Sub-D sockets connection

Sub-D 9-pole, X1 (Channel 1) X2 (Channel 2)	Description
Pin No.	
1	n.c
2	Tx+
3	Rx+
4	n.c.
5	GND
6	Output +5 V DC
7	Tx-
8	Rx-
9	Input +5 V DC

Use of the RS interface

Level interfaces

The devices operate with a differential RS485/422 level.

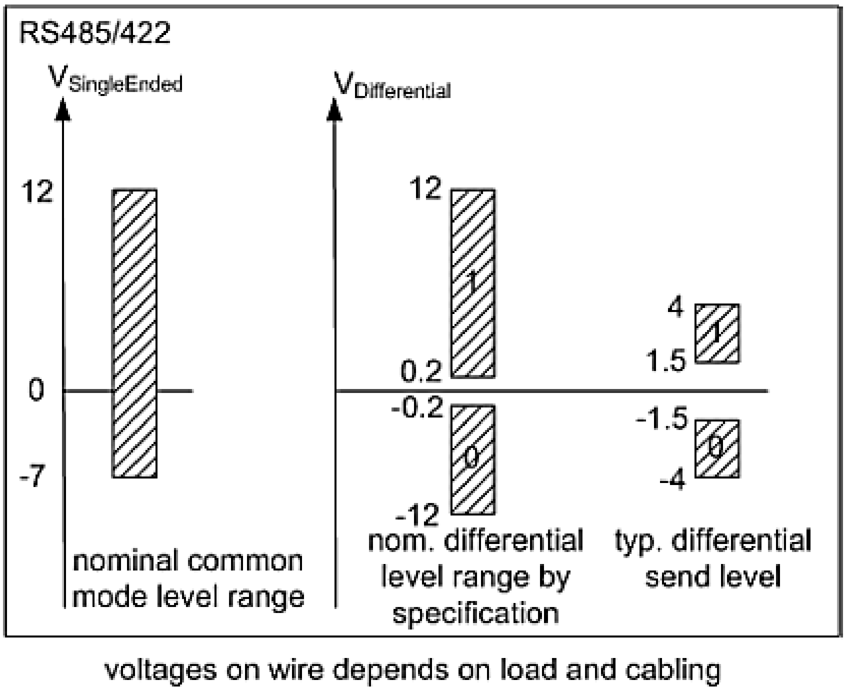


Fig. 11: Level interfaces RS485/422

Termination and topology

The serial RS422 and RS485 communication technologies operate with voltage levels on a 2-wire line. Reflections at high-resistance line ends can lead to signal distortion. For this reason termination resistors are required at the receiver. For RS422/485 these are 120 Ω resistors, which together with the line resistance result in a voltage drop over the transmission link.

Permitted cable length

The line resistance together with the termination resistor results in an overall voltage drop over the transmission link. An unacceptably high number of termination resistors would result in excessive attenuation of the signal.

The system design should ensure that the voltage does not drop below 200 mV at the receiver (see Fig.), which is the minimum voltage required.

In RS422 mode each line must be terminated with 120 Ω at the receiver.

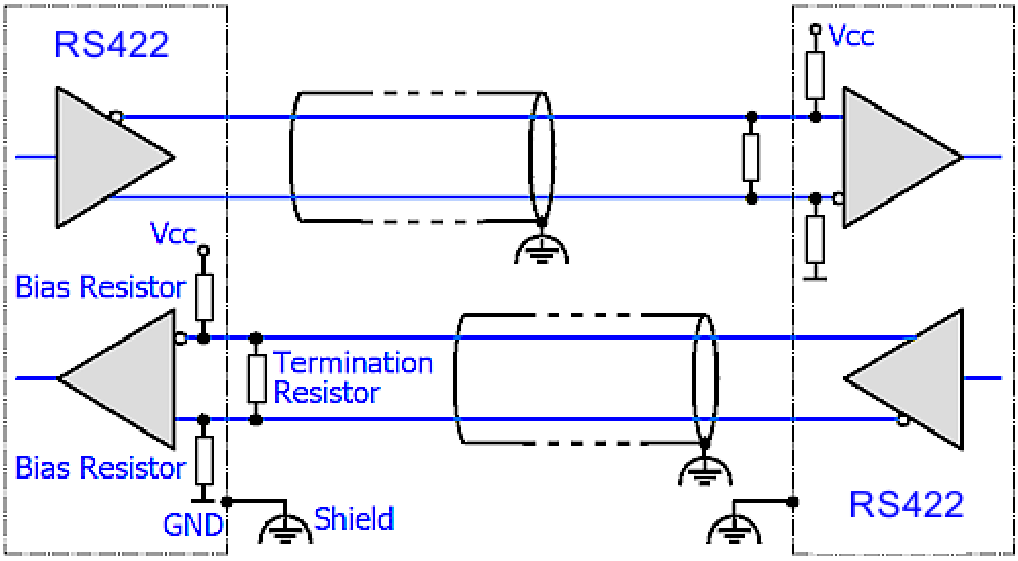


Fig. 12: RS422 termination

In RS485 mode with several devices, termination resistors are only used at the two end devices.

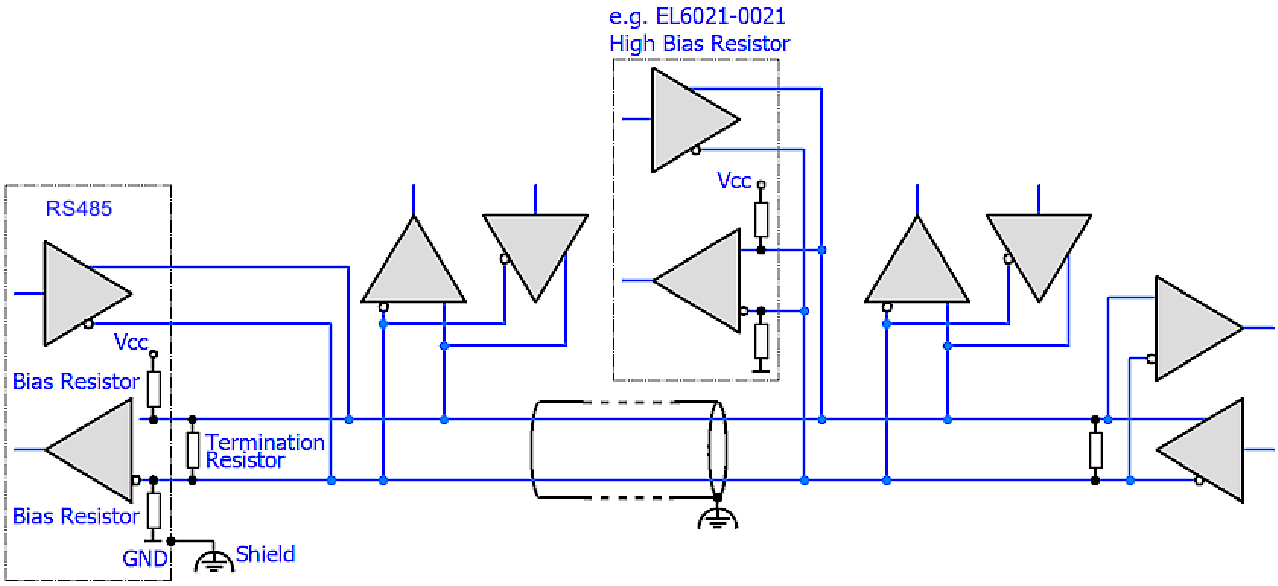


Fig. 13: RS485 termination

The background is the different design of RS422/EIA-422 and RS485/EIA-485:

- RS422: 1 Rx → Tx n (maximum 10 receivers)
- RS485: n Rx → Tx m (maximum 32/128 devices, depending on the resulting bus loading)

Components for RS485 usually have a higher input impedance, resulting in lower bus load.

Topology

The termination and the bias resistors generate a load on the bus. However, they are essential for unambiguous bus levels and therefore have to be positioned with diligence. Ideally the RS422/485 bus should be configured as a daisy chain or a simple chain, see Fig. The following topologies may be problematic:

- Star topologies: each end point should ideally be terminated, but this can lead to excessive bus loading and ambiguous signal levels. Other potential issues are reflections and runtime variations.
- Intermeshed topologies: no clear end points, which means reflections and circulating currents are possible.

Shielding/shield

NOTICE
Do not use functional earth for discharge of residual currents or potential differences! The terminals offer a shielded connection for discharging EMC interference via the cable shield (FE, functional earth). The shield must not be misused for discharging residual currents or potential differences.

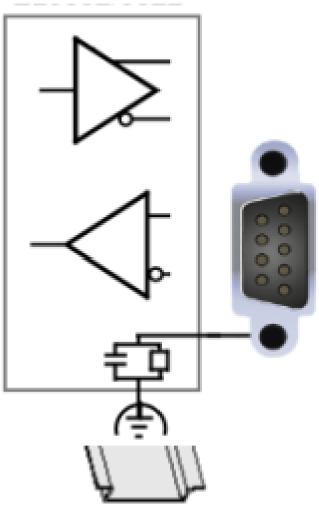


Fig. 14: ShieldConnection EL1262-0010.png

The EL1262-0010 uses the RS interface "only" for electrical reasons, in order to be able to transmit the short bit times differentially. It is not an independent user for telegram-based serial communication! If the terminal is to act as a serial device, the telegram structure (7E2, 8N1...), handshake, etc. must be carried out completely on the PLC side and the corresponding bits must then be transferred to/read from the terminal via the process data.

Termination is recommended!

3.3 Start

For commissioning:

- mount the EL126x/ EL2262 as described in the section Mounting and wiring [▶ 40].
- configure the EL126x/ EL2262 in TwinCAT as described in the section Commissioning [▶ 60].

4 Basics communication

4.1 EtherCAT basics

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

4.2 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.

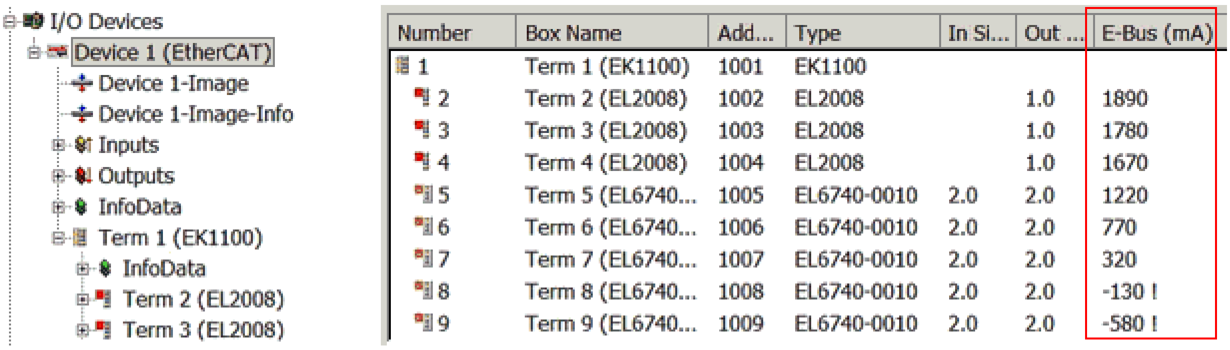


Fig. 15: System manager current calculation

NOTICE

Malfunction possible!

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

4.3 General notes for setting the watchdog

The EtherCAT 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 features two watchdogs:

- Sync Manager (SM) watchdog (default: 100 ms)
- Process Data (PDI) watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:

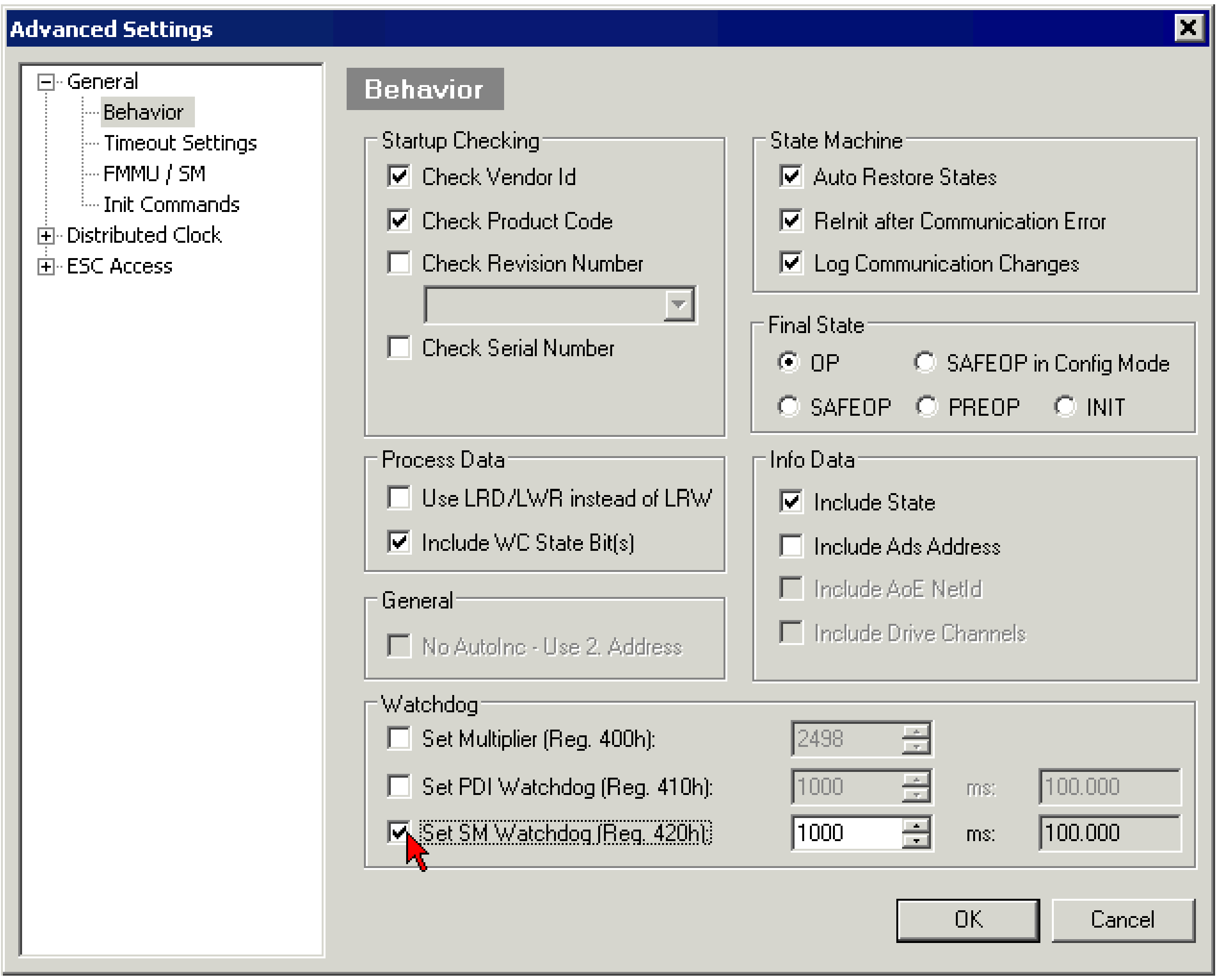


Fig. 16: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. 0x0400) 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 400h, 410h and 420h: 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 register 400h/420h but executed by the microcontroller (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 ESC for longer than the set and activated Process Data Interface (PDI) watchdog time, this watchdog is triggered.

The PDI 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)] * SM/PDI watchdog

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

The value in "Watchdog multiplier + 2" in the formula above corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

The function for switching off the SM watchdog via SM watchdog = 0 is only implemented in terminals from revision -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.

4.4 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
- Operational

- Bootstrap

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

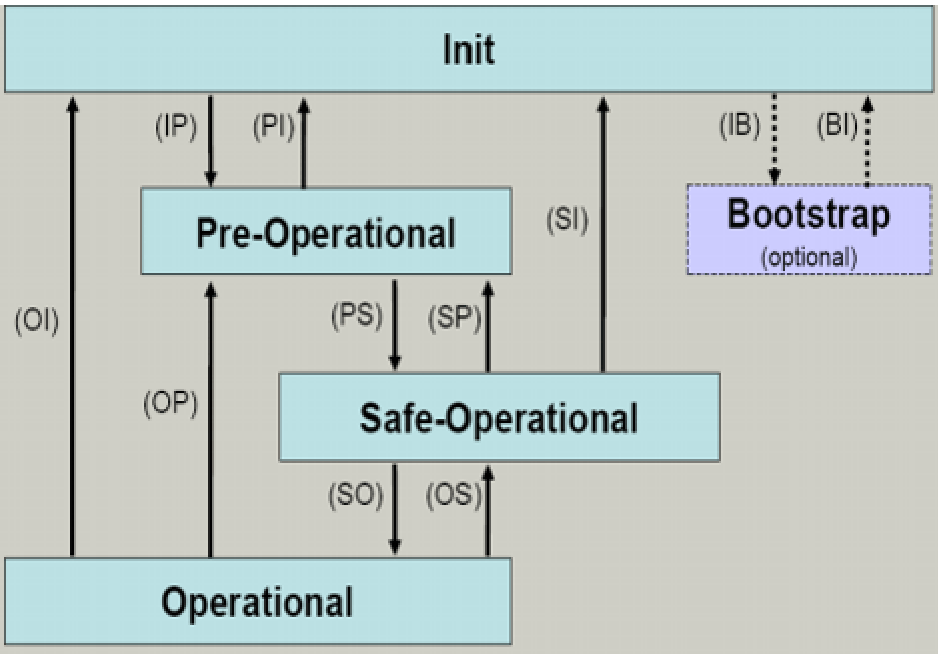


Fig. 17: States of the EtherCAT State Machine

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 Fieldbus Memory Management Unit (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 Dual Port (DP)-RAM areas of the ESC.

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 monitoring sets the outputs of the ESC 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 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.

4.5 CoE Interface

General description

The CoE interface (CAN application protocol over EtherCAT interface) 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 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 data 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 (“inputs” from the perspective of the EtherCAT master)
- 0x7000: Output PDOs (“outputs” 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:

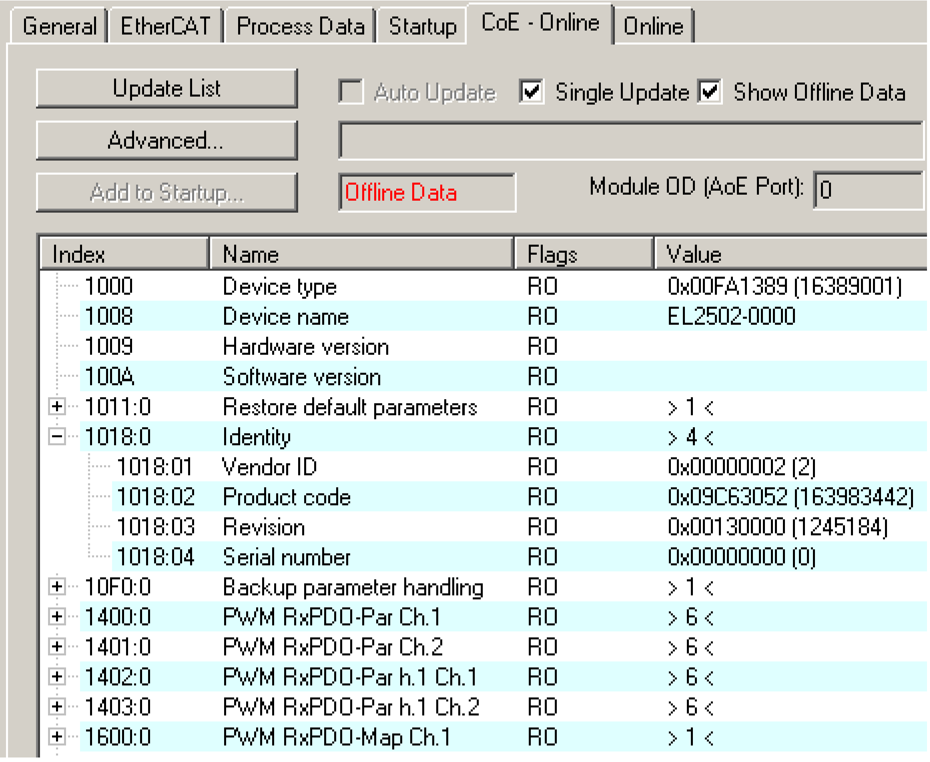


Fig. 18: “CoE Online” tab

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

NOTICE

Changes in the CoE directory (CAN over EtherCAT directory), 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 TwinCAT 3 | 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,

- via the System Manager (Fig. “CoE Online” tab) by clicking.
  This is useful for commissioning of the system or slaves. Click on the row of the index to be parameterized and enter a value in the “SetValue” dialog.
- from the control system or 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 index 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.
- If the 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.

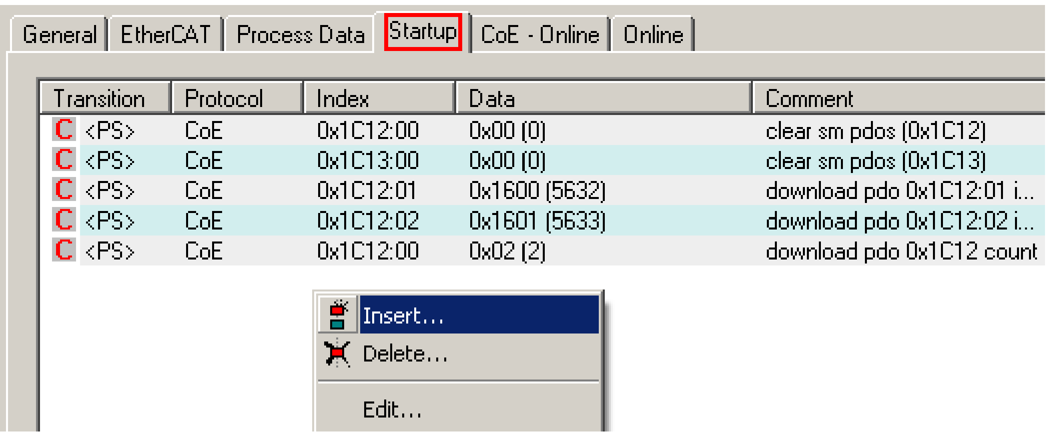


Fig. 19: Startup list in the TwinCAT System Manager

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

Online / offline list

When working with the TwinCAT System Manager, a distinction must be made as to whether the EtherCAT device is currently "available", i.e. switched on and connected via EtherCAT - i.e. **online** - or whether a configuration is created **offline** without slaves being connected.

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 Data** is shown in red.

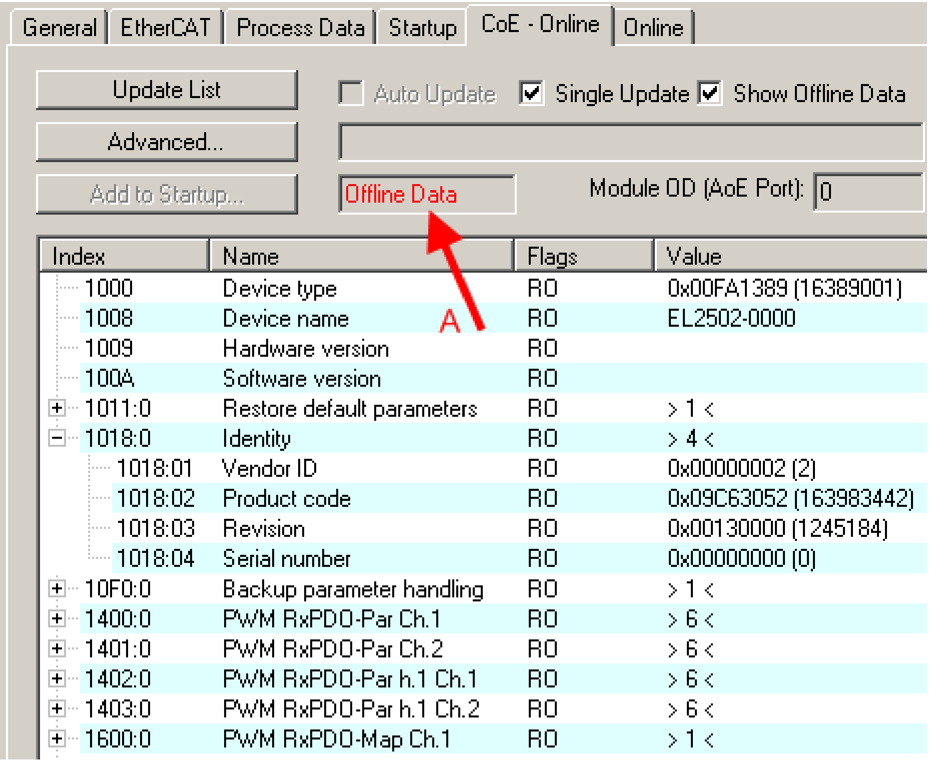


Fig. 20: Offline list

- 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 status of the device is displayed in the CoE.
  - **Online Data** is shown in green.

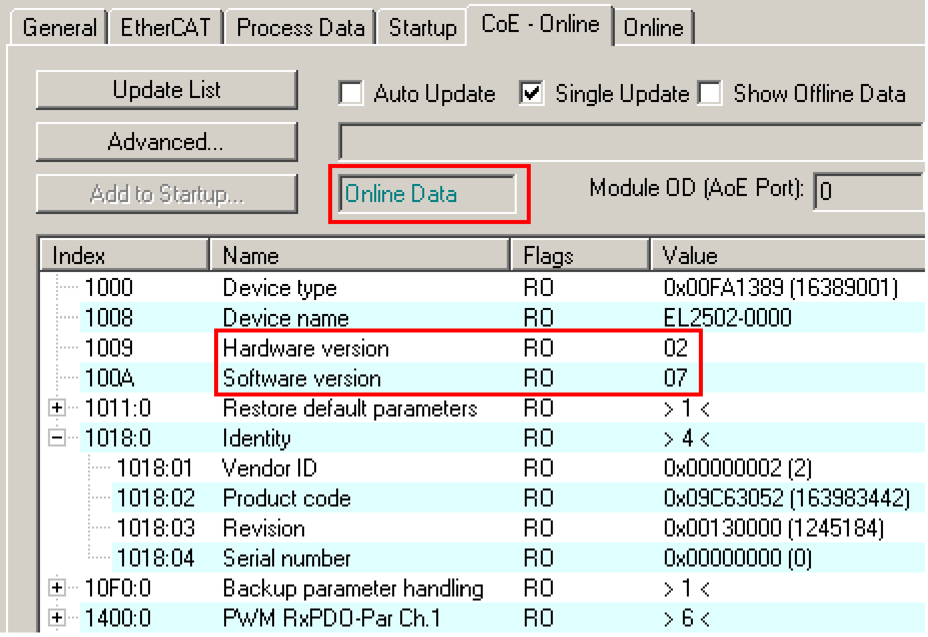


Fig. 21: Online list

Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels, for example, a 4-channel analog 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} or 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.

4.6 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.

5 Mounting and wiring

5.1 Instructions for ESD protection

NOTICE

Destruction of the devices by electrostatic discharge possible!

The devices contain components at risk from electrostatic discharge caused by improper handling.

- When handling the components, ensure that there is no electrostatic discharge; also avoid touching the spring contacts directly (see illustration).
- Contact with highly insulating materials (synthetic fibers, plastic films, etc.) should be avoided when handling components at the same time.
- When handling the components, ensure that the environment (workplace, packaging and persons) is properly earthed.
- Each bus station must be terminated on the right-hand side with the EL9011 or EL9012 end cap to ensure the degree of protection and ESD protection.

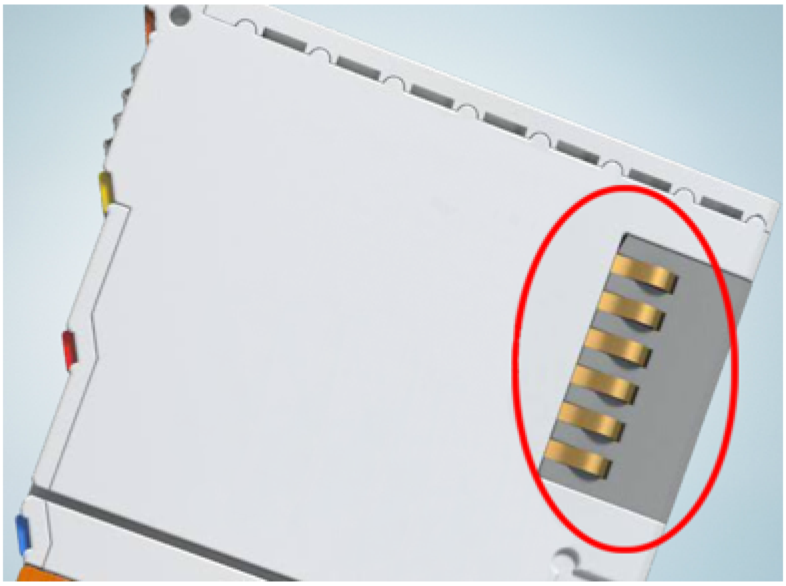


Fig. 22: Spring contacts of the Beckhoff I/O components

5.2 Explosion protection

5.2.1 ATEX - Special conditions (standard temperature range)

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components with standard temperature range in potentially explosive areas (directive 2014/34/EU)!

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60079-15! The environmental conditions during use are thereby to be taken into account!
- For dust (only the fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9): The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-31 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of 0 to 55°C for the use of Beckhoff fieldbus components standard temperature range in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

The fundamental health and safety requirements are fulfilled by compliance with the following standards:

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010
- EN 60079-31:2013 (only for certificate no. KEMA 10ATEX0075 X Issue 9)

Marking

The Beckhoff fieldbus components with standard temperature range certified according to the ATEX directive for potentially explosive areas bear one of the following markings:

II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: 0 … +55°C

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

or

II 3G KEMA 10ATEX0075 X Ex nA nC IIC T4 Gc Ta: 0 … +55°C

II 3D KEMA 10ATEX0075 X Ex tc IIIC T135°C Dc Ta: 0 ... +55°C
(only for fieldbus components of certificate no. KEMA 10ATEX0075 X Issue 9)

5.2.2 IECEx - Special conditions

⚠ **WARNING**

Observe the special conditions for the intended use of Beckhoff fieldbus components in potentially explosive areas!

- For gas: The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to IEC 60079-15, taking into account the environmental conditions under which the equipment is used!
- For dust (only the fieldbus components of certificate no. IECEx DEK 16.0078X Issue 3):
  The equipment shall be installed in a suitable enclosure providing a degree of protection of IP54 according to EN 60079-31 for group IIIA or IIIB and IP6X for group IIIC, taking into account the environmental conditions under which the equipment is used!
- The equipment shall only be used in an area of at least pollution degree 2, as defined in IEC 60664-1!
- Provisions shall be made to prevent the rated voltage from being exceeded by transient disturbances of more than 119 V!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range for the use of Beckhoff fieldbus components in potentially explosive areas!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The front hatch of certified units may only be opened if the supply voltage has been switched off or a non-explosive atmosphere is ensured!

Standards

The fundamental health and safety requirements are fulfilled by compliance with the following standards:

- EN 60079-0:2011
- EN 60079-15:2010
- EN 60079-31:2013 (only for certificate no. IECEx DEK 16.0078X Issue 3)

Marking

Beckhoff fieldbus components that are certified in accordance with IECEx for use in areas subject to an explosion hazard bear the following markings:

Marking for fieldbus components of certificate no. IECEx DEK 16.0078X Issue 3:	**IECEx DEK 16.0078 X** **Ex nA IIC T4 Gc** **Ex tc IIIC T135°C Dc**
Marking for fieldbus components of certficates with later issues:	**IECEx DEK 16.0078 X** **Ex nA IIC T4 Gc**

5.2.3 Continuative documentation for ATEX and IECEx

NOTICE	
	Continuative documentation about explosion protection according to ATEX and IECEx Pay also attention to the continuative documentation **Ex. Protection for Terminal Systems** Notes on the use of the Beckhoff terminal systems in hazardous areas according to ATEX and IECEx, that is available for download within the download area of your product on the Beckhoff homepage www.beckhoff.com!

5.2.4 cFMus - Special conditions

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components in potentially explosive areas!

- The equipment shall be installed within an enclosure that provides a minimum ingress protection of IP54 in accordance with ANSI/UL 60079-0 (US) or CSA C22.2 No. 60079-0 (Canada).
- The equipment shall only be used in an area of at least pollution degree 2, as defined in IEC 60664-1.
- Transient protection shall be provided that is set at a level not exceeding 140% of the peak rated voltage value at the supply terminals to the equipment.
- The circuits shall be limited to overvoltage Category II as defined in IEC 60664-1.
- The Fieldbus Components may only be removed or inserted when the system supply and the field supply are switched off, or when the location is known to be non-hazardous.
- The Fieldbus Components may only be disconnected or connected when the system supply is switched off, or when the location is known to be non-hazardous.

Standards

The fundamental health and safety requirements are fulfilled by compliance with the following standards:

M20US0111X (US):

- FM Class 3600:2018
- FM Class 3611:2018
- FM Class 3810:2018
- ANSI/UL 121201:2019
- ANSI/ISA 61010-1:2012
- ANSI/UL 60079-0:2020
- ANSI/UL 60079-7:2017

FM20CA0053X (Canada):

- CAN/CSA C22.2 No. 213-17:2017
- CSA C22.2 No. 60079-0:2019
- CAN/CSA C22.2 No. 60079-7:2016
- CAN/CSA C22.2 No.61010-1:2012

Marking

Beckhoff fieldbus components that are certified in accordance with cFMus for use in areas subject to an explosion hazard bear the following markings:

FM20US0111X (US):	**Class I, Division 2, Groups A, B, C, D** **Class I, Zone 2, AEx ec IIC T4 Gc**
FM20CA0053X (Canada):	**Class I, Division 2, Groups A, B, C, D** **Ex ec T4 Gc**

5.2.5 Continuative documentation for cFMus

NOTICE

Continuative documentation about explosion protection according to cFMus

Pay also attention to the continuative documentation

Control Drawing I/O, CX, CPX
Connection diagrams and Ex markings,

that is available for download within the download area of your product on the Beckhoff homepage www.beckhoff.com!

5.3 UL notice

⚠ CAUTION

Application

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

⚠ CAUTION

Examination

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

⚠ CAUTION

For devices with Ethernet connectors

Not for connection to telecommunication circuits.

Basic principles

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

5.4 Installation on mounting rails

⚠ WARNING

Risk of electric shock and damage of device!

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

The Bus Terminal system and is designed for mounting in a control cabinet or terminal box.

Assembly

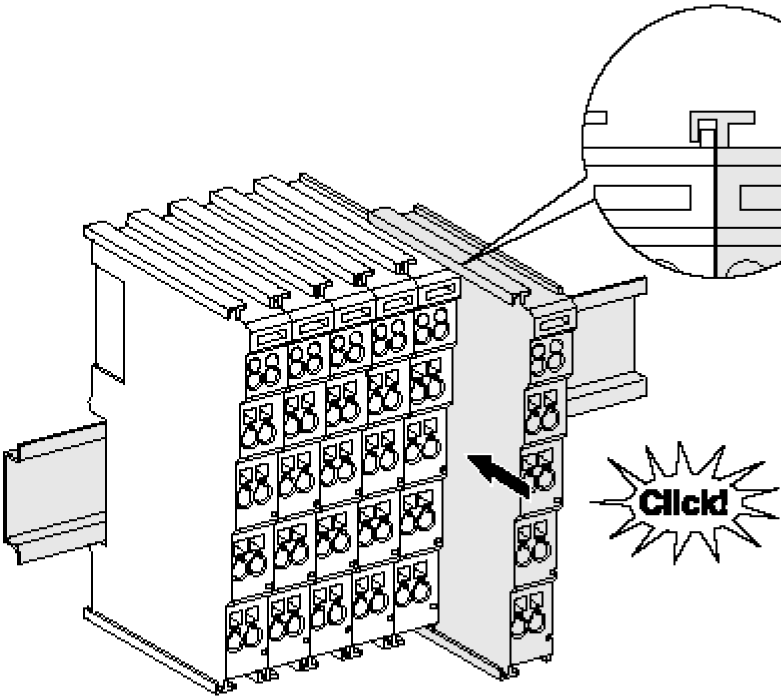


Fig. 23: Attaching on mounting rail

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

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

Fixing of mounting rails

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

Disassembly

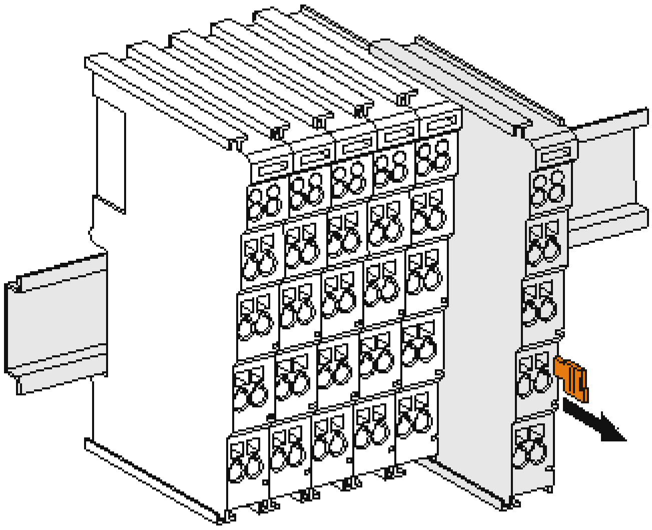


Fig. 24: Disassembling of terminal

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

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

Connections within a bus terminal block

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

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

Power Contacts

During the design of a bus terminal block, the pin assignment of the individual Bus Terminals must be taken account of, since some types (e.g. analog Bus Terminals or digital 4-channel Bus Terminals) do not or not fully loop through the power contacts. Power Feed Terminals (KL91xx, KL92xx or EL91xx, EL92xx) interrupt the power contacts and thus represent the start of a new supply rail.

PE power contact

The power contact labeled PE can be used as a protective earth. For safety reasons this contact mates first when plugging together, and can ground short-circuit currents of up to 125 A.

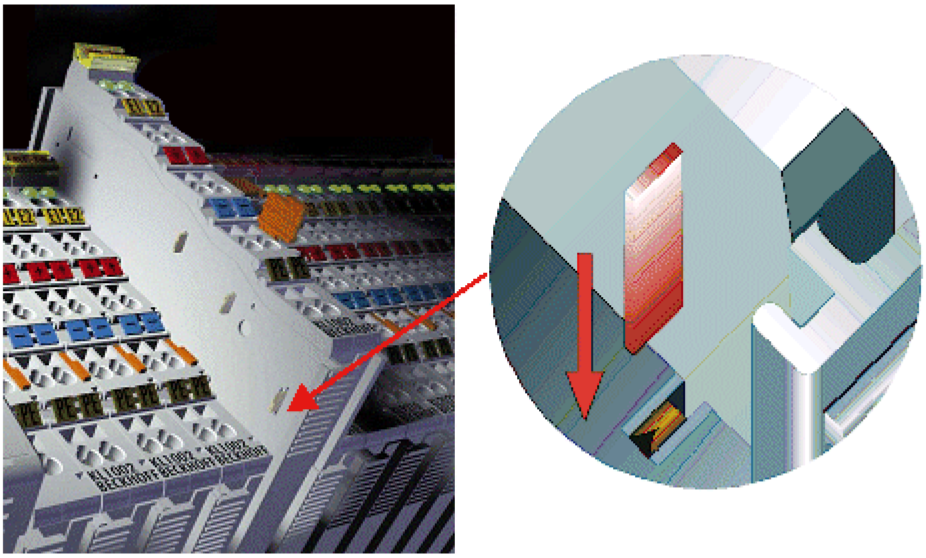


Fig. 25: Power contact on left side

NOTICE

Possible damage of the device

Note that, for reasons of electromagnetic compatibility, the PE contacts are capacitatively coupled to the mounting rail. This may lead to incorrect results during insulation testing or to damage on the terminal (e.g. disruptive discharge to the PE line during insulation testing of a consumer with a nominal voltage of 230 V). For insulation testing, disconnect the PE supply line at the Bus Coupler or the Power Feed Terminal! In order to decouple further feed points for testing, these Power Feed Terminals can be released and pulled at least 10 mm from the group of terminals.

⚠ WARNING

Risk of electric shock!

The PE power contact must not be used for other potentials!

5.5 Installation instructions for enhanced mechanical load capacity

⚠ WARNING

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

Bring the Bus Terminal system into a safe, de-energized state before starting mounting, disassembly or wiring of the Bus Terminals!

Additional checks

The terminals have undergone the following additional tests:

Verification	Explanation
Vibration	10 frequency runs in 3 axes
	6 Hz < f < 60 Hz displacement 0.35 mm, constant amplitude
	60.1 Hz < f < 500 Hz acceleration 5 *g*, constant amplitude
Shocks	1000 shocks in each direction, in 3 axes
	25 *g*, 6 ms

Additional installation instructions and notes

For terminals with enhanced mechanical load capacity, the following additional installation instructions and notes apply:

- The enhanced mechanical load capacity is valid for all permissible installation positions.
- Use a mounting rail according to EN 60715 TH35-15.
- Fix the terminal segment on both sides of the mounting rail with a mechanical fixture, e.g. an earth terminal or reinforced end clamp.
- The maximum total extension of the terminal segment (without coupler) is:
  64 terminals (12 mm mounting width) or 32 terminals (24 mm mounting width)
- Avoid deformation, twisting, crushing and bending of the mounting rail during edging and installation of the rail.
- The mounting points of the mounting rail must be set at 5 cm intervals.
- Use countersunk head screws to fasten the mounting rail.
- The free length between the strain relief and the wire connection should be kept as short as possible. A distance of approx. 10 cm should be maintained to the cable duct.

5.6 Connection

5.6.1 Connection system

⚠ WARNING

Risk of electric shock and damage of device!

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

Overview

The bus terminal system offers different connection options for optimum adaptation to the respective application:

- The terminals of ELxxxx and KLxxxx series with standard wiring include electronics and connection level in a single enclosure.
- The terminals of ESxxxx and KSxxxx series feature a pluggable connection level and enable steady wiring while replacing.
- The High Density Terminals (HD Terminals) include electronics and connection level in a single enclosure and have advanced packaging density.

Standard wiring (ELxxxx / KLxxxx)

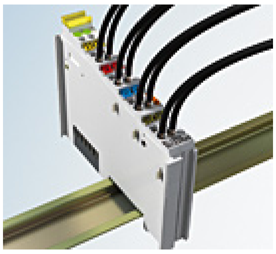


Fig. 26: Standard wiring

The terminals of the ELxxxx and KLxxxx series integrate screwless spring-cage technology for quick and easy wiring.

Pluggable wiring (ESxxxx / KSxxxx)

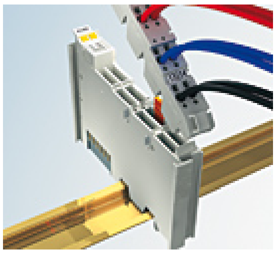


Fig. 27: Pluggable wiring

The terminals of ESxxxx and KSxxxx series feature a pluggable connection level.
The assembly and wiring procedure is the same as for the ELxxxx and KLxxxx series.
The pluggable connection level enables the complete wiring to be removed as a plug connector from the top of the housing for servicing.
The lower section can be removed from the terminal block by pulling the unlocking tab.
Insert the new component and plug in the connector with the wiring. This reduces the installation time and eliminates the risk of wires being mixed up.

The familiar dimensions of the terminal only had to be changed slightly. The new connector adds about 3 mm. The maximum height of the terminal remains unchanged.

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

Conductor cross sections between 0.08 mm^2 and 2.5 mm^2 can continue to be used with the proven spring force technology.

The overview and nomenclature of the product names for ESxxxx and KSxxxx series has been retained as known from ELxxxx and KLxxxx series.

High Density Terminals (HD Terminals)

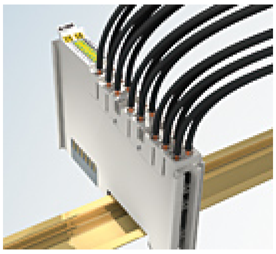


Fig. 28: High Density Terminals

The terminals from these series with 16/32 terminal points are distinguished by a particularly compact design, as the packaging density is twice as large as that of the standard 12 mm bus terminals. Massive conductors and conductors with a wire end sleeve can be inserted directly into the spring loaded terminal point without tools.

Wiring HD Terminals

The High Density Terminals of the ELx8xx and KLx8xx series doesn't support pluggable wiring.

Ultrasonically compacted (ultrasonically welded) strands

Ultrasonically compacted (ultrasonically welded) strands

Ultrasonically compacted (ultrasonically welded) strands can also be connected to the standard and high-density terminals. In this case, please note the tables concerning the wire-size width [▶ 54]!

5.6.2 Wiring

⚠ WARNING

Risk of electric shock and damage of device!

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

Terminals for standard wiring ELxxxx/KLxxxx and for pluggable wiring ESxxxx/KSxxxx

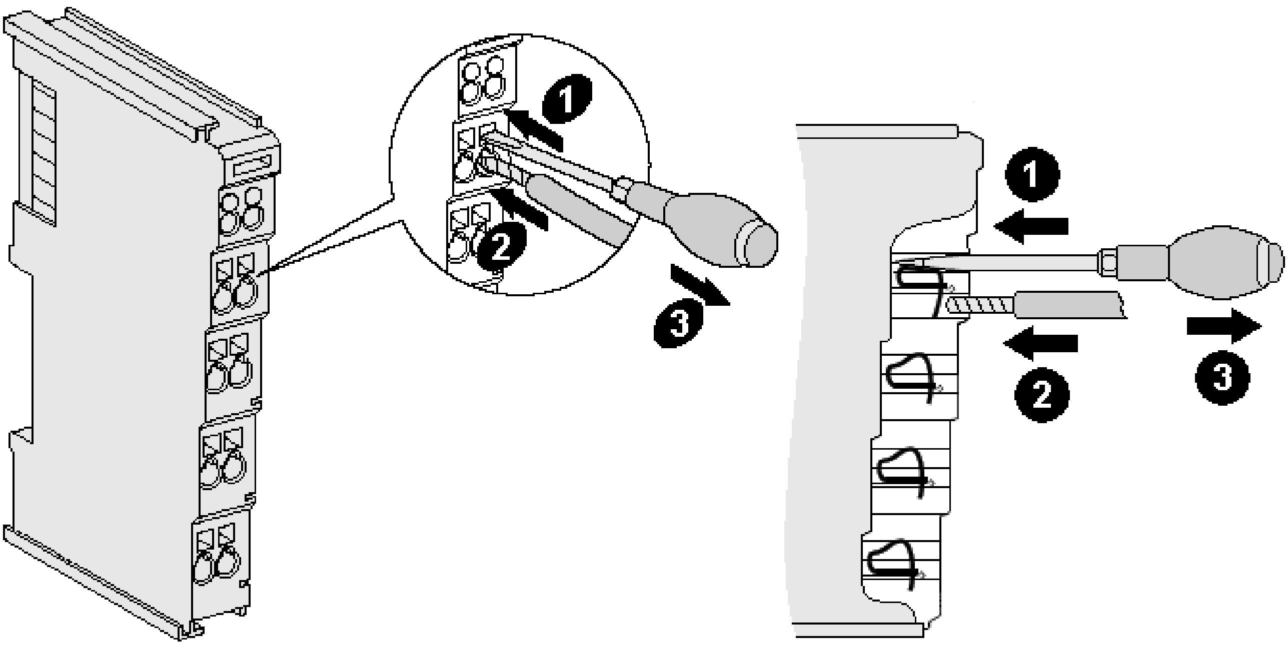


Fig. 29: Connecting a cable on a terminal point

Up to eight terminal points enable the connection of solid or finely stranded cables to the bus terminal. The terminal points are implemented in spring force technology. Connect the cables as follows (see fig. “Connecting a cable on a terminal point”:

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

See the following table for the suitable wire size width:

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width (single core wires)	0.08 ... 2.5 mm^2	0.08 ... 2.5 mm^2
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm^2	0.08 ... 2.5 mm^2
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm^2	0.14 ... 1.5 mm^2
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals (HD Terminals [▸ 52]) with 16/32 terminal points

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

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm^2
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm^2
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm^2
Wire size width (ultrasonically compacted [ultrasonically welded] strands)	only 1.5 mm^2 (see notice [▸ 52])
Wire stripping length	8 ... 9 mm

5.6.3 Shielding

Shielding

Encoder, analog sensors and actuators should always be connected with shielded, twisted paired wires.

5.7 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 supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV supply also requires a safe connection to the protective conductor.

5.8 Installation positions

NOTICE

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.

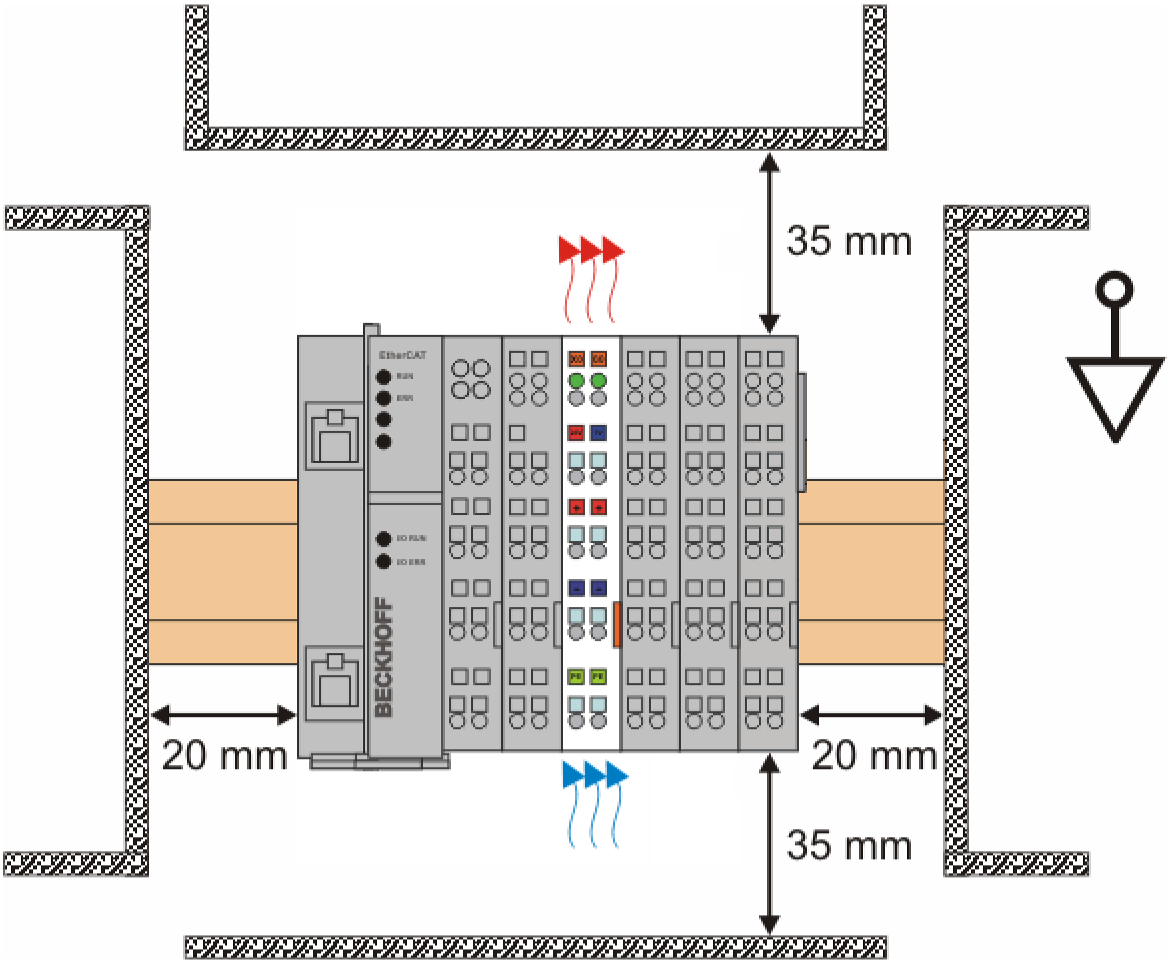


Fig. 30: Recommended distances for standard installation position

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.

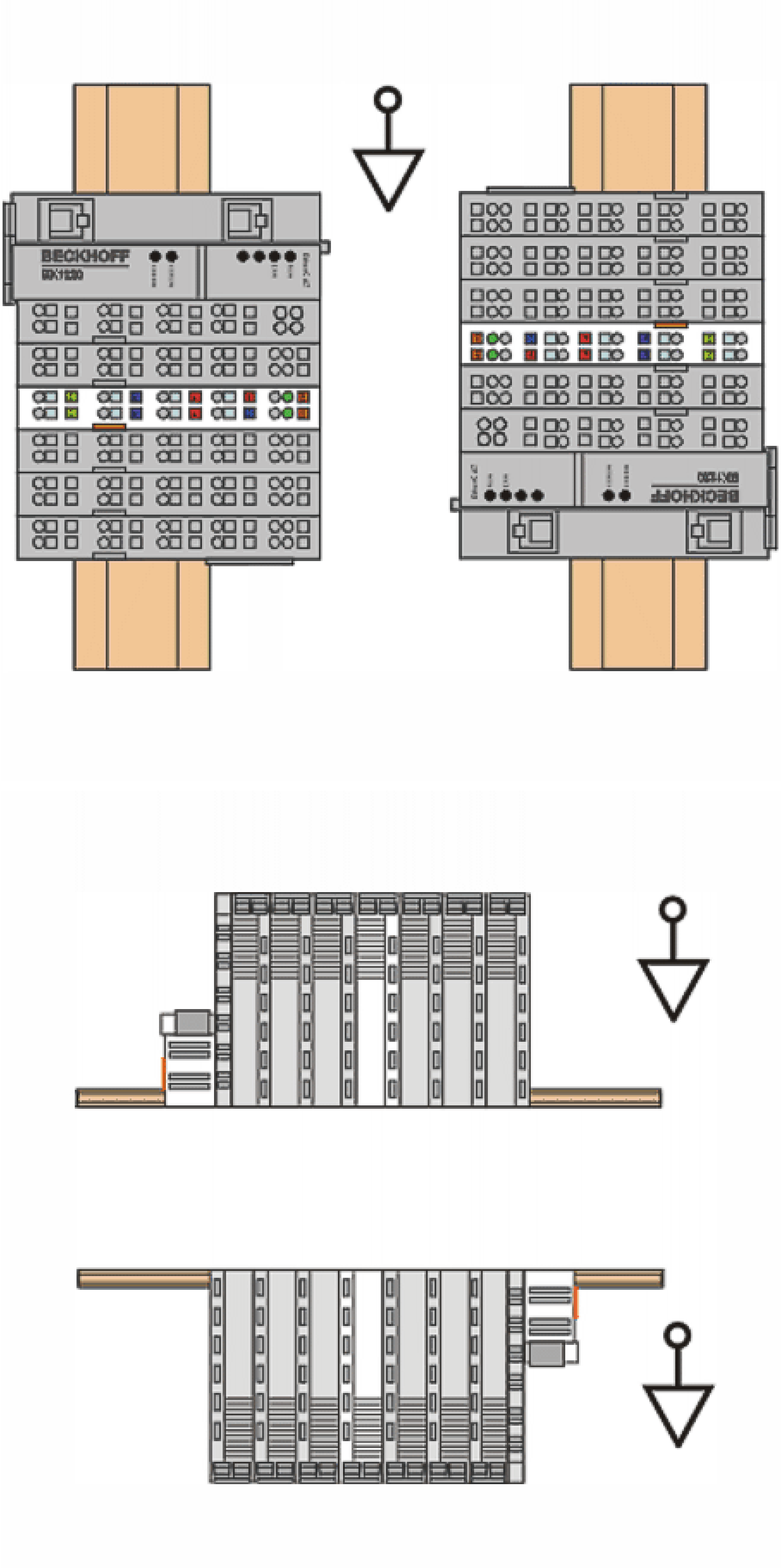


Fig. 31: Other installation positions

5.9 Positioning of passive Terminals

Hint for positioning of passive terminals in the bus terminal block

EtherCAT 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 two passive terminals!

Examples for positioning of passive terminals (highlighted)

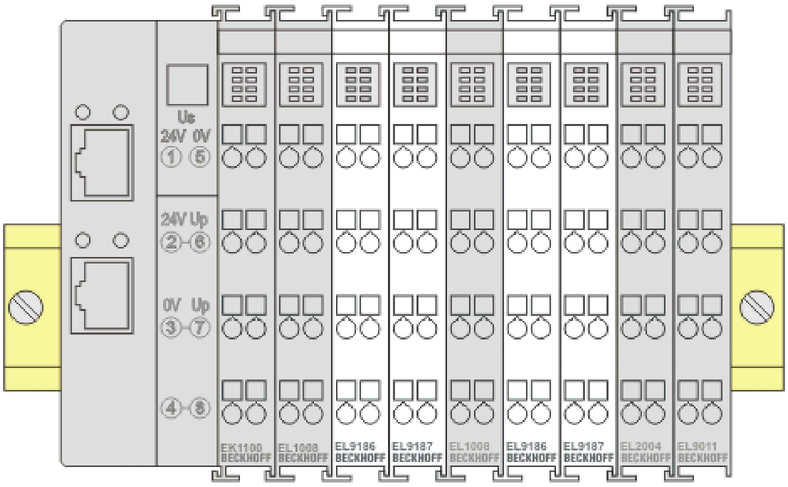


Fig. 32: Correct positioning

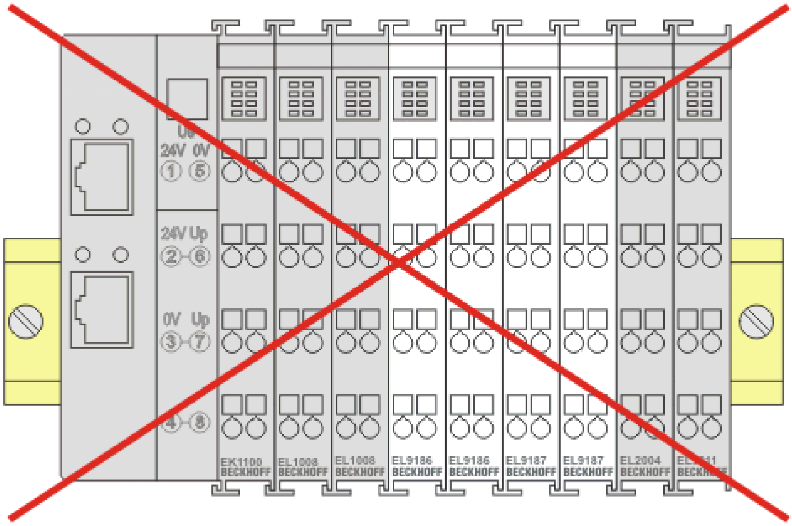


Fig. 33: Incorrect positioning

5.10 Disposal

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

6 Commissioning

6.1 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:

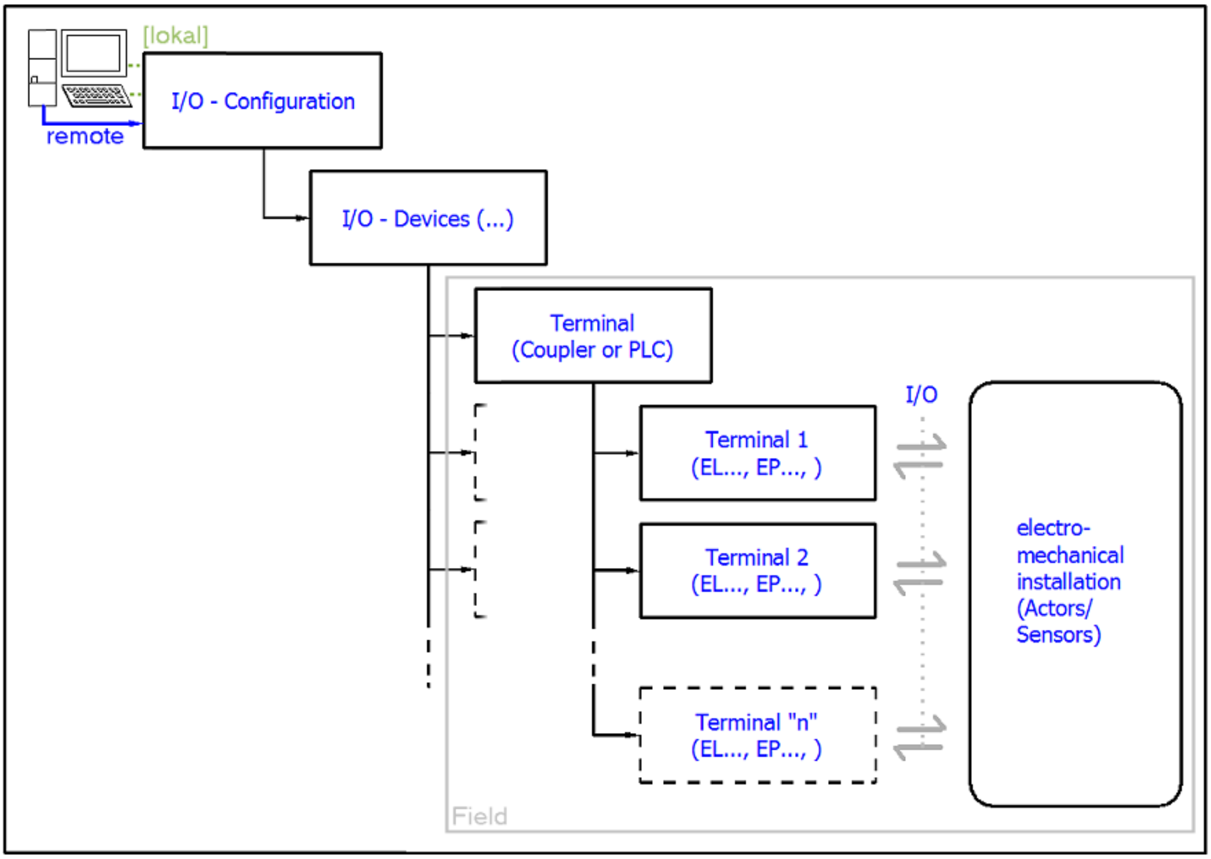


Fig. 34: Relationship between user side (commissioning) and installation

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)

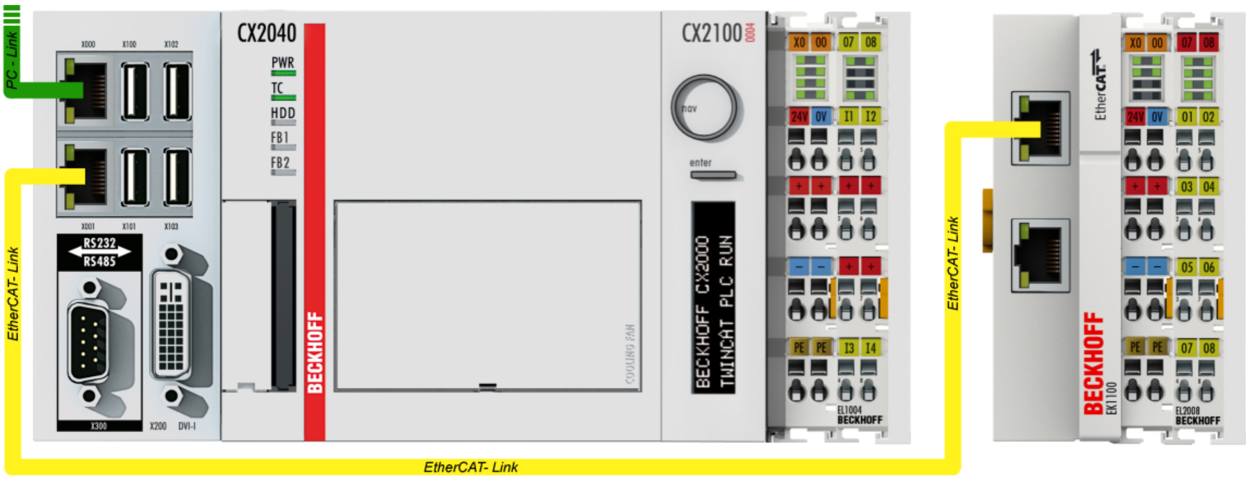


Fig. 35: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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.

6.1.1 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:

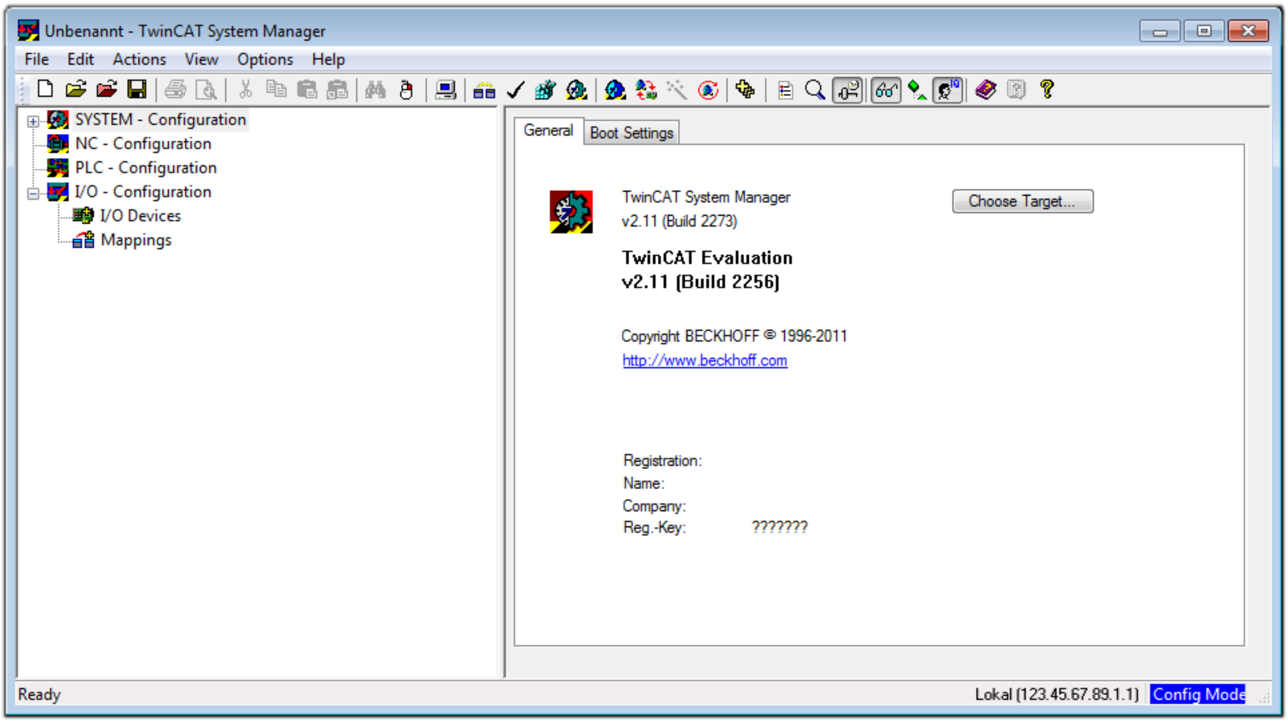


Fig. 36: Initial TwinCAT 2 user interface

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 [▸ 65]".

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:

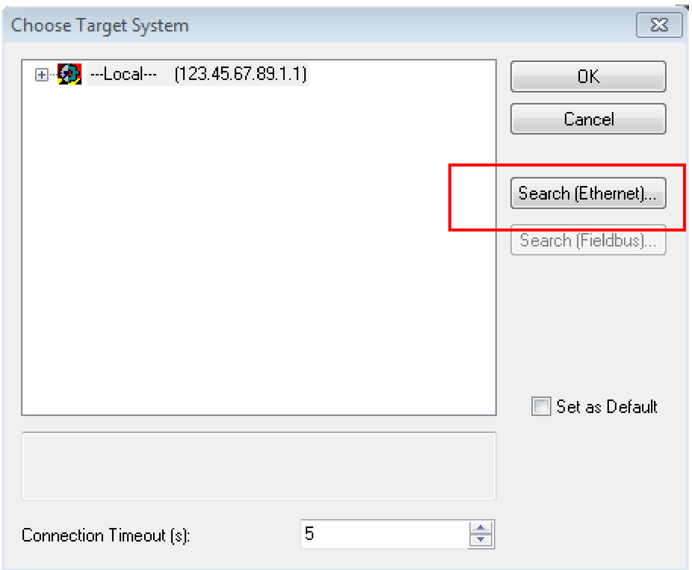


Fig. 37: Selection of the target system

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

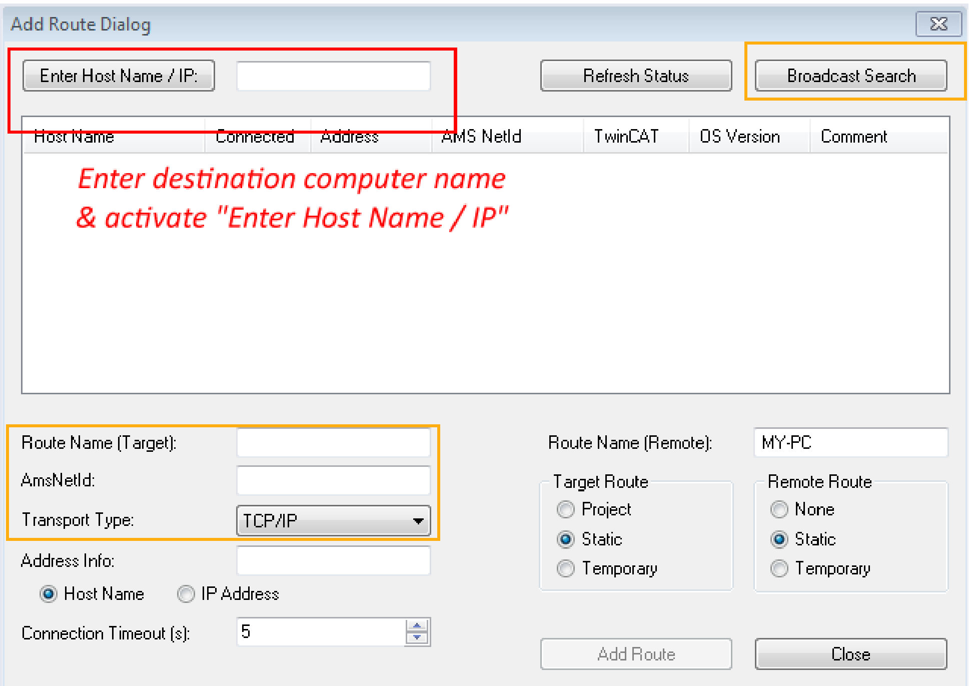
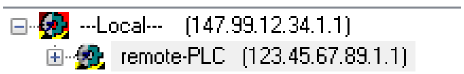


Fig. 38: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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).

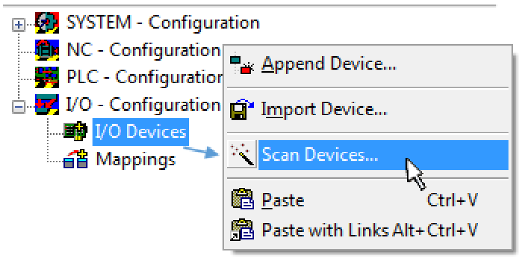


Fig. 39: Select “Scan Devices...”

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

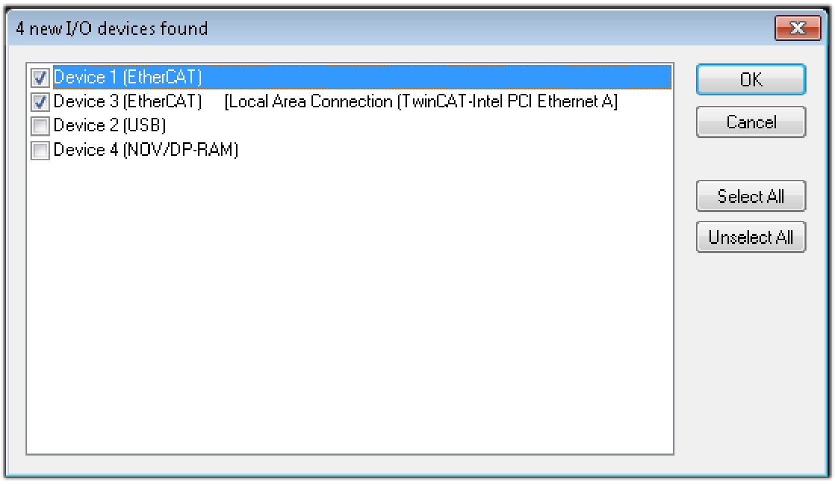


Fig. 40: Automatic detection of I/O devices: selection of the devices to be integrated

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 [▸ 61] described at the beginning of this section, the result is as follows:

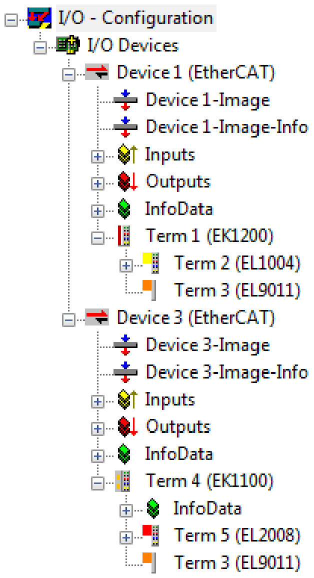


Fig. 41: Mapping of the configuration in the TwinCAT 2 System Manager

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:

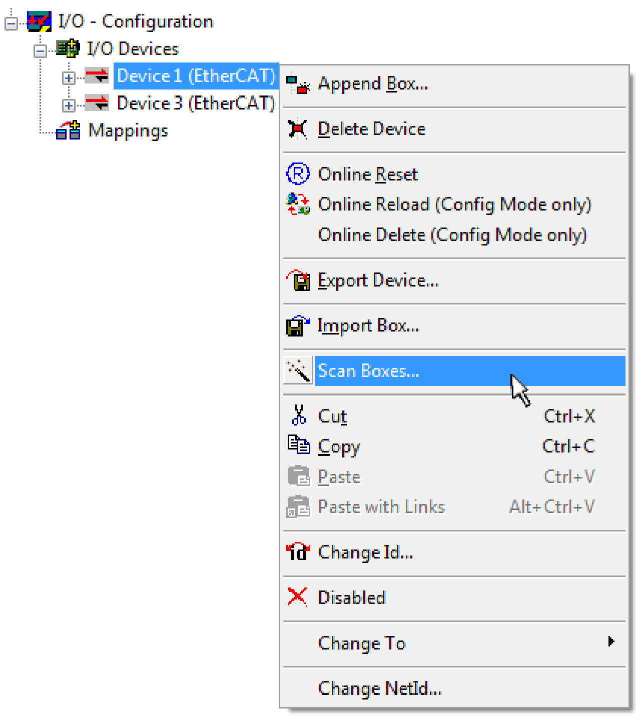


Fig. 42: Reading of individual terminals connected to a device

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:

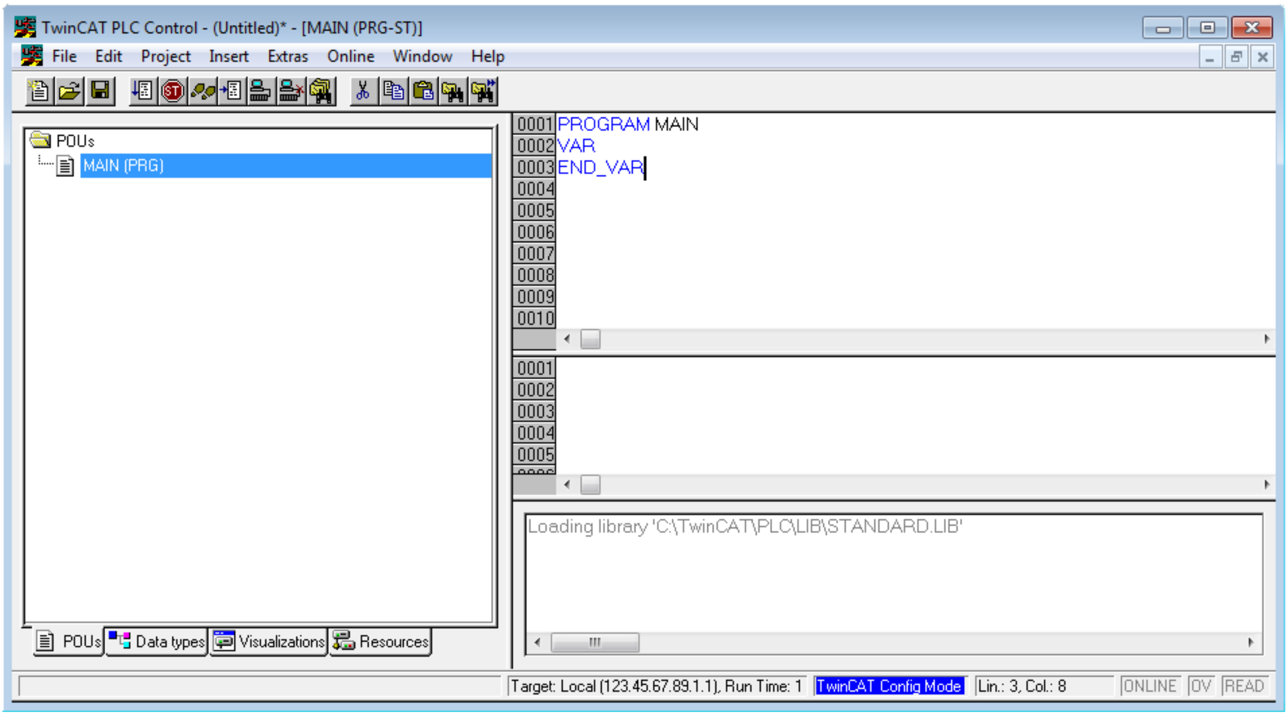


Fig. 43: TwinCAT PLC Control after startup

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

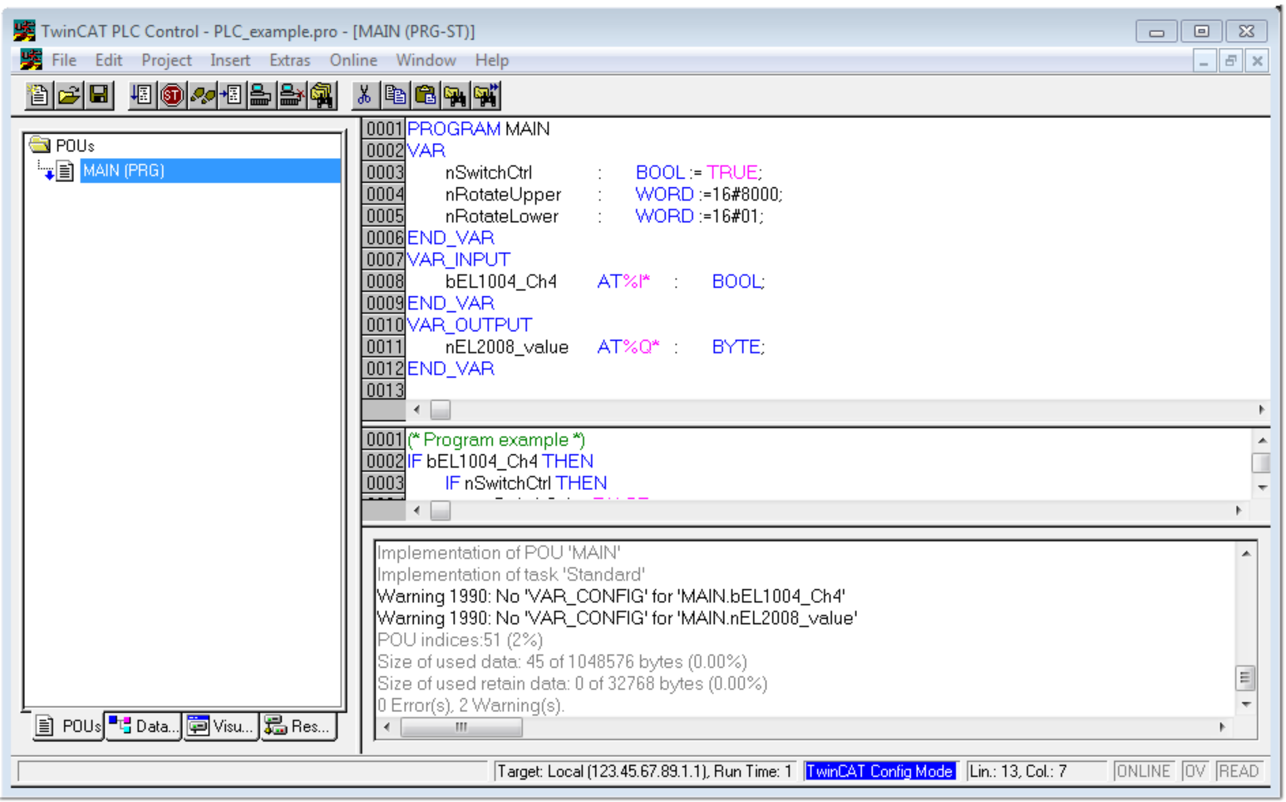


Fig. 44: Example program with variables after a compile process (without variable integration)

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…”:

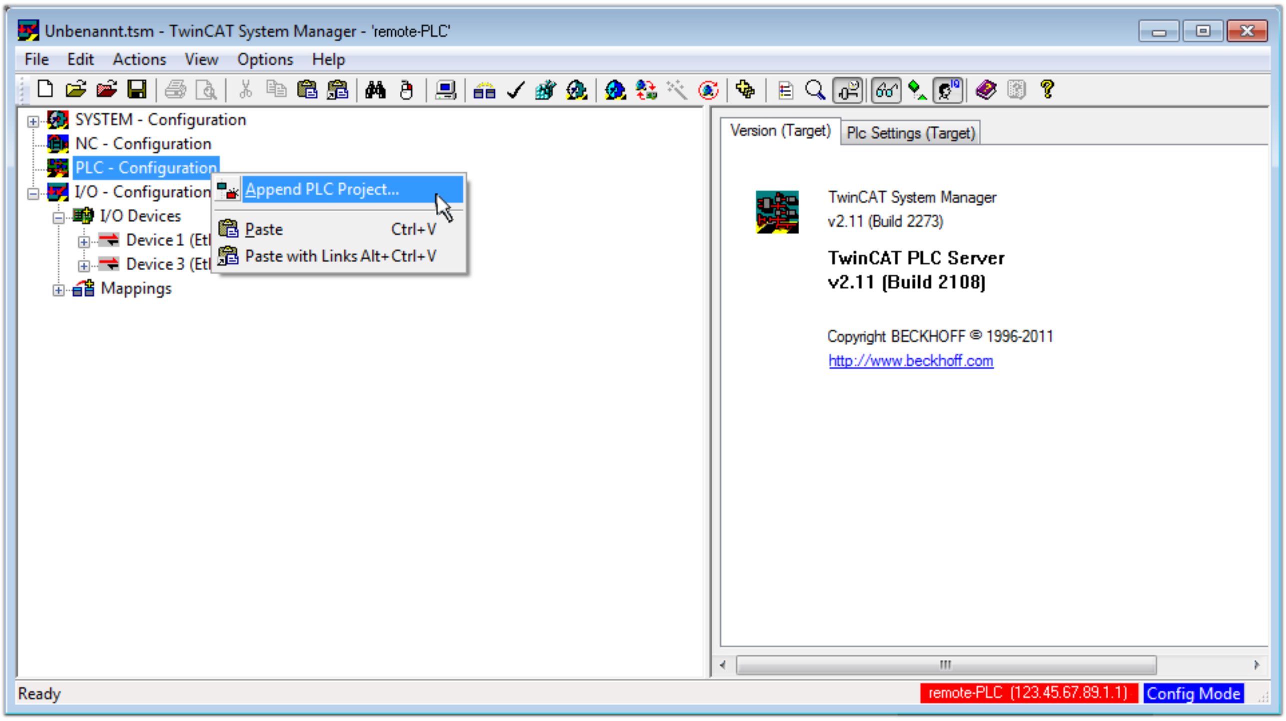


Fig. 45: Appending the TwinCAT PLC Control project

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:

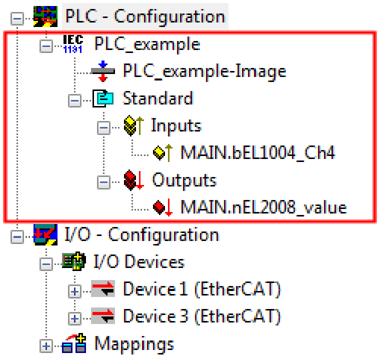


Fig. 46: PLC project integrated in the PLC configuration of the System Manager

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”:

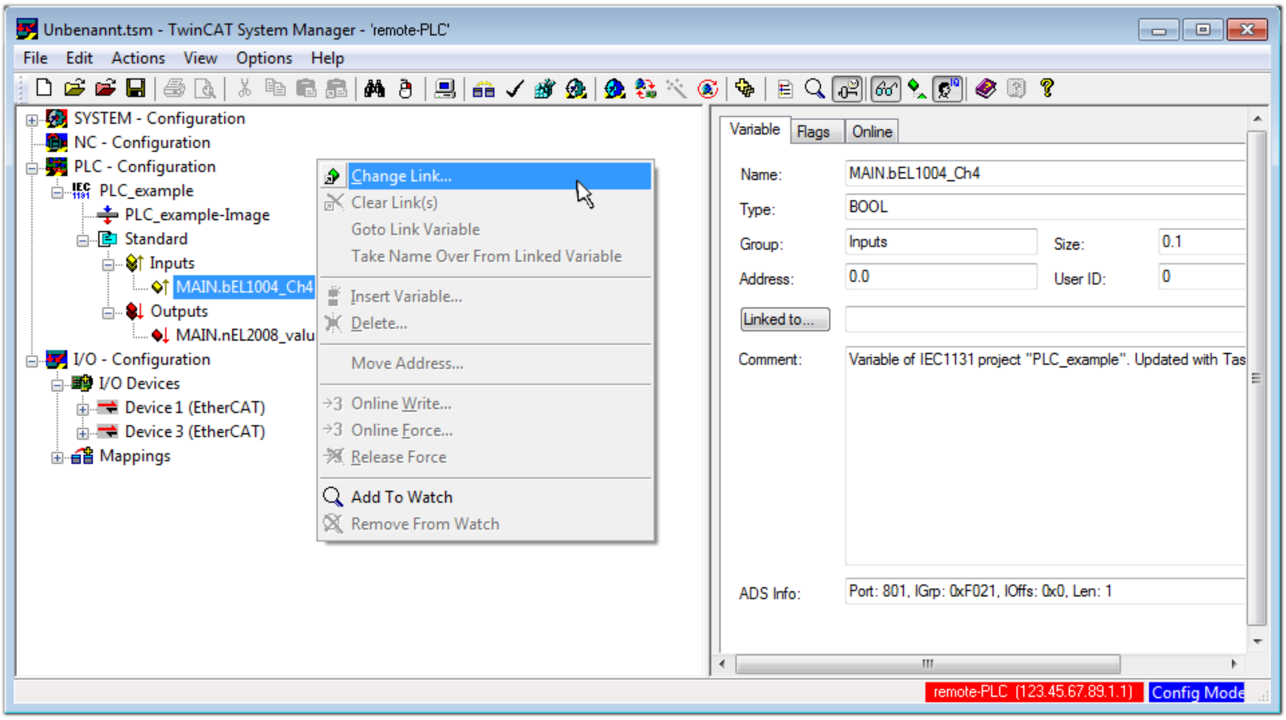


Fig. 47: Creating the links between PLC variables and process objects

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

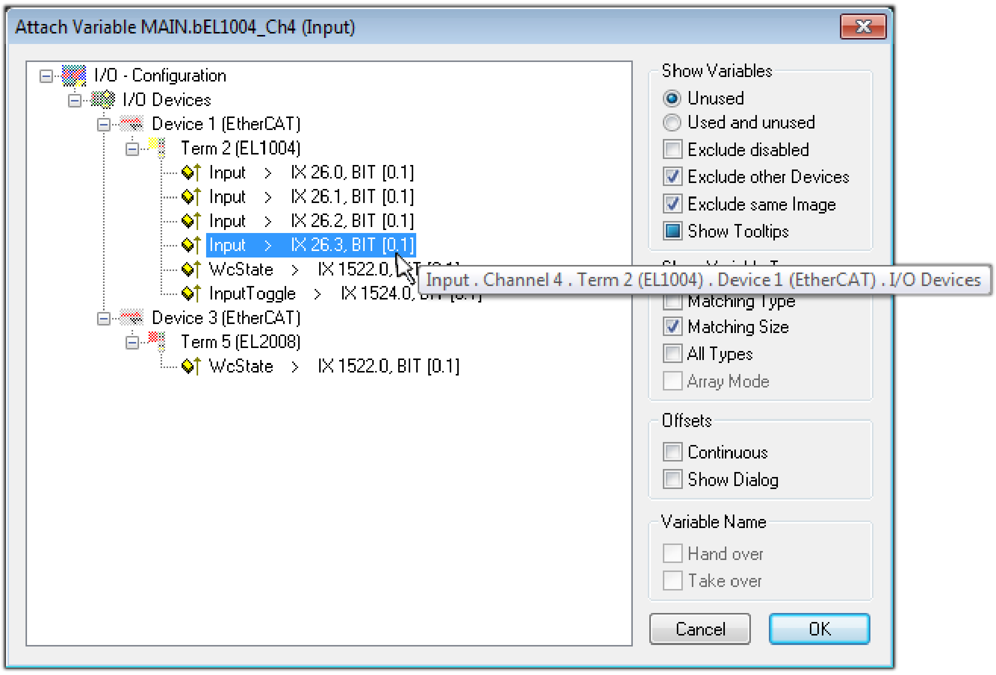


Fig. 48: Selecting BOOL-type PDO

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:

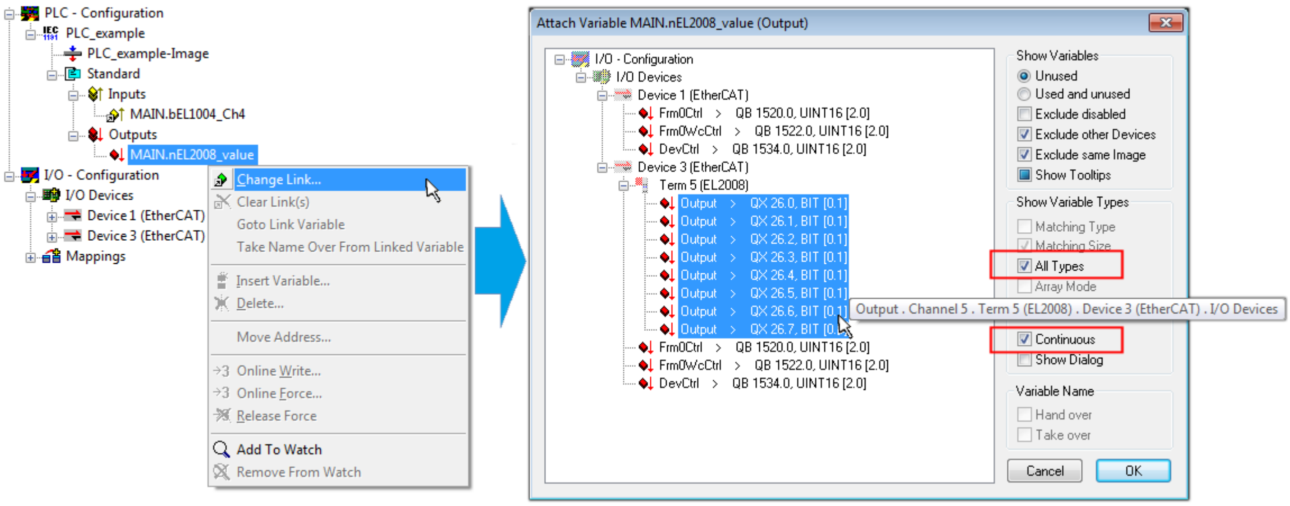


Fig. 49: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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:

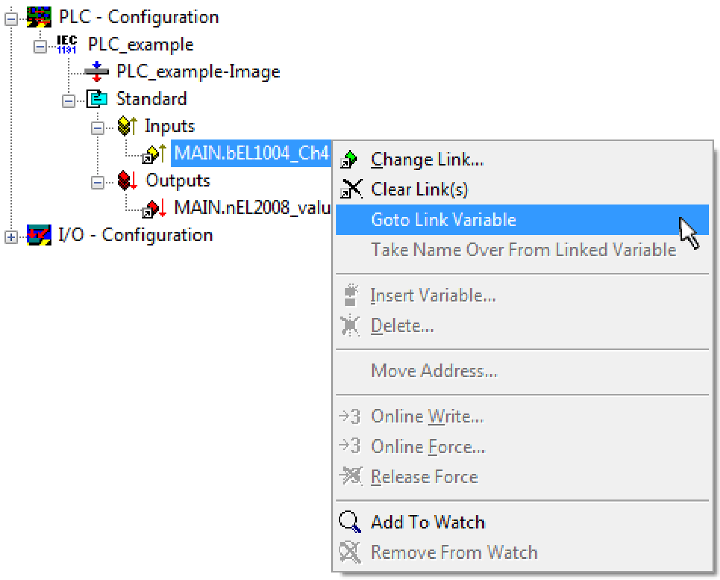
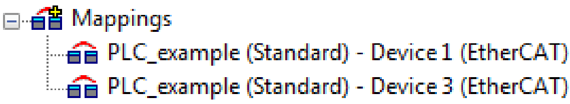


Fig. 50: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

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…”:

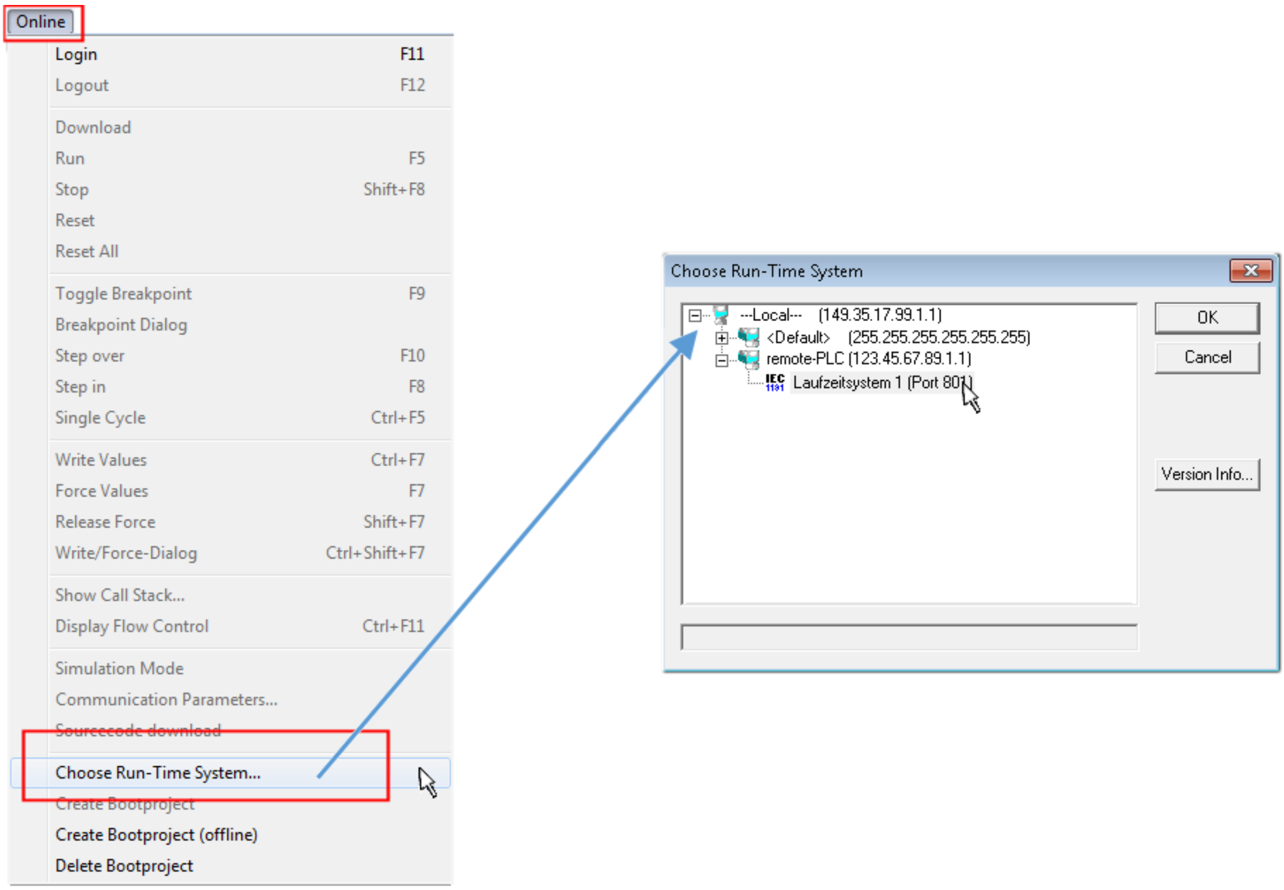
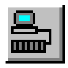


Fig. 51: Choose target system (remote)

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:

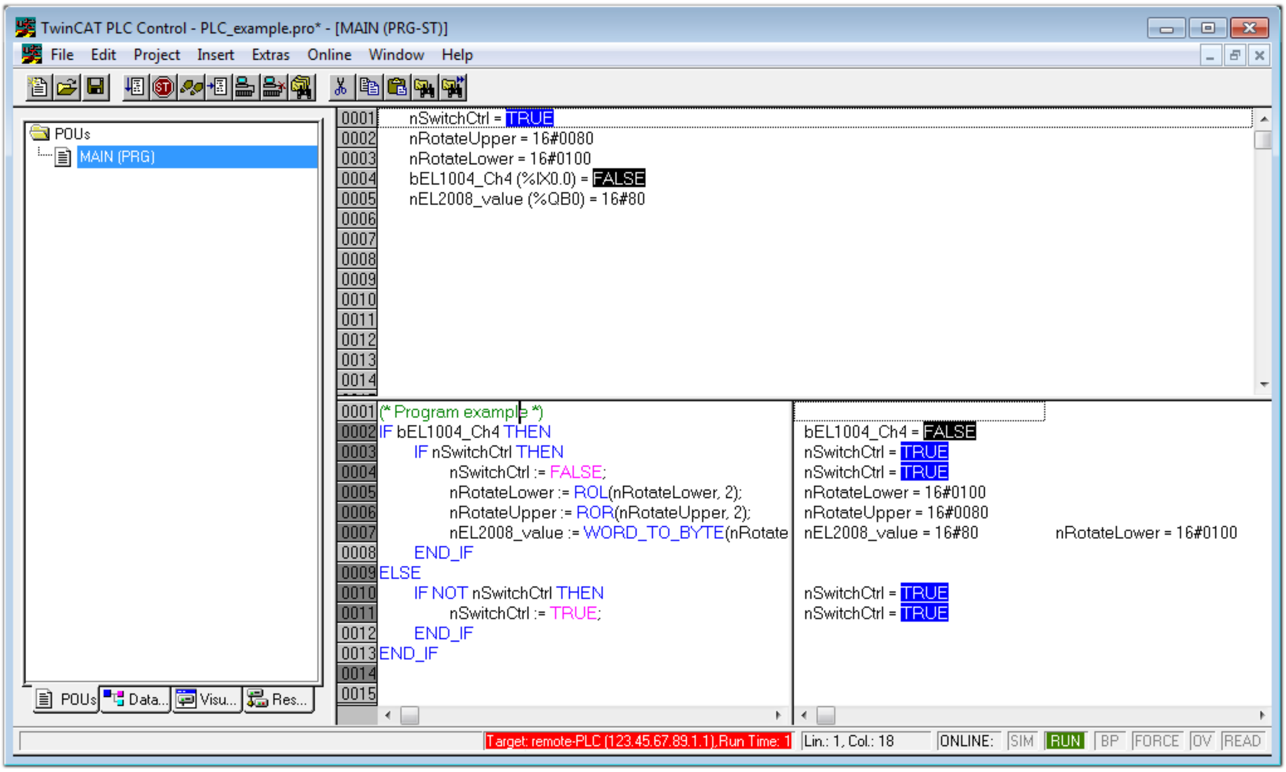


Fig. 52: PLC Control logged in, ready for program startup

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

6.1.2 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:

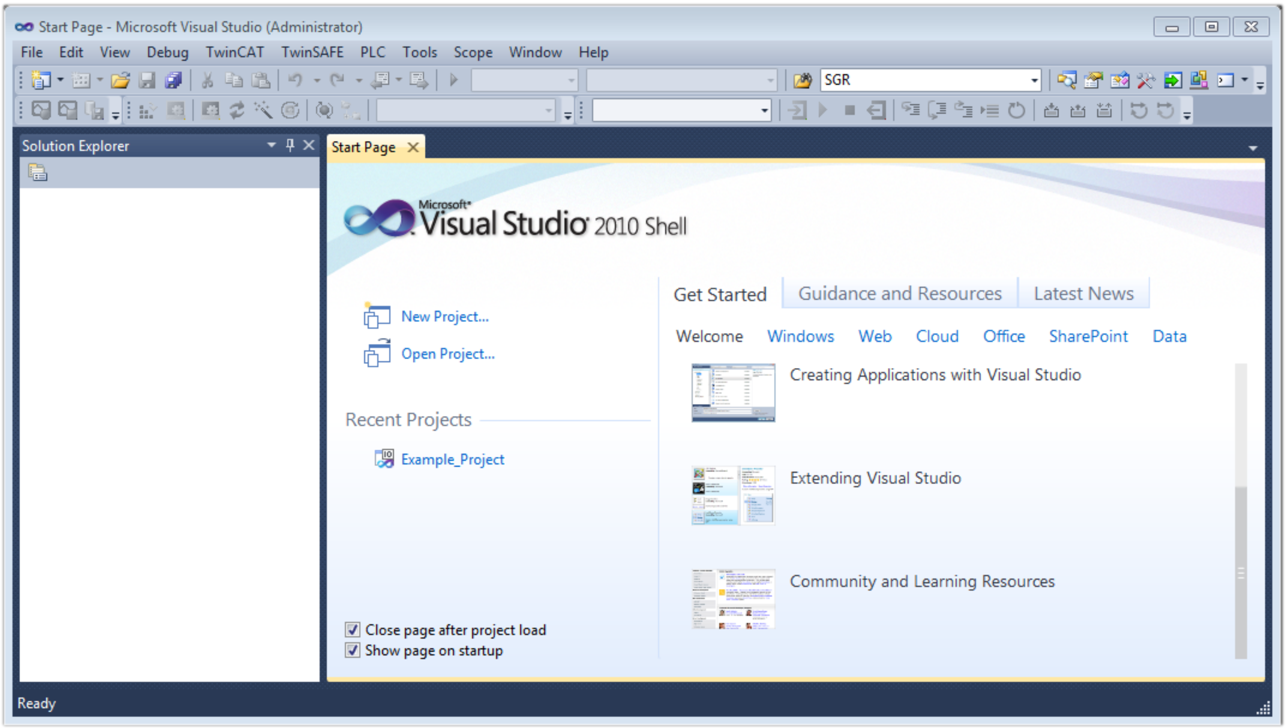


Fig. 53: Initial TwinCAT 3 user interface

First create a new project via 

(or under “File”→“New”→ “Project…”). In the following dialog, make the corresponding entries as required (as shown in the diagram):

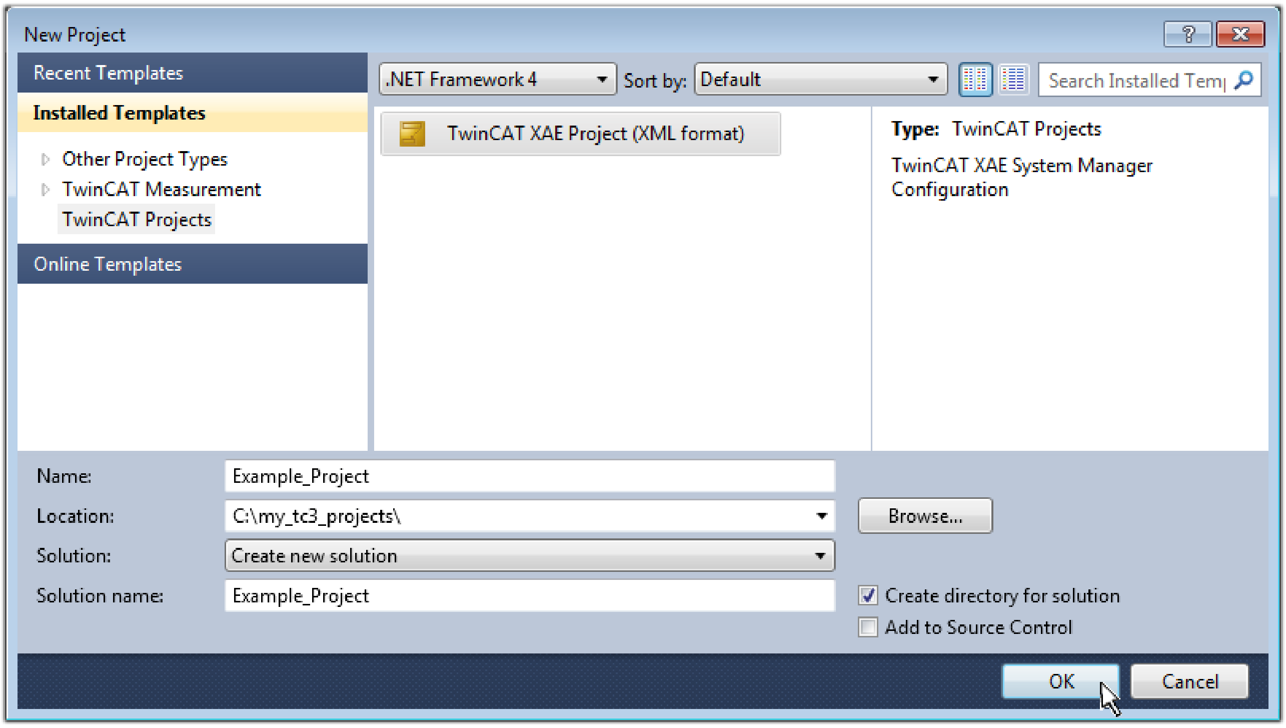


Fig. 54: Create new TwinCAT 3 project

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

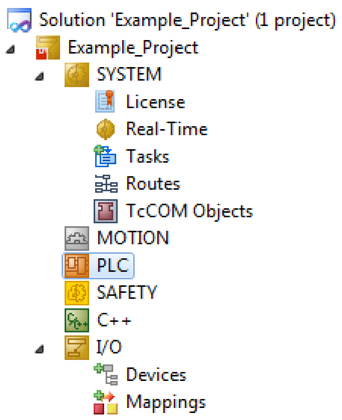
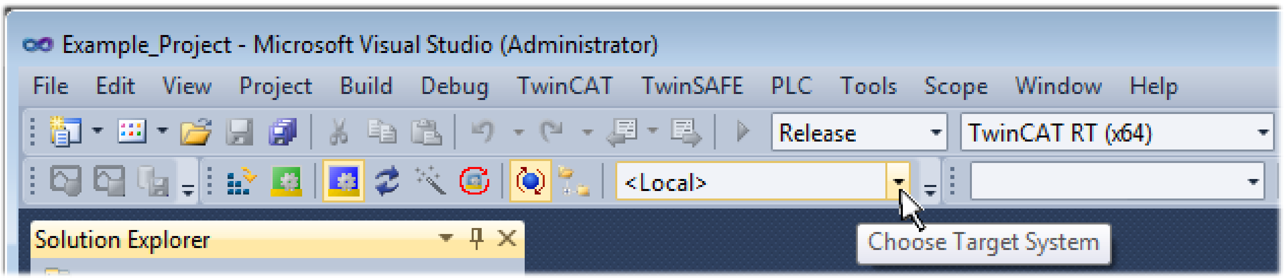


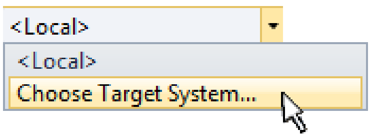
Fig. 55: New TwinCAT 3 project in the project folder explorer

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 [▶ 76]".

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:

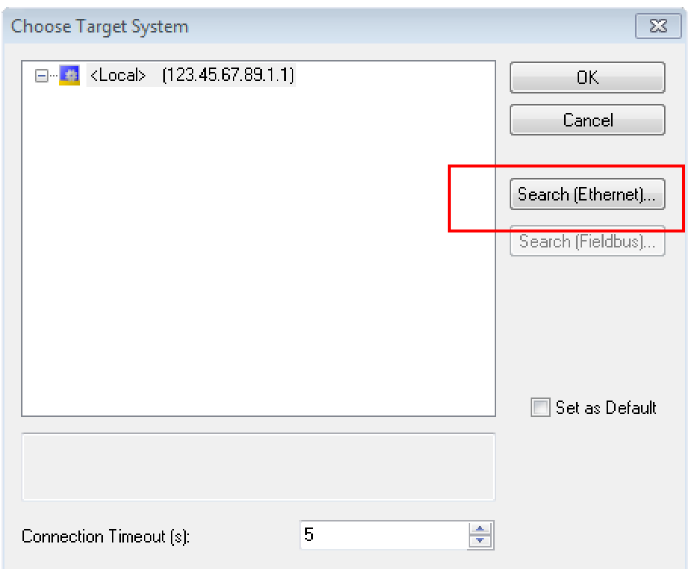


Fig. 56: Selection dialog: Choose the target system

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

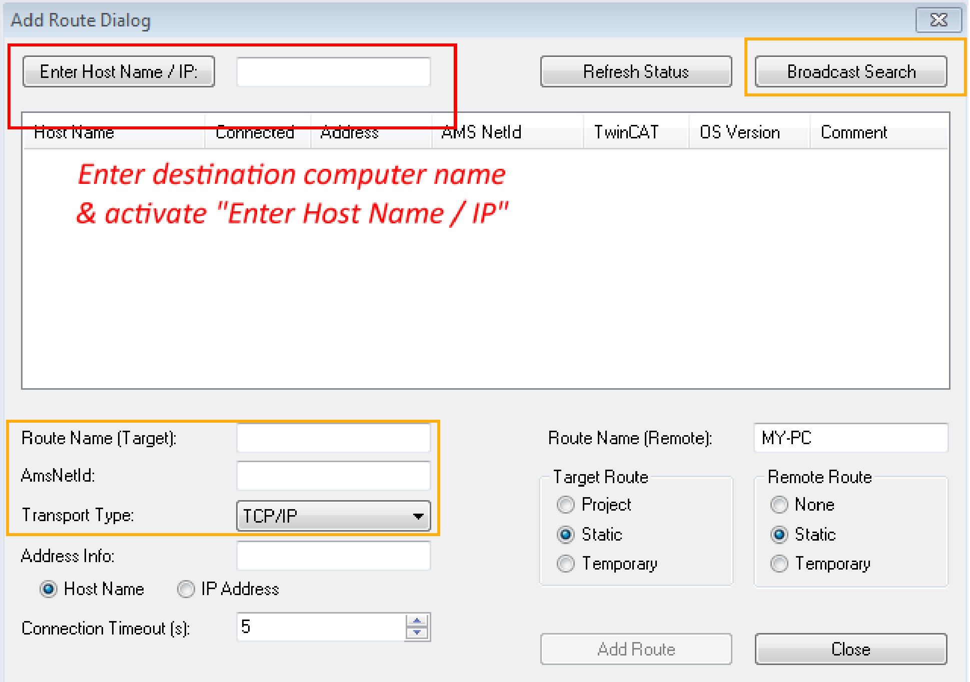
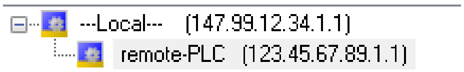


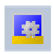
Fig. 57: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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)”.

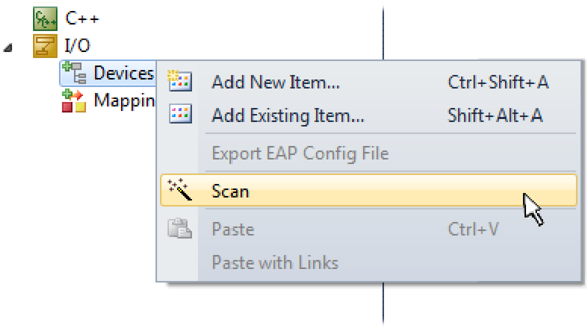


Fig. 58: Select “Scan”

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

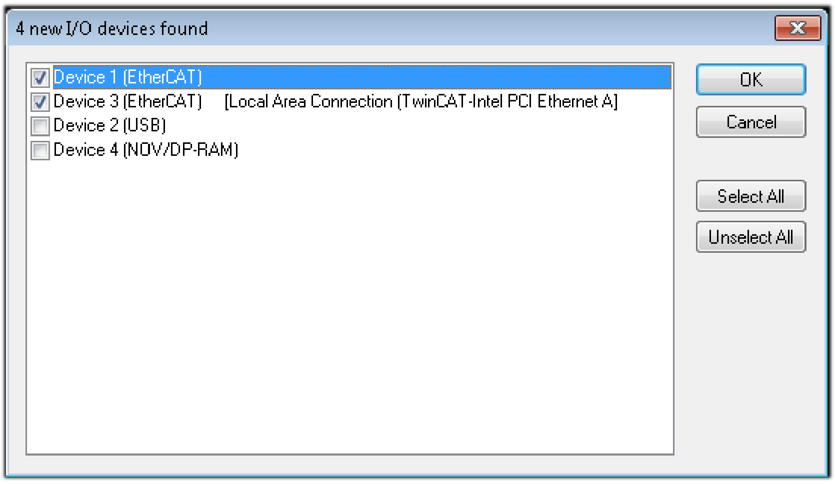


Fig. 59: Automatic detection of I/O devices: selection of the devices to be integrated

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 [▸ 61] described at the beginning of this section, the result is as follows:

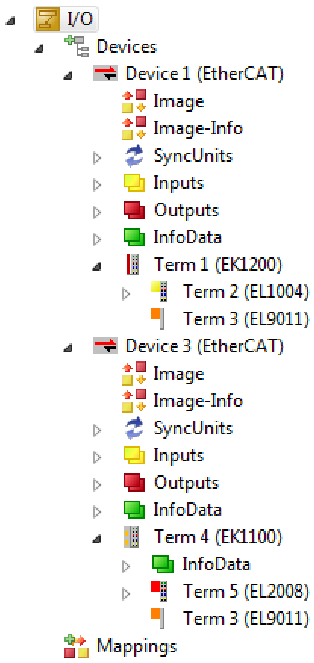


Fig. 60: Mapping of the configuration in VS shell of the TwinCAT 3 environment

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:

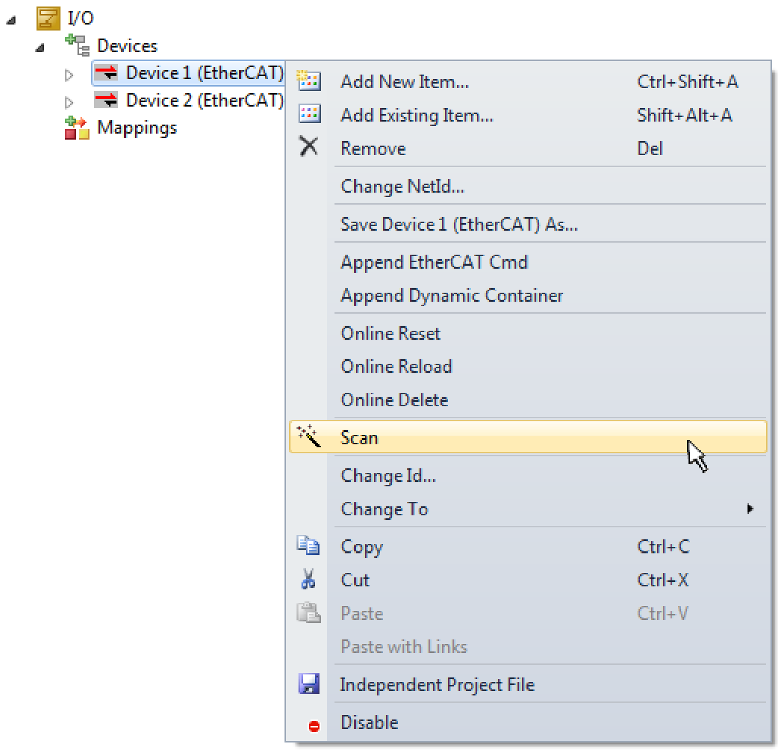


Fig. 61: Reading of individual terminals connected to a device

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….”:

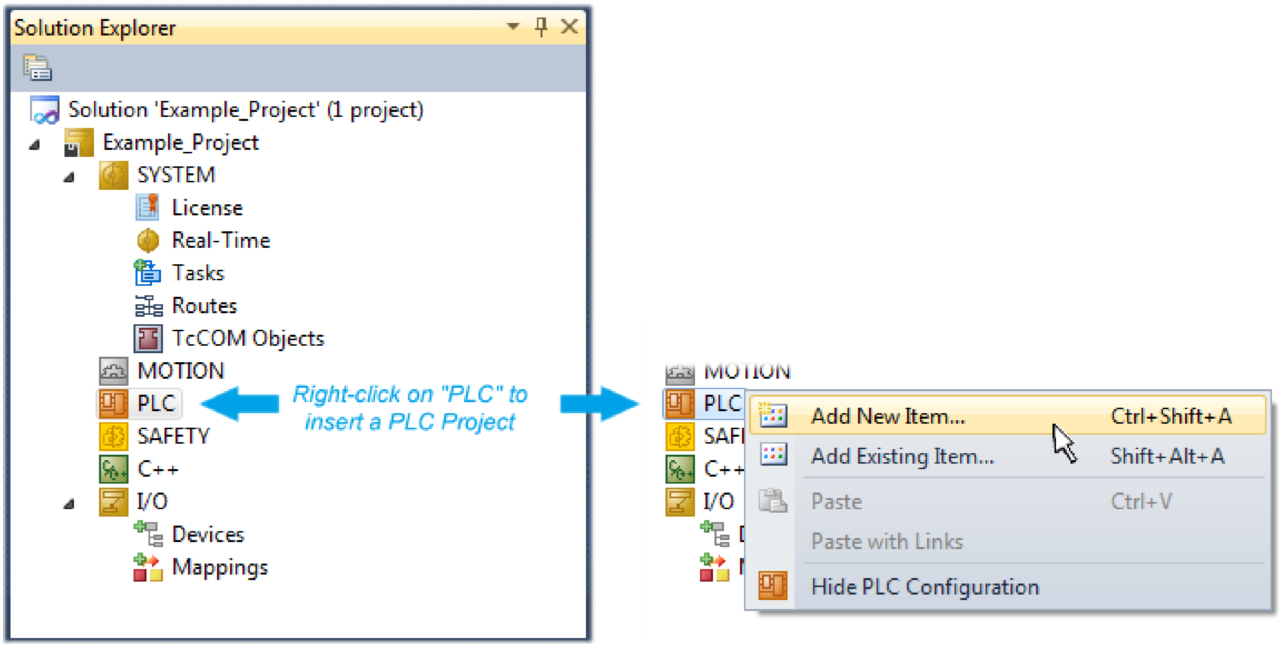


Fig. 62: Adding the programming environment in “PLC”

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

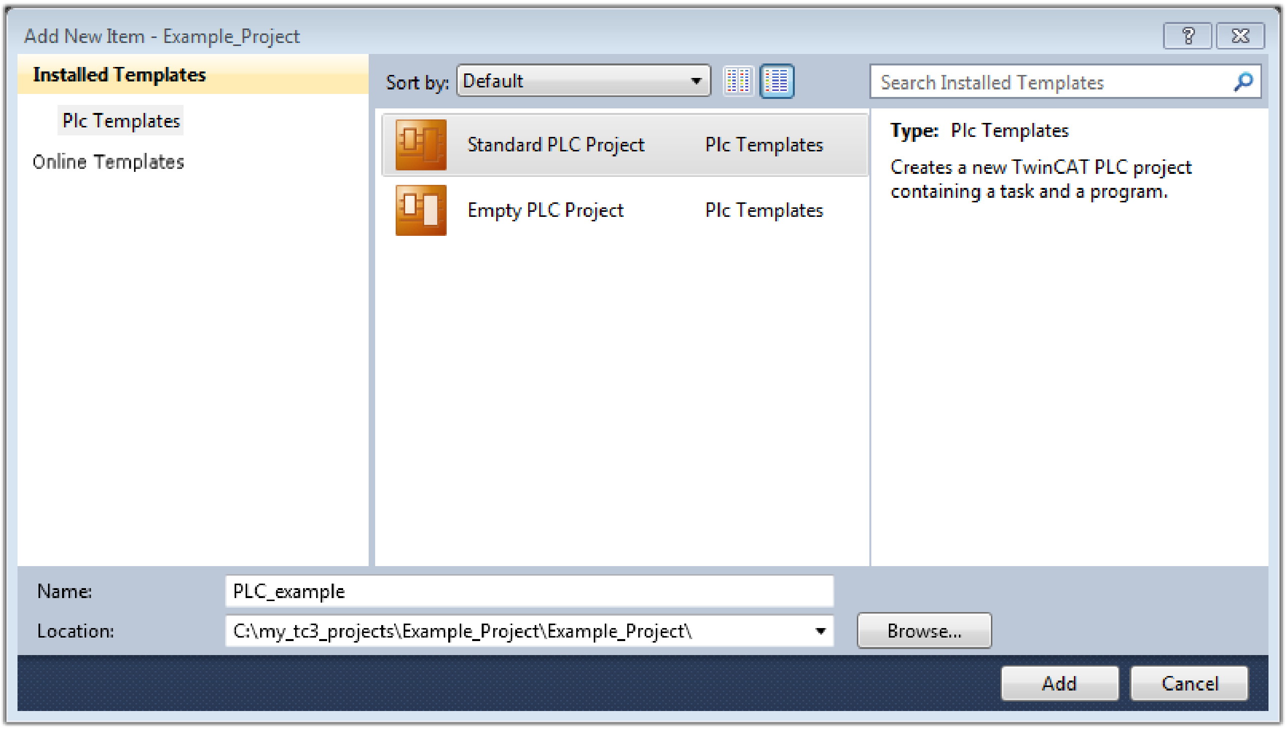


Fig. 63: Specifying the name and directory for the PLC programming environment

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:

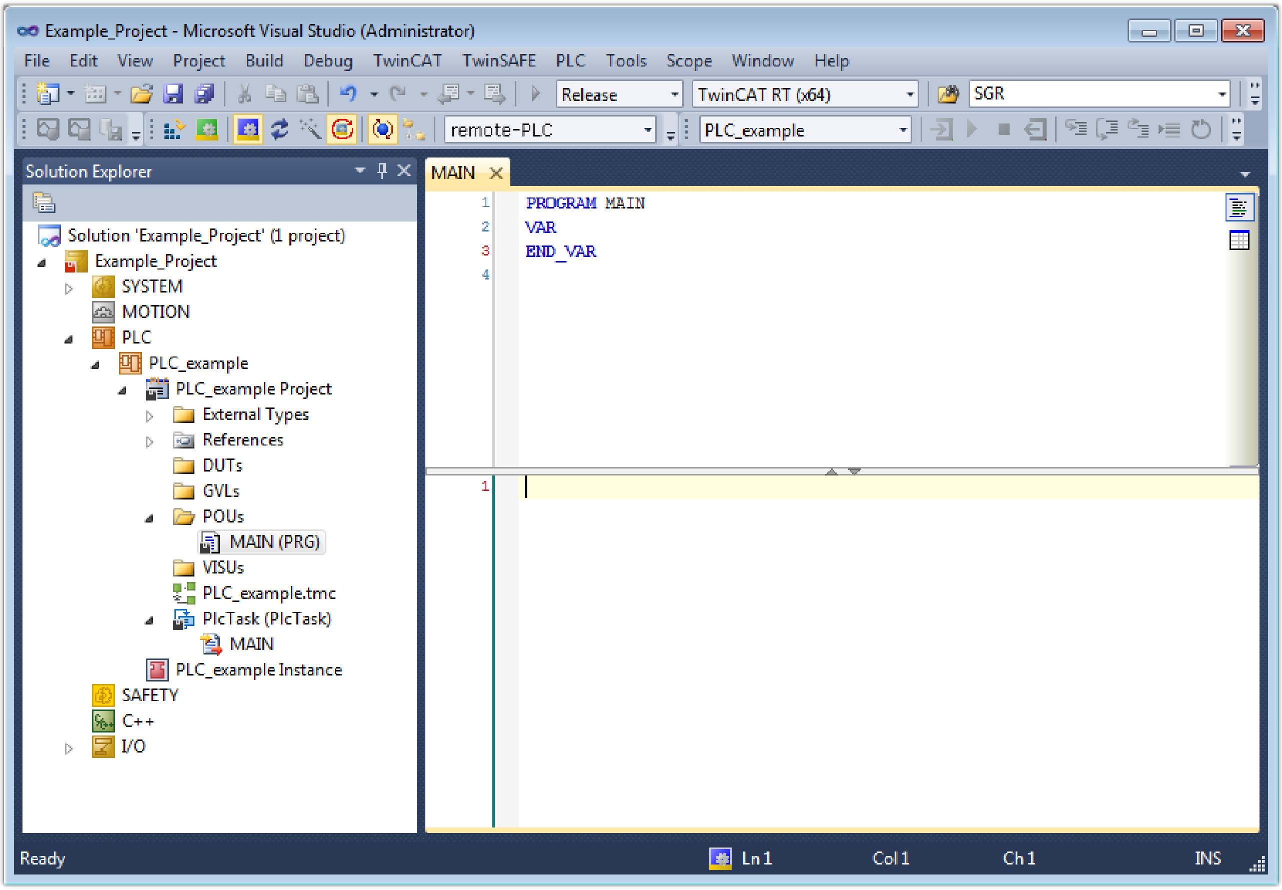


Fig. 64: Initial “Main” program for the standard PLC project

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

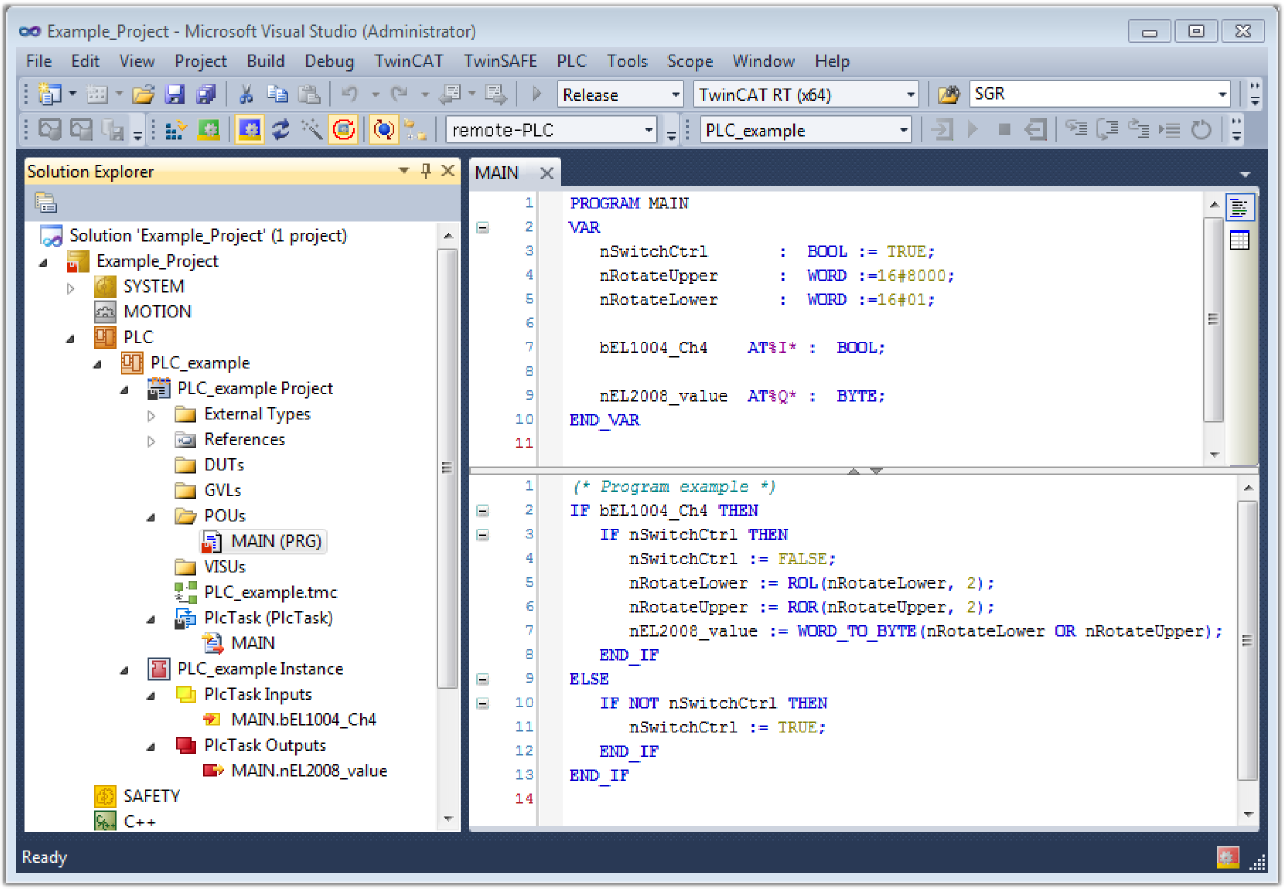


Fig. 65: Example program with variables after a compile process (without variable integration)

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

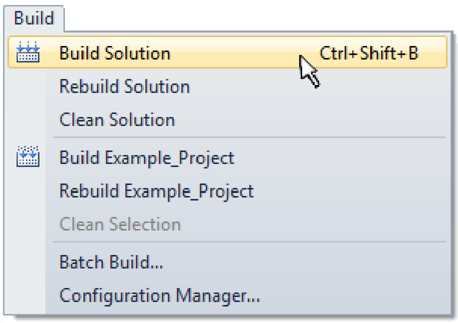
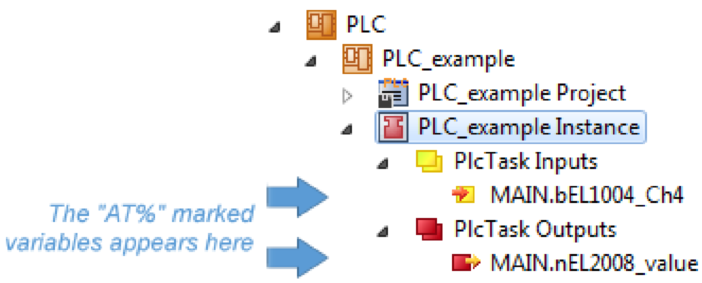


Fig. 66: Start program compilation

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:

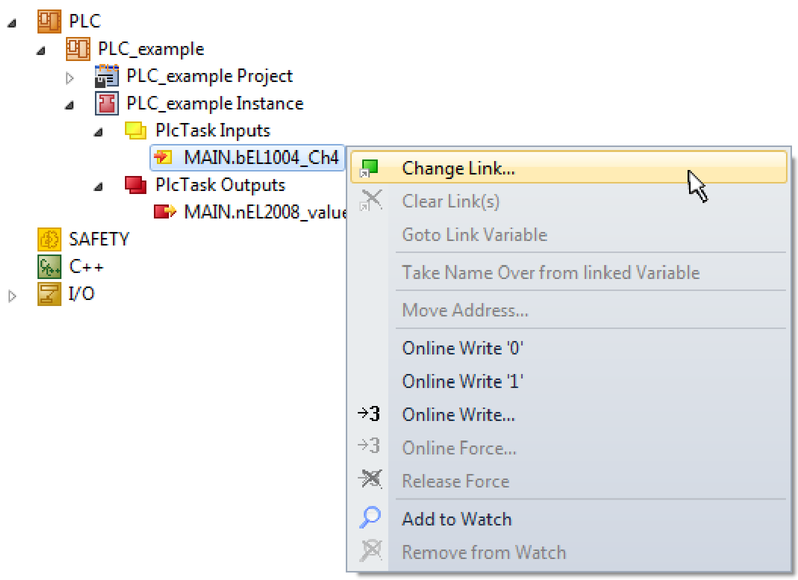


Fig. 67: Creating the links between PLC variables and process objects

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

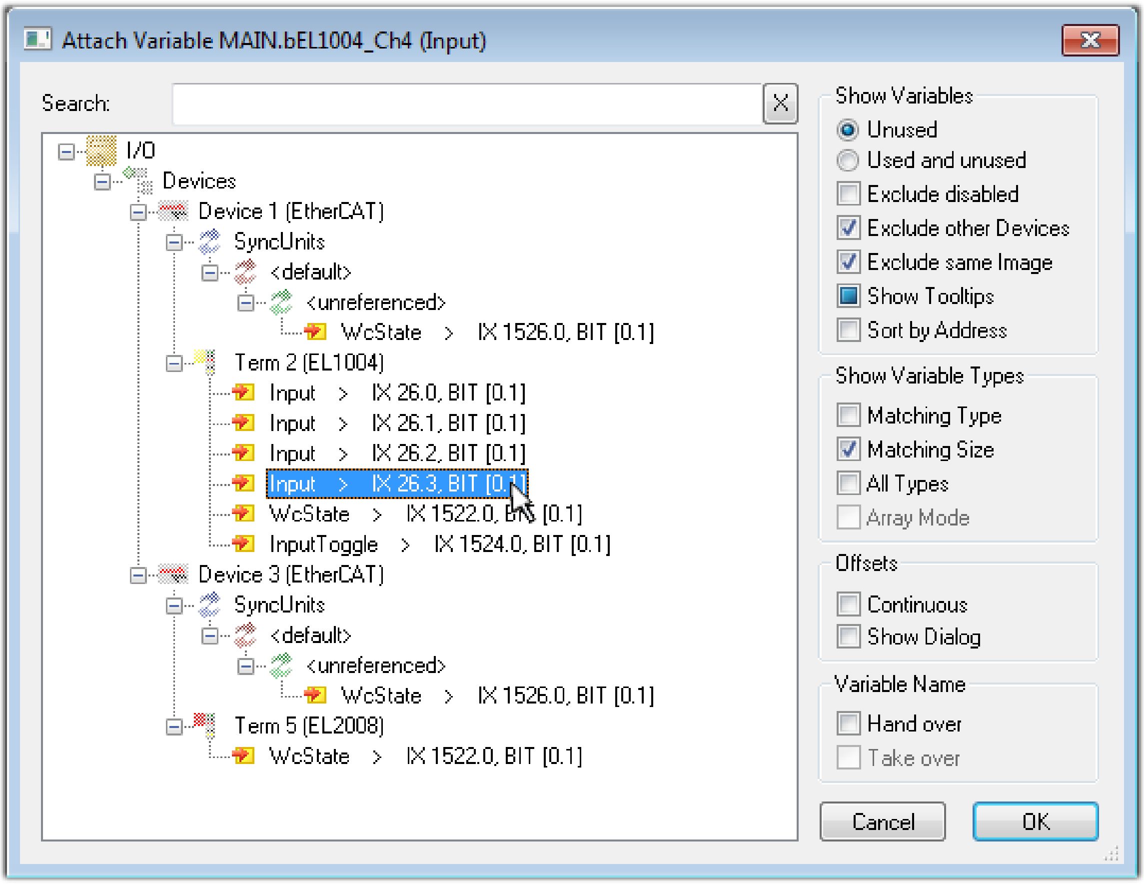


Fig. 68: Selecting BOOL-type PDO

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:

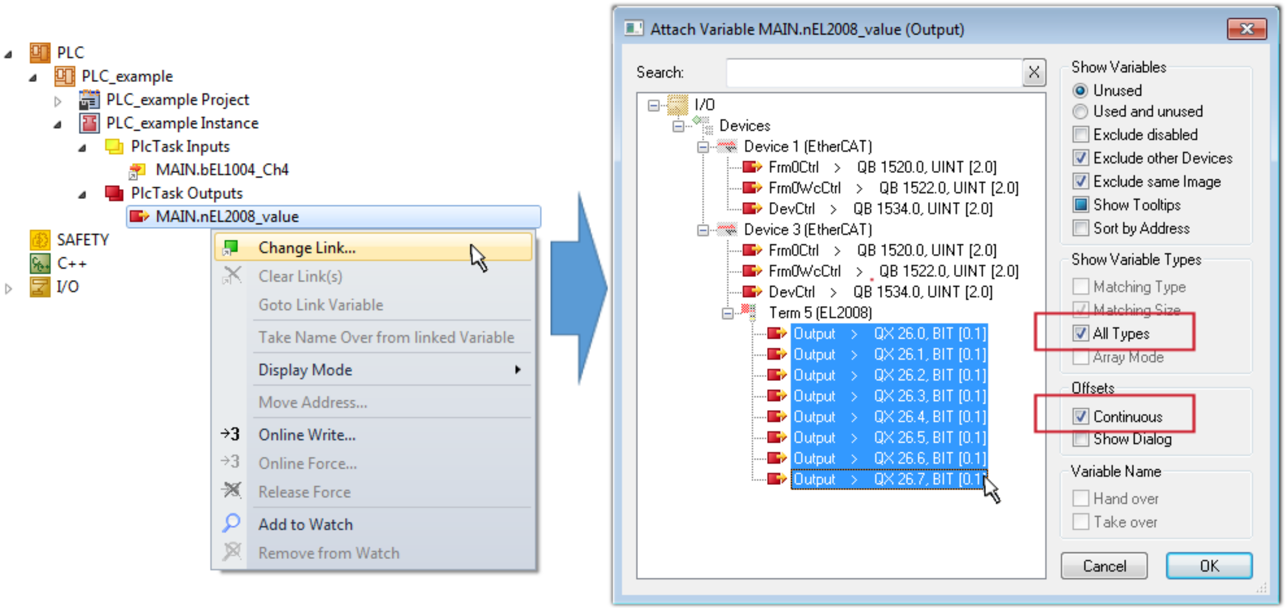


Fig. 69: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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:

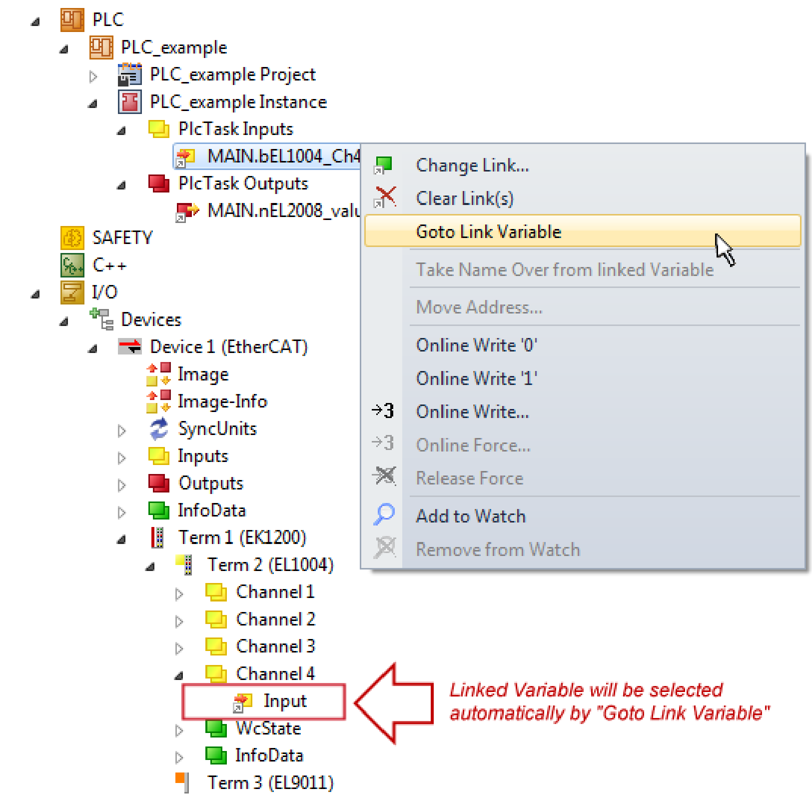


Fig. 70: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

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.

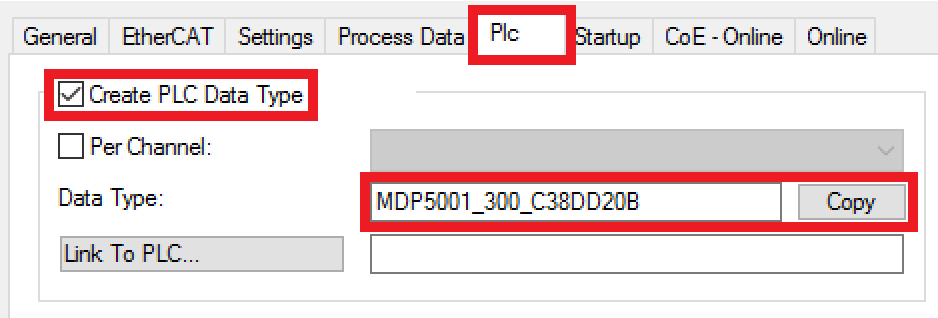


Fig. 71: Creating a PLC data type

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

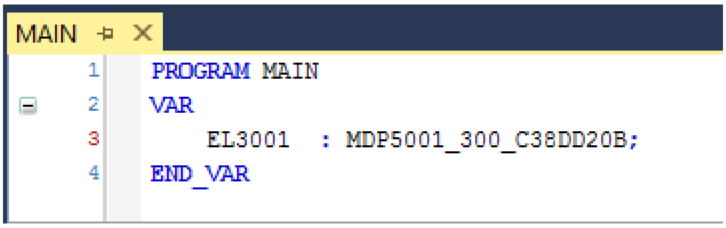


Fig. 72: Instance_of_struct

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.

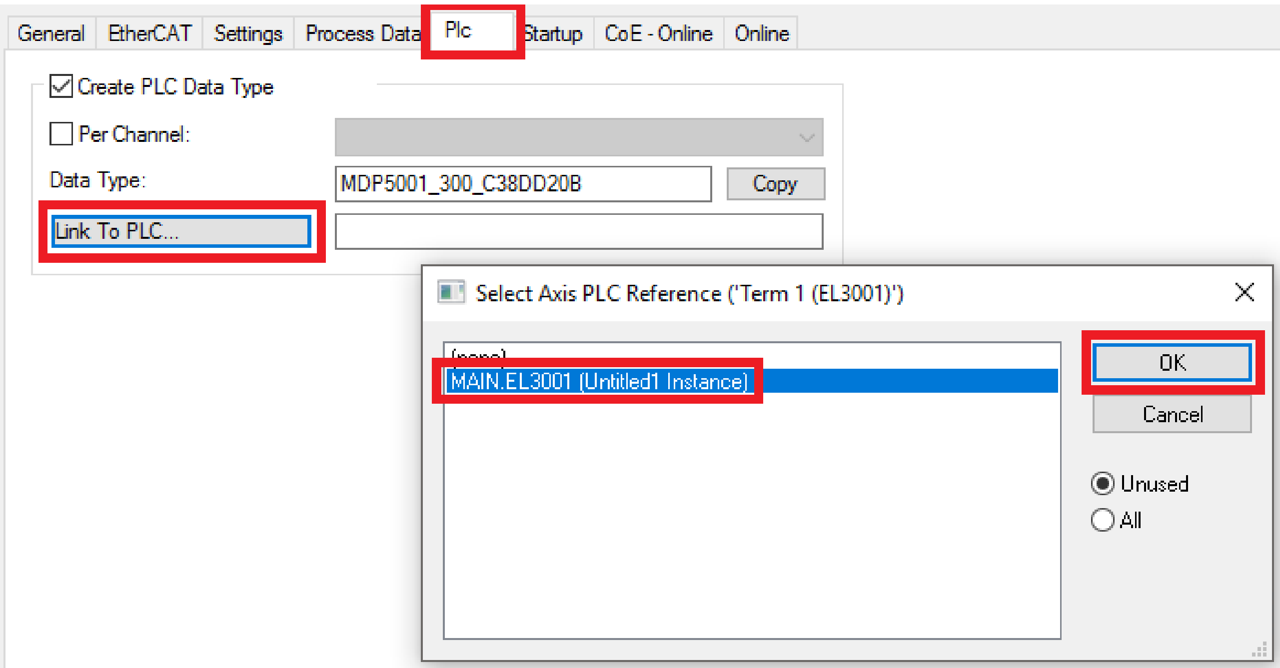


Fig. 73: Linking the structure

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

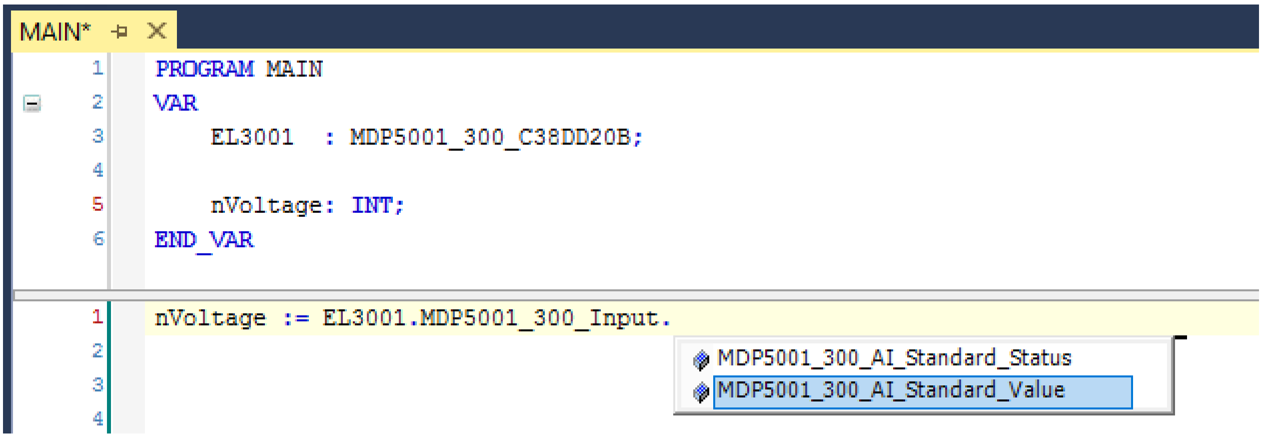
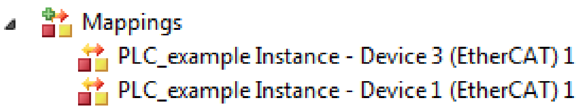


Fig. 74: Reading a variable from the structure of the process data

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:

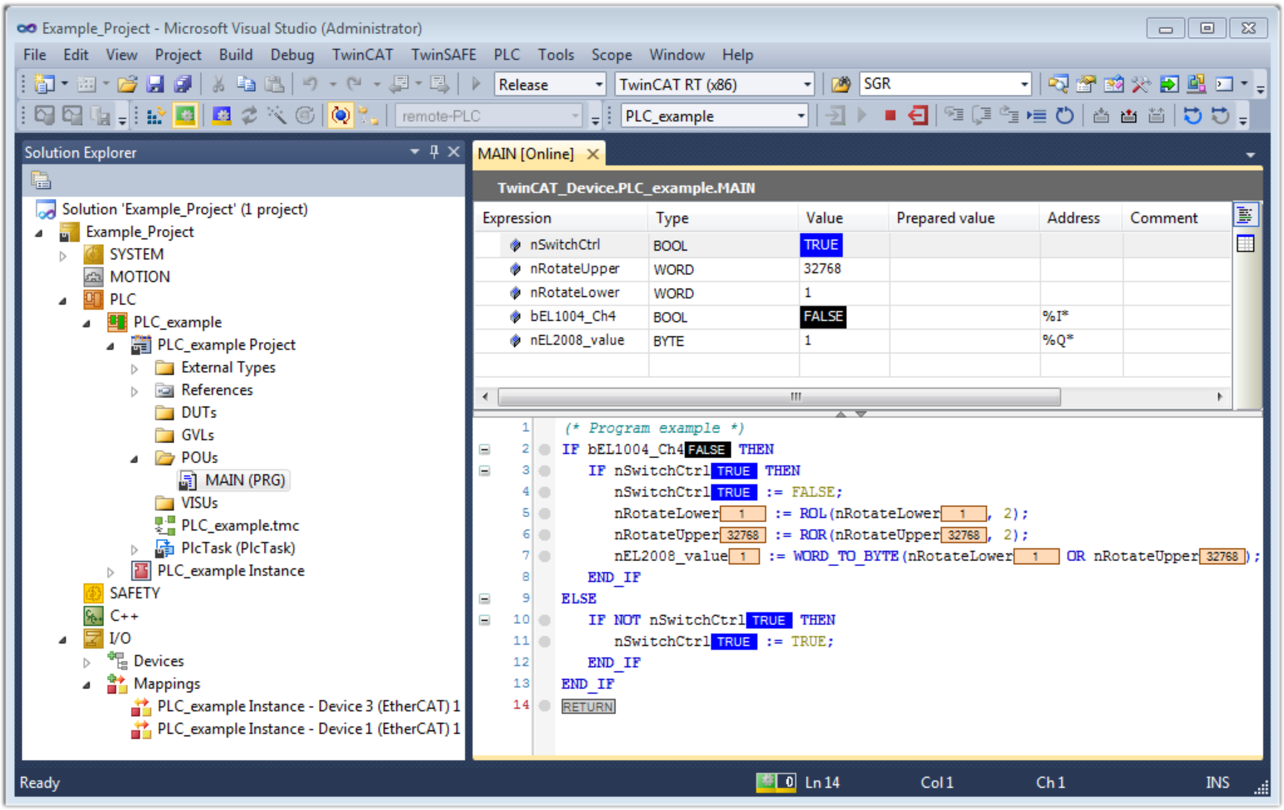


Fig. 75: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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).

6.2 TwinCAT Development Environment

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

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

Details:

- **TwinCAT 2:**
    - Connects I/O devices to tasks in a variable-oriented manner
    - Connects tasks to tasks in a variable-oriented manner
    - Supports units at the bit level
    - Supports synchronous or asynchronous relationships
    - Exchange of consistent data areas and process images
    - Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)
    - Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
    - Interconnection to all common fieldbusses
    - More...

Additional features:

- **TwinCAT 3** (eXtended Automation)**:**
    - Visual Studio® integration
    - Choice of the programming language
    - Supports object orientated extension of IEC 61131-3
    - Usage of C/C++ as programming language for real time applications
    - Connection to MATLAB®/Simulink®
    - Open interface for expandability
    - Flexible run-time environment
    - Active support of multi-core- and 64 bit operating system
    - 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.

6.2.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways.

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.

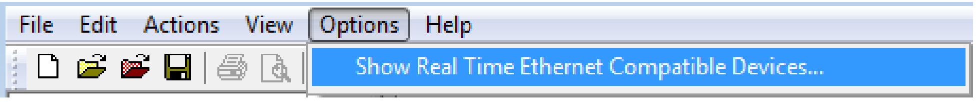


Fig. 76: System Manager “Options” (TwinCAT 2)

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

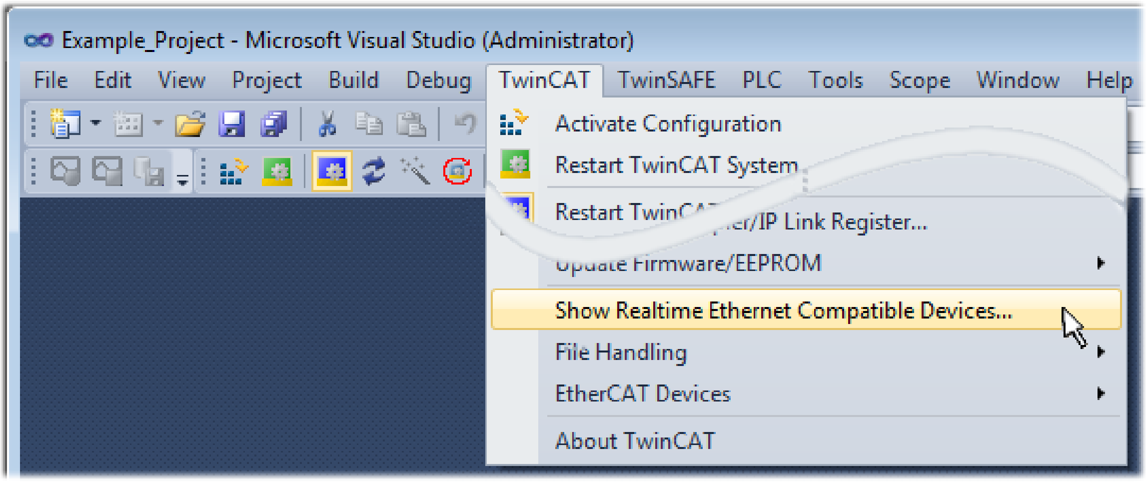


Fig. 77: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

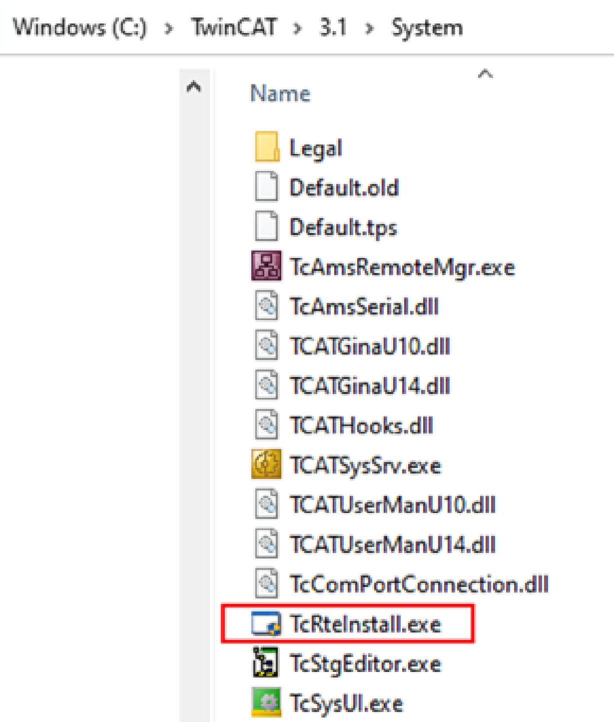


Fig. 78: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

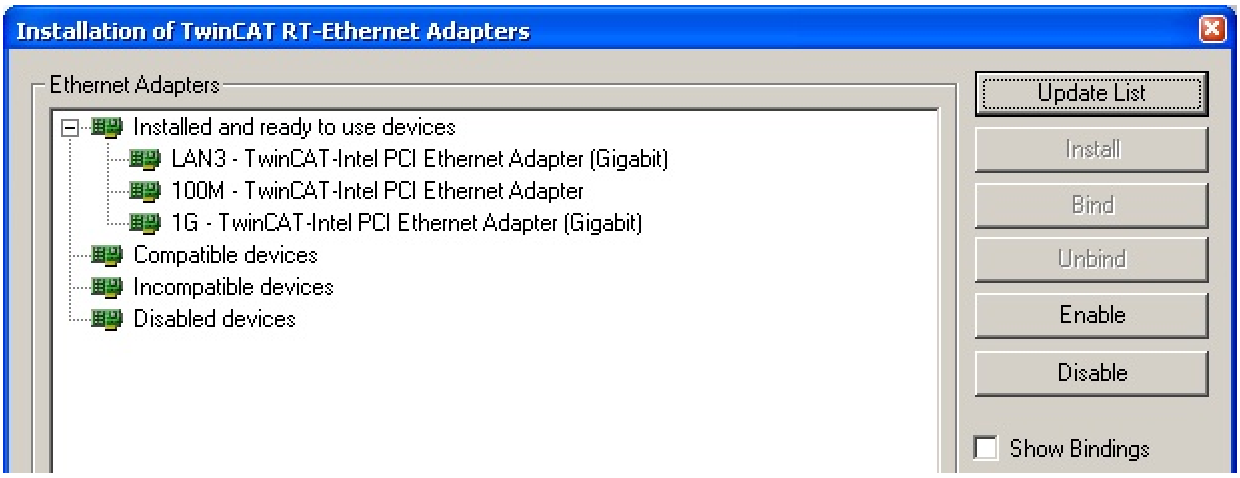


Fig. 79: Overview of network interfaces

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

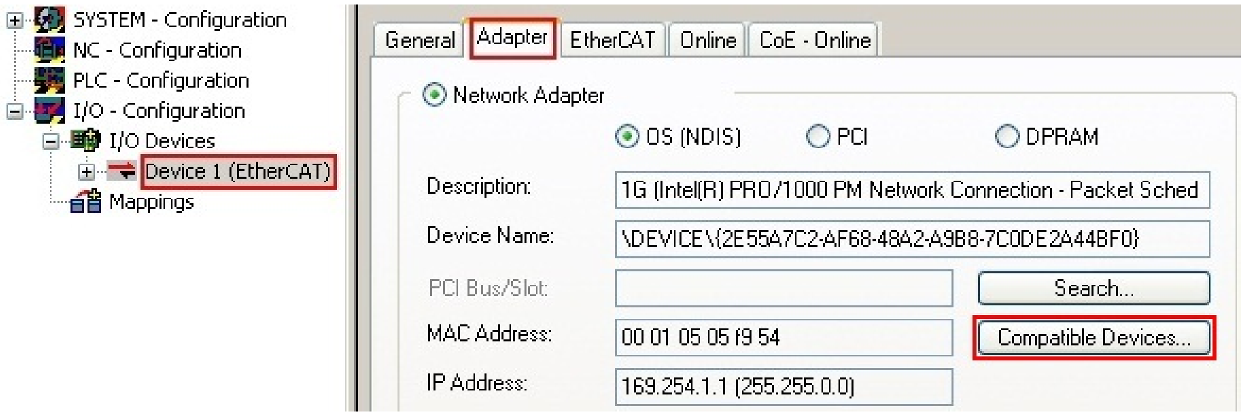
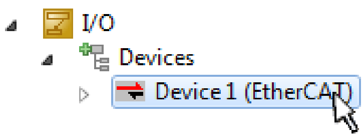


Fig. 80: EtherCAT device properties (TwinCAT 2): click on “Compatible Devices…” of tab “Adapter”

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)

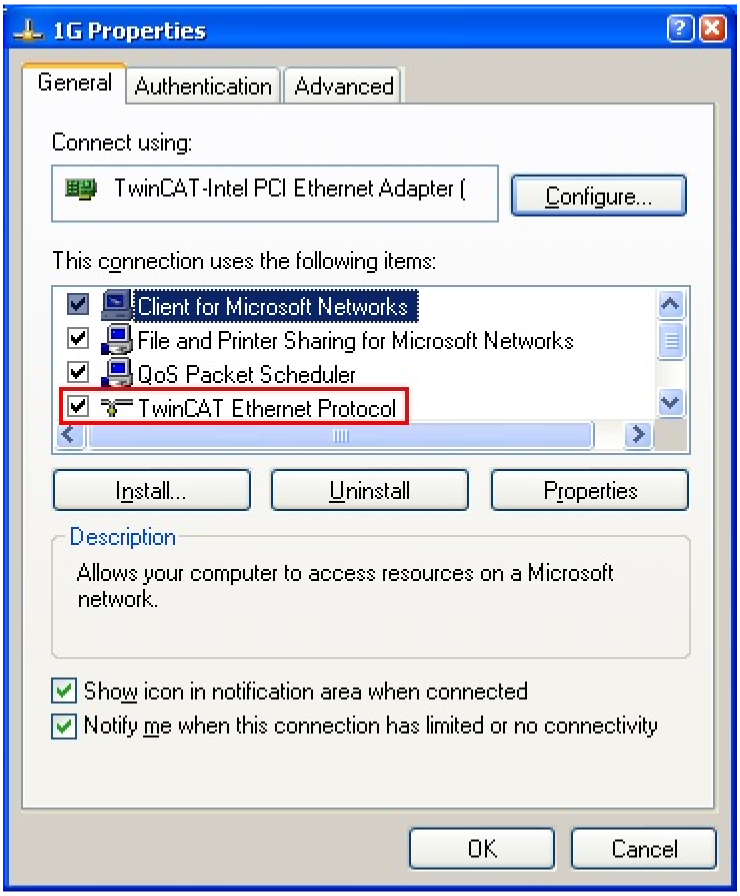


Fig. 81: Windows properties of the network interface

A correct setting of the driver could be:

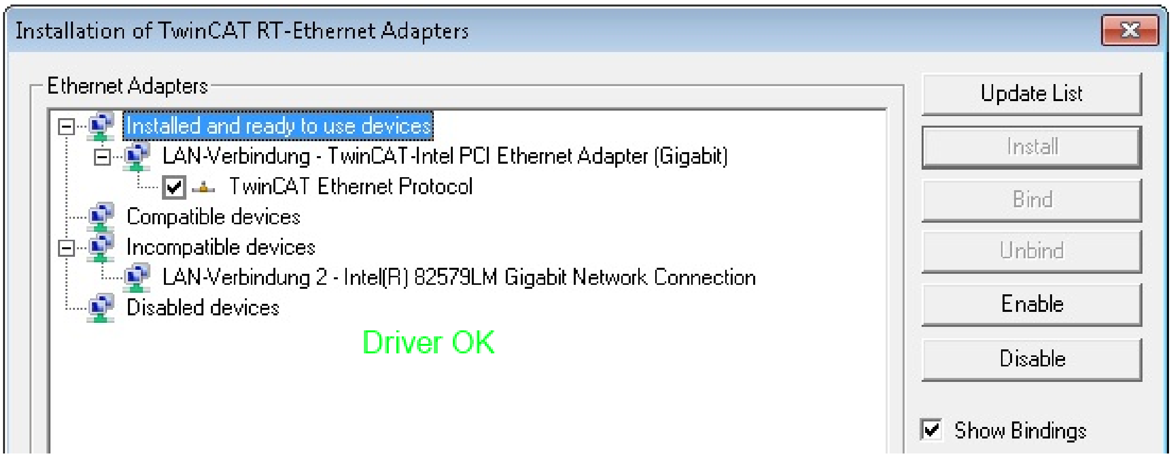


Fig. 82: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

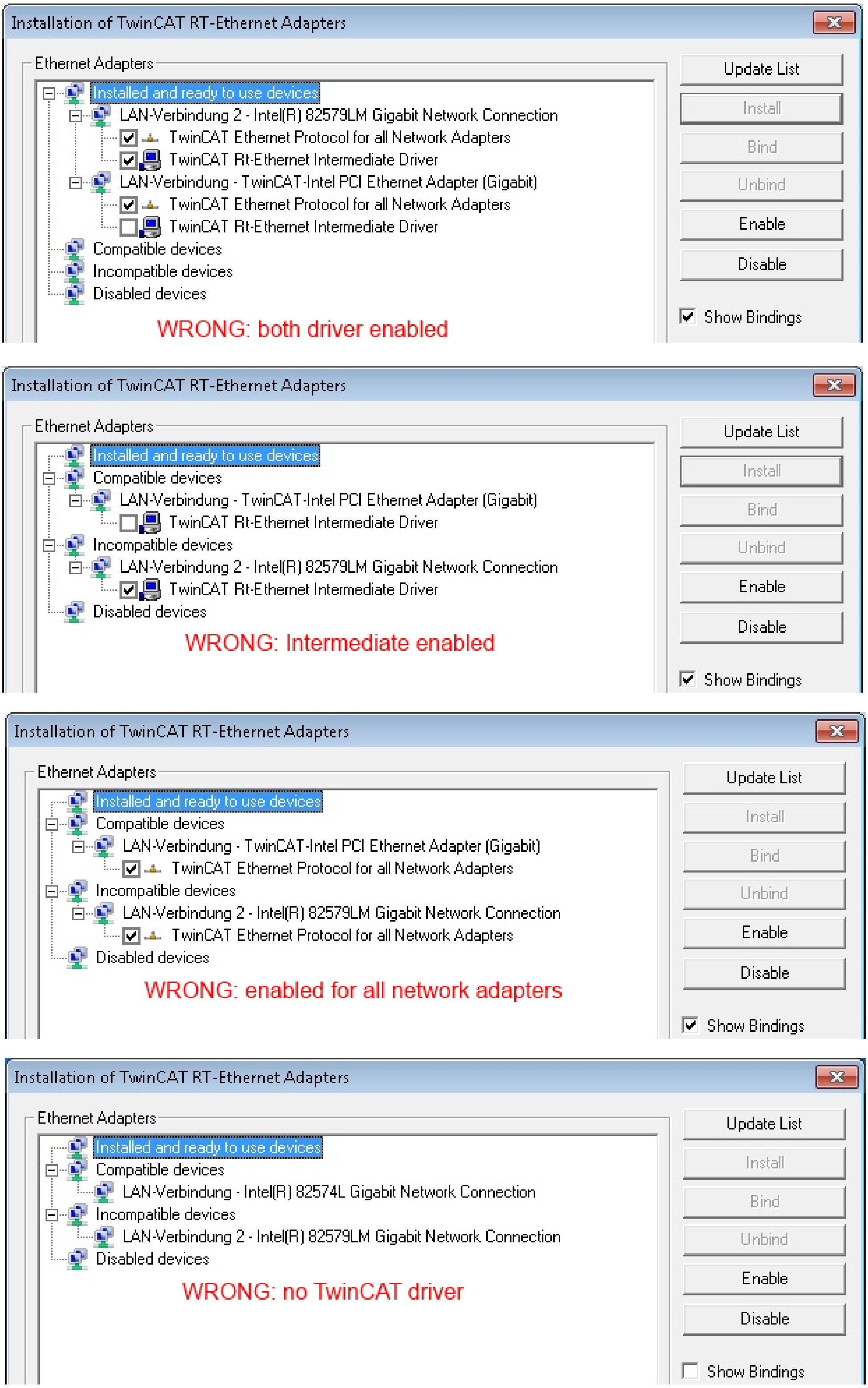


Fig. 83: Incorrect driver settings for the Ethernet port

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.

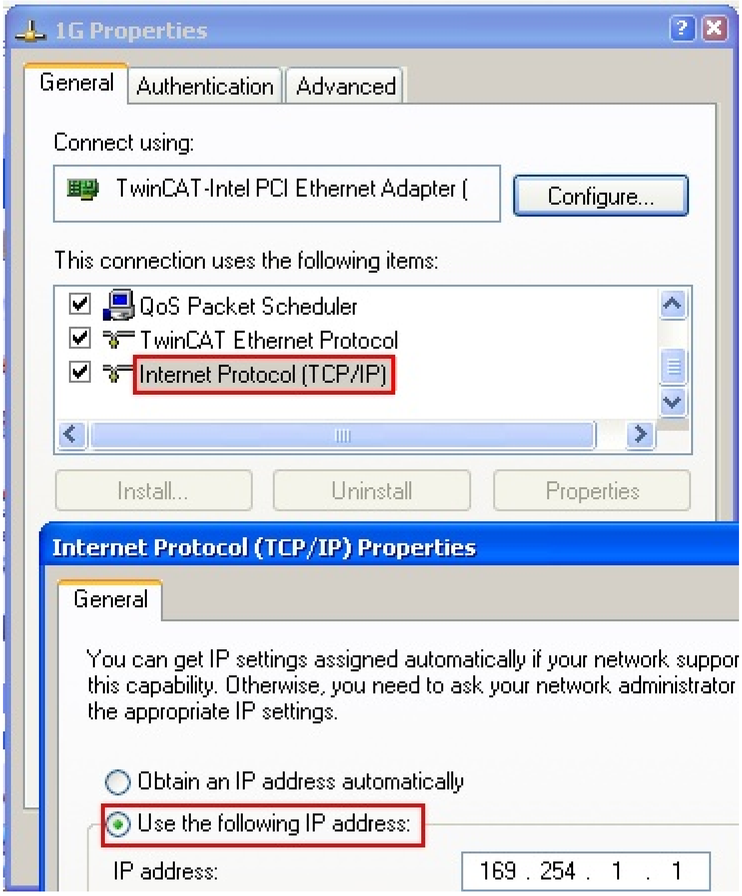


Fig. 84: TCP/IP setting for the Ethernet port

6.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

The TwinCAT EtherCAT master/System Manager needs the device description files for the devices to be used in order to 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 [▸ 97] 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"

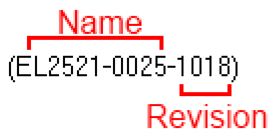


Fig. 85: Identifier structure

The order identifier consisting of name + type (here: EL2521-0025) 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 [▸ 11].

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.

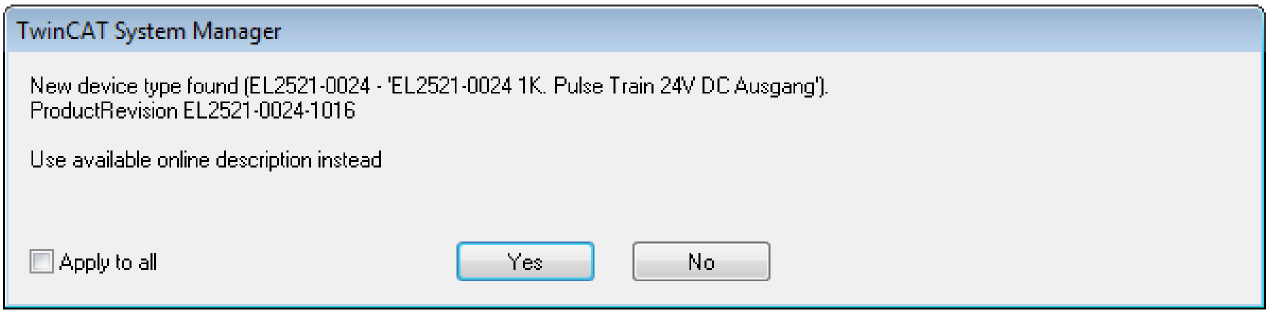


Fig. 86: OnlineDescription information window (TwinCAT 2)

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

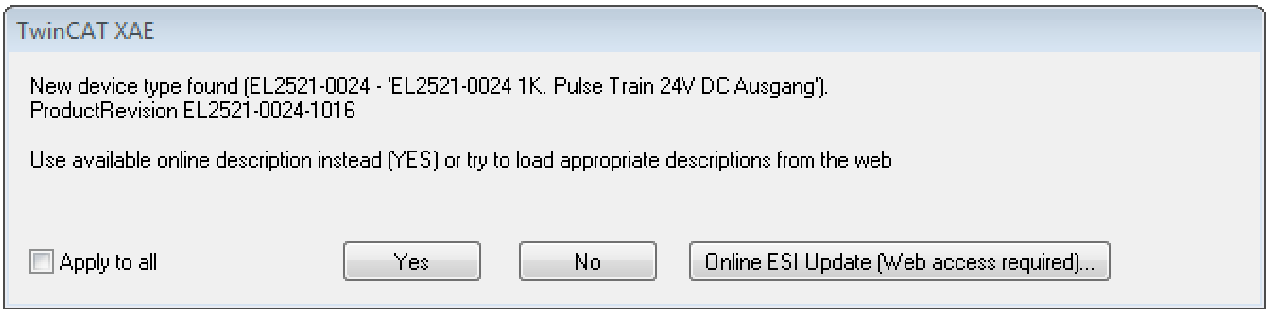


Fig. 87: Information window OnlineDescription (TwinCAT 3)

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

NOTICE

Changing the “usual” configuration through a scan

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

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

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

Refer in particular to the chapter “General notes on the use of Beckhoff EtherCAT IO components” and for manual configuration to the chapter “Offline configuration creation [▶ 98]”.

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. 88: 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)*.

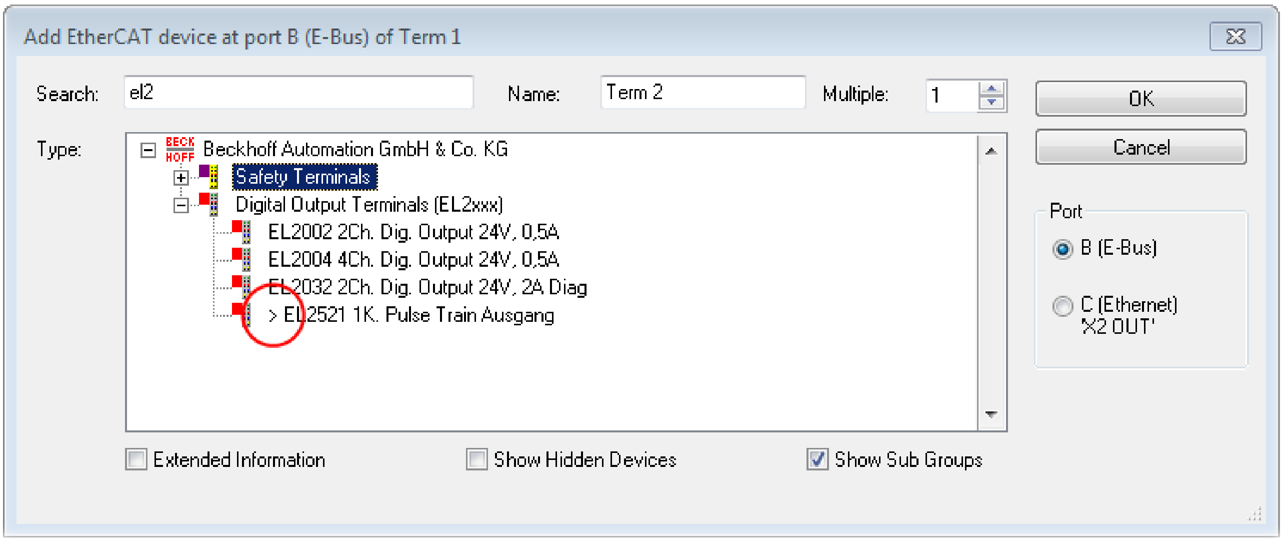


Fig. 89: Indication of an online recorded ESI of EL2521 as an example

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.

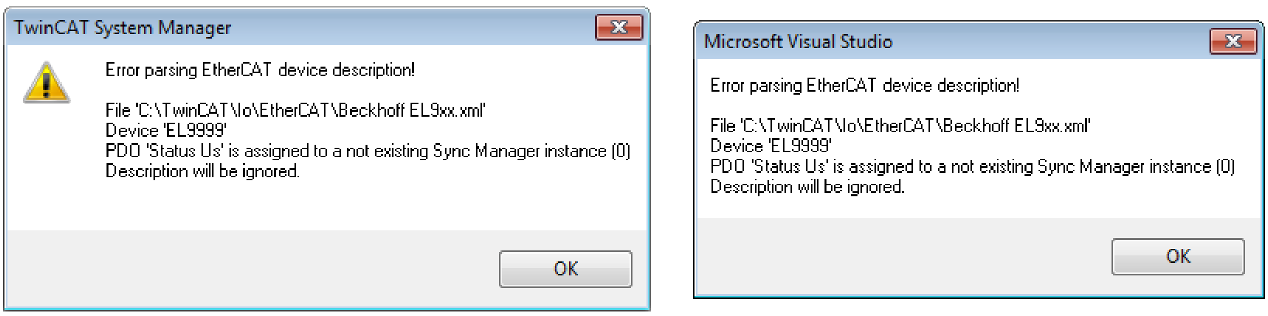


Fig. 90: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

6.2.3 TwinCAT ESI Updater

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

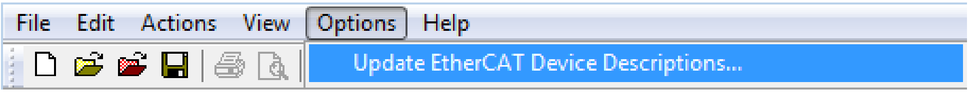


Fig. 91: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

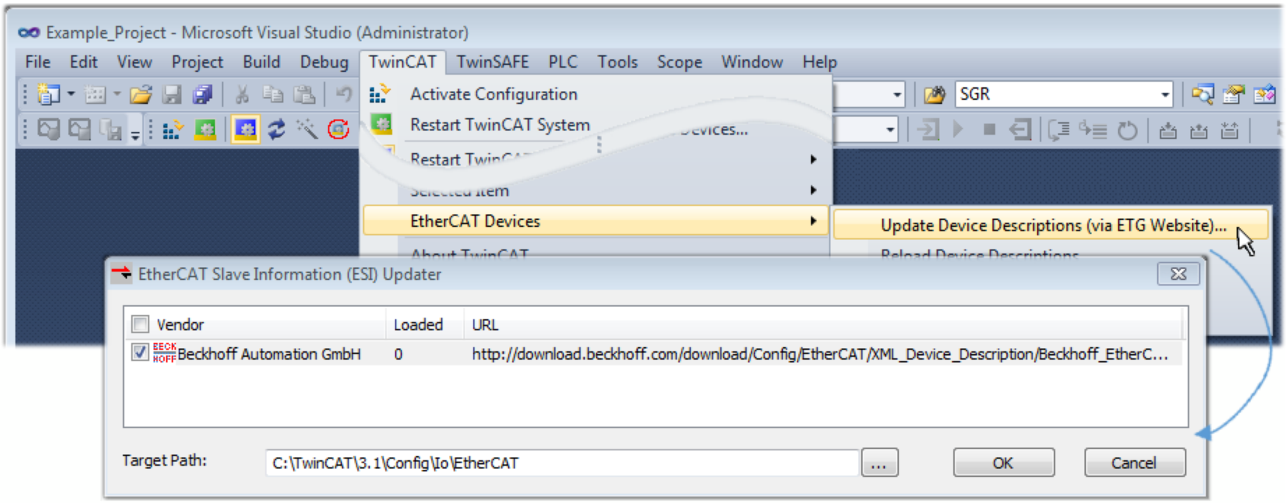


Fig. 92: Using the ESI Updater (TwinCAT 3)

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)…”.

6.2.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note “Installation of the latest ESI-XML device description” [▸ 93].

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 [▶ 103] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 104]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 107]

The scan with existing configuration [▶ 108] can also be carried out for comparison.

6.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

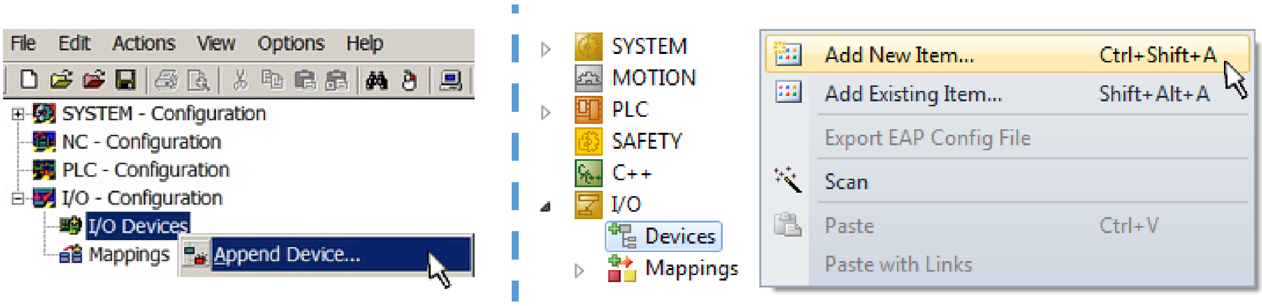


Fig. 93: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

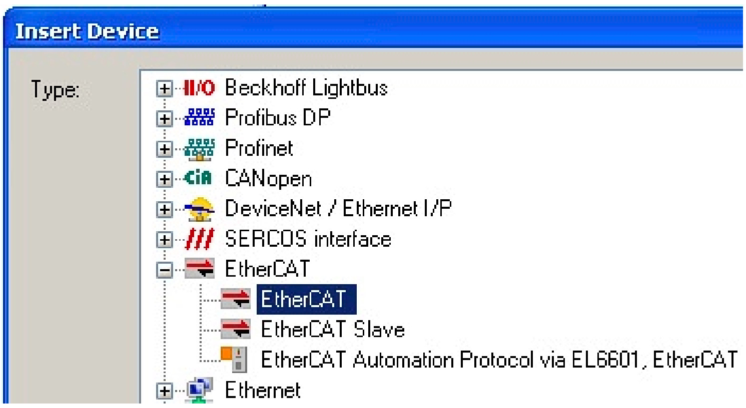


Fig. 94: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

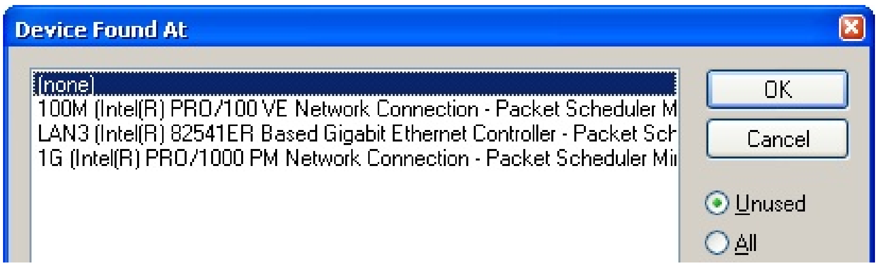


Fig. 95: Selecting the Ethernet port

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)”.

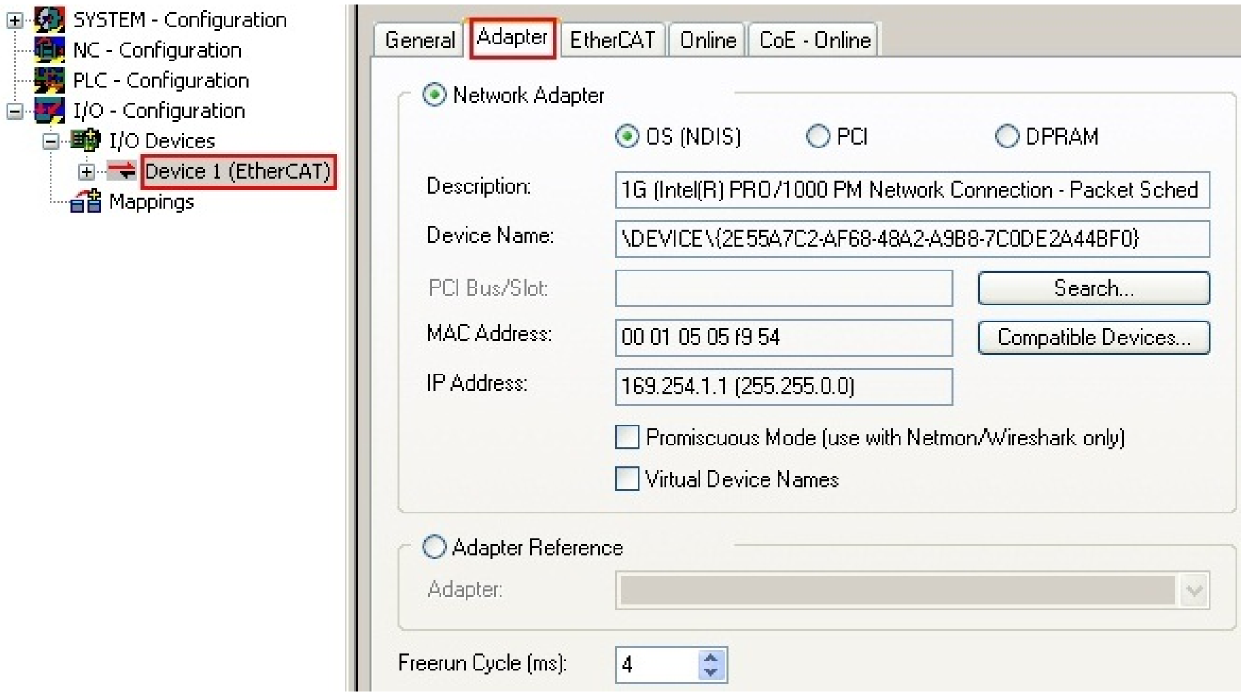
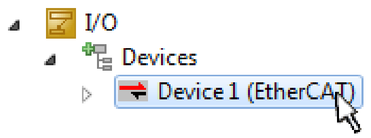


Fig. 96: EtherCAT device properties (TwinCAT 2)

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 [▶ 87].

Defining EtherCAT slaves

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

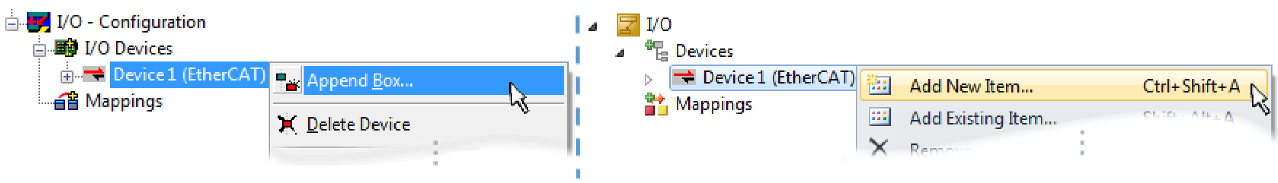


Fig. 97: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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).

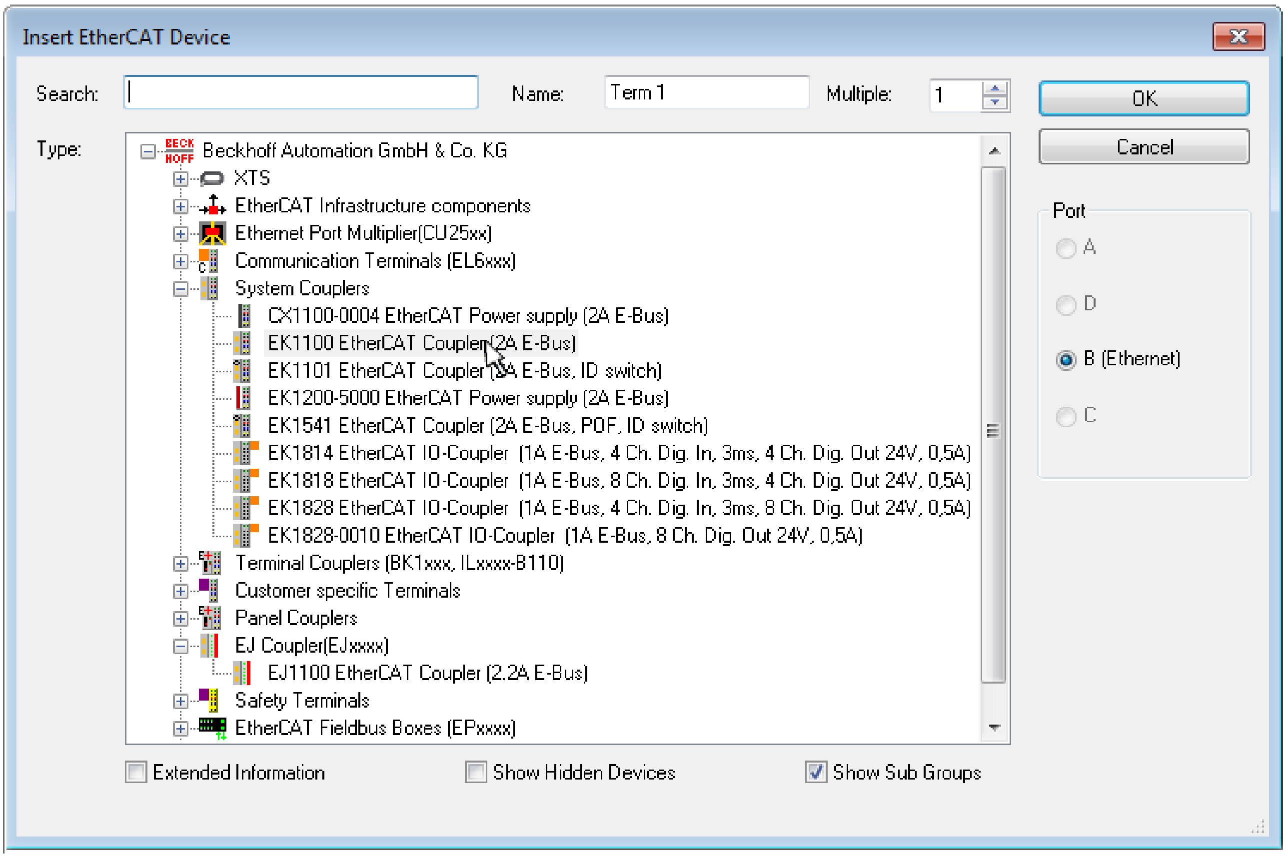


Fig. 98: Selection dialog for new EtherCAT device

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".

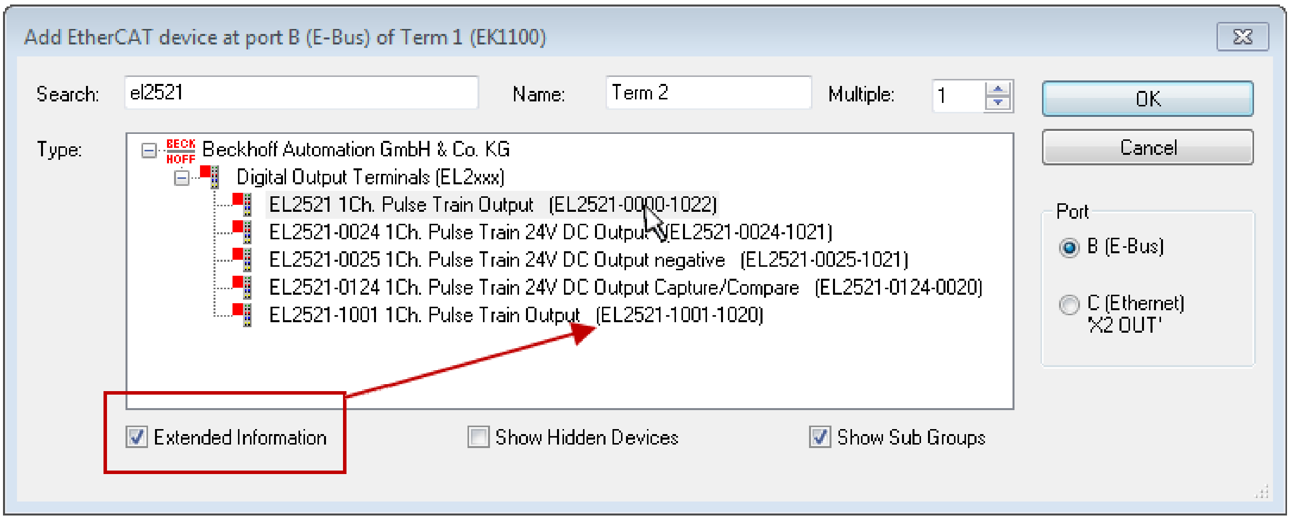


Fig. 99: Display of device revision

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".

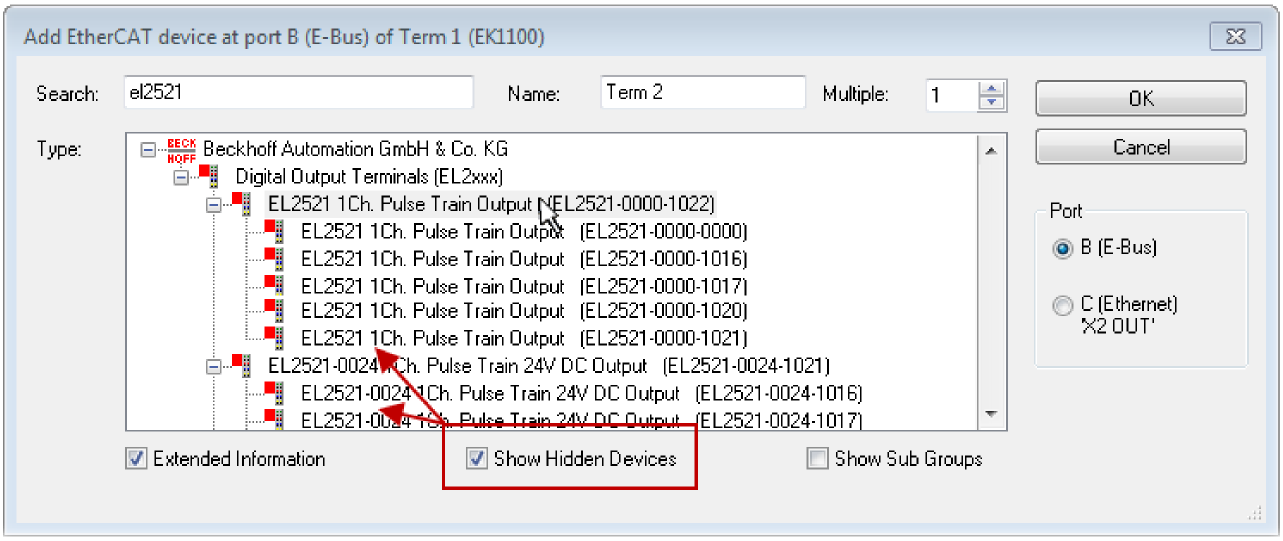


Fig. 100: Display of previous revisions

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.

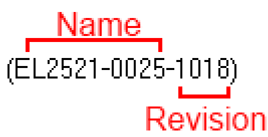


Fig. 101: Name/revision of the terminal

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, ...

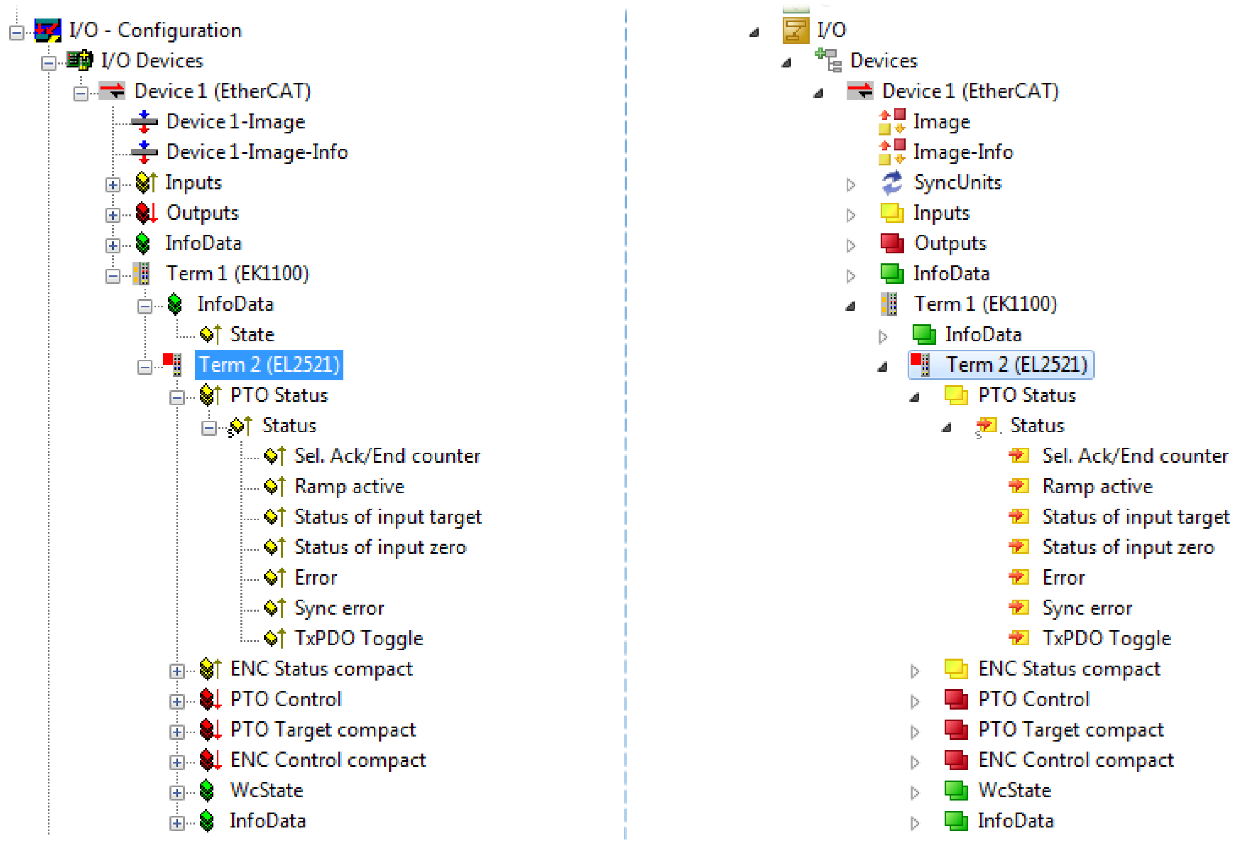


Fig. 102: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

6.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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

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

TwinCAT can be set into this mode:

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

Online scanning in Config mode

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

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

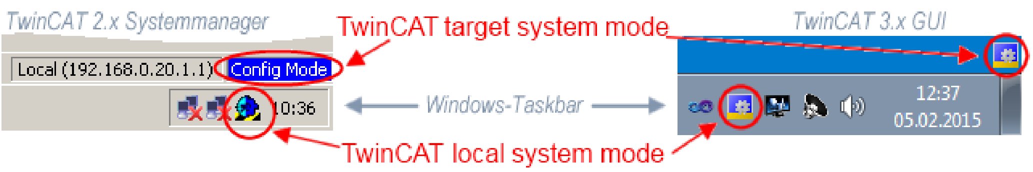


Fig. 103: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

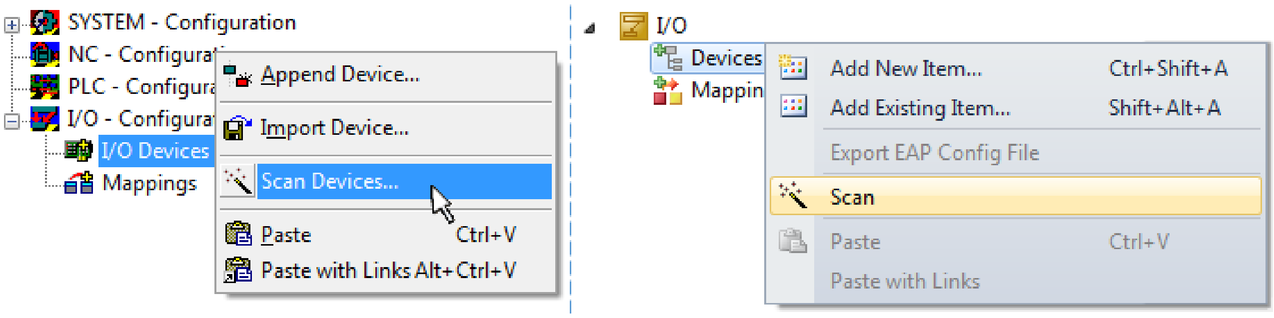


Fig. 104: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

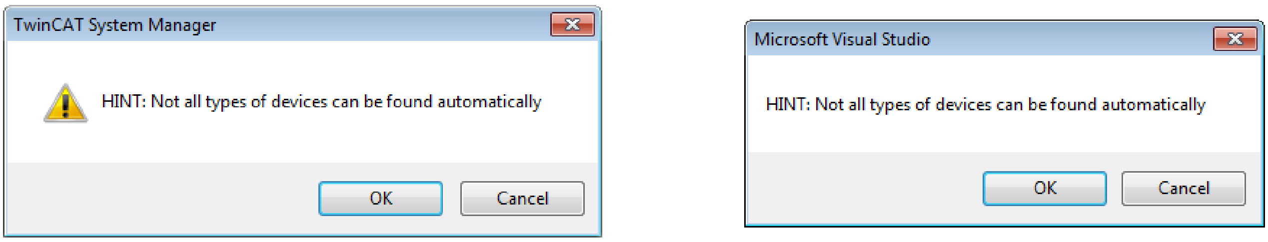


Fig. 105: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

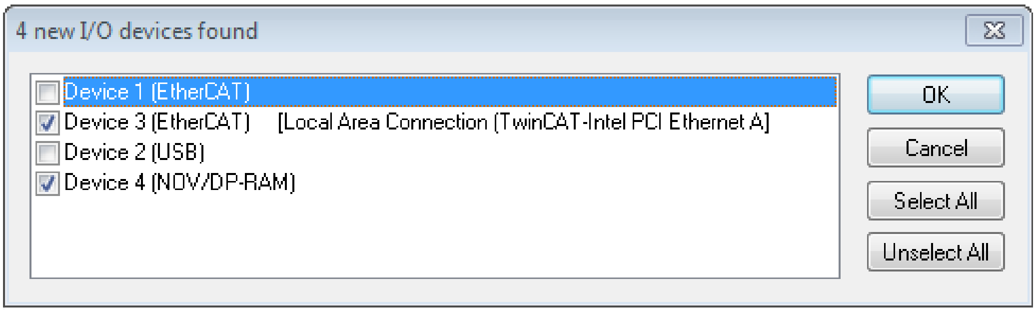


Fig. 106: Detected Ethernet devices

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 [▶ 87].

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.

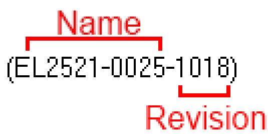


Fig. 107: Example default state

NOTICE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▶ 108] 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:

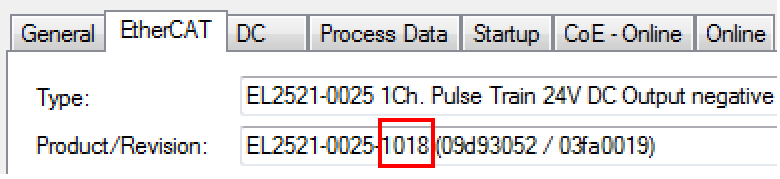


Fig. 108: Installing EthetCAT terminal with revision -1018

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 [▶ 108] 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:

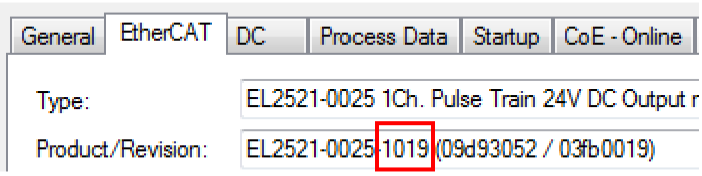


Fig. 109: Detection of EtherCAT terminal with revision -1019

This is usually not noticed by the commissioning engineers. TwinCAT cannot signal anything either, since a new configuration is essentially 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. 110: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

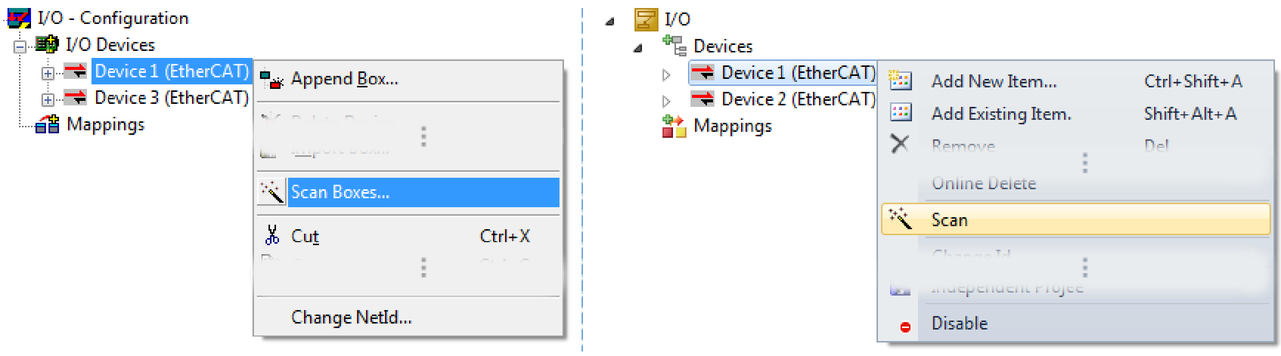


Fig. 111: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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. 112: *Scan progressexemplary by TwinCAT 2*

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

Fig. 113: 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).

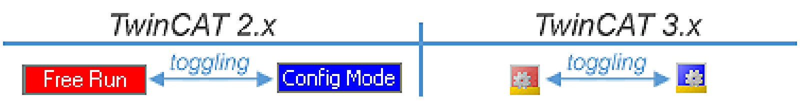


Fig. 114: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

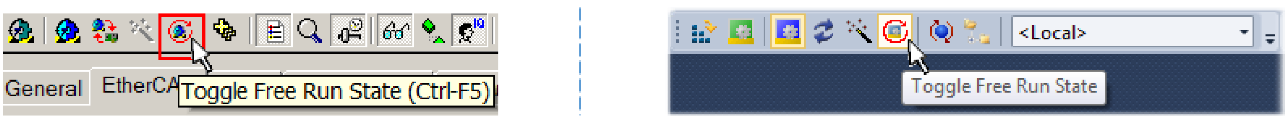


Fig. 115: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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

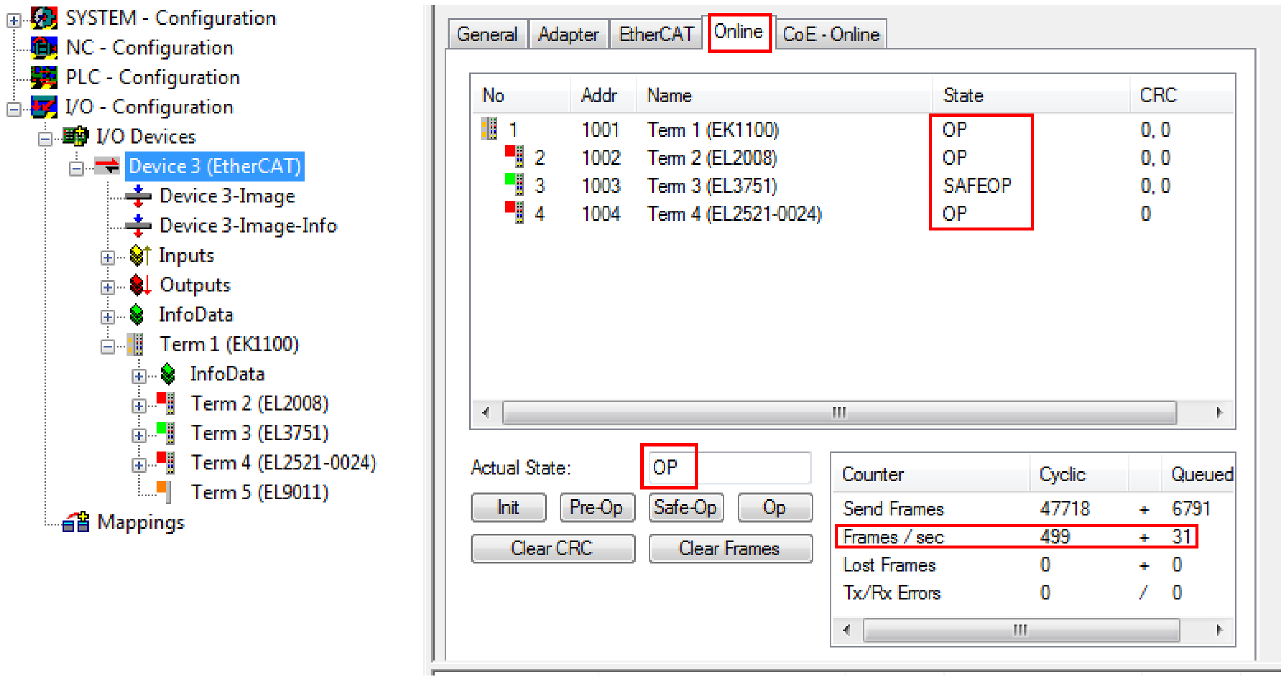


Fig. 116: Online display example

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 [▸ 98].

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.

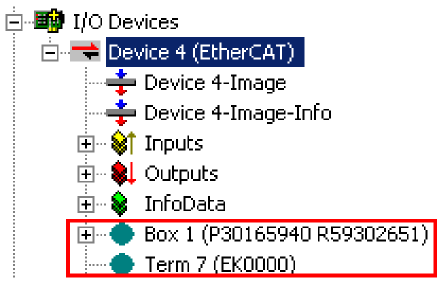


Fig. 117: Faulty identification

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

Scan over existing Configuration

NOTICE

Change of the configuration after comparison

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

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

Fig. 118: 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.

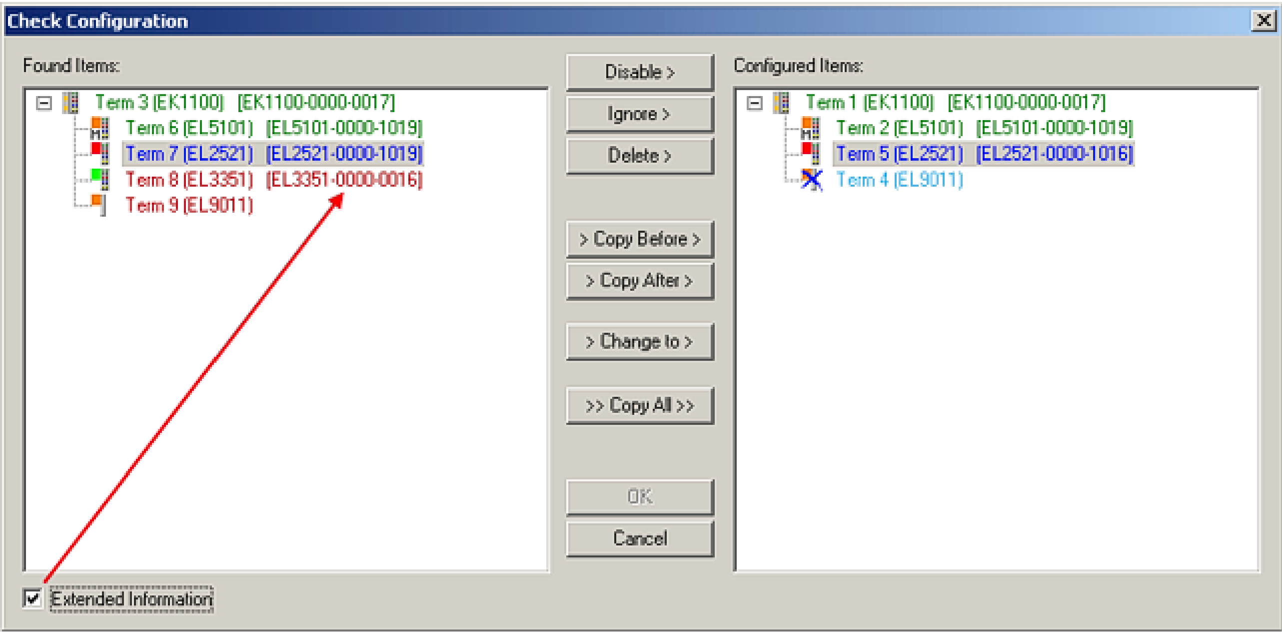


Fig. 119: Correction dialog

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.

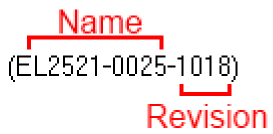


Fig. 120: Name/revision of the terminal

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, ...

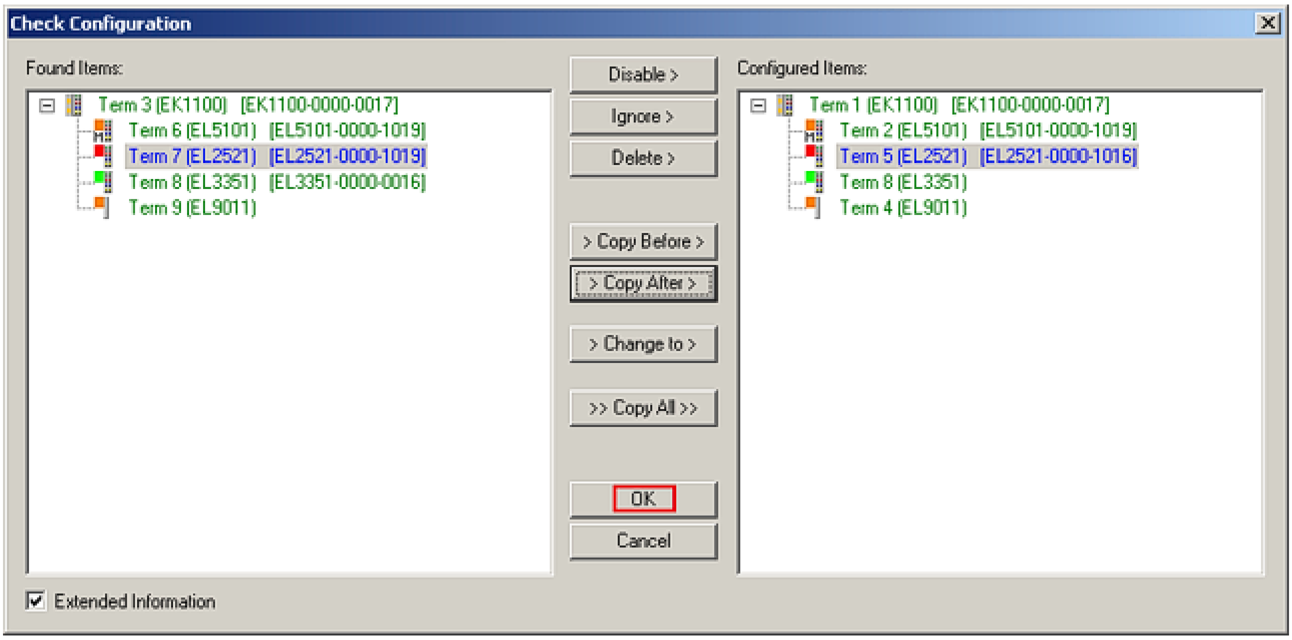


Fig. 121: Correction dialog with modifications

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.

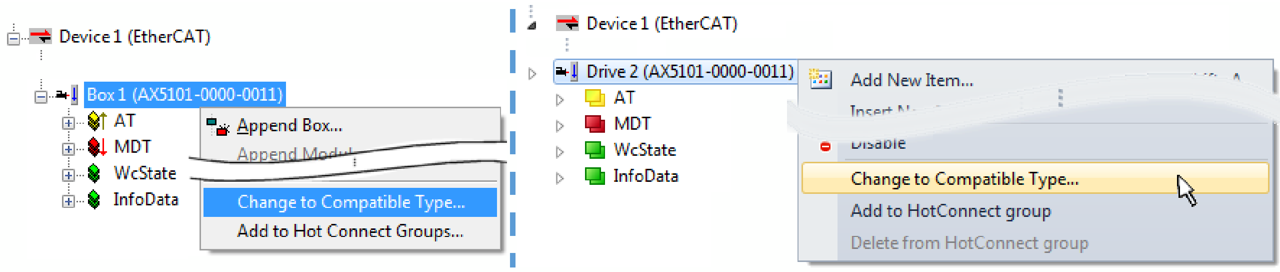


Fig. 122: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

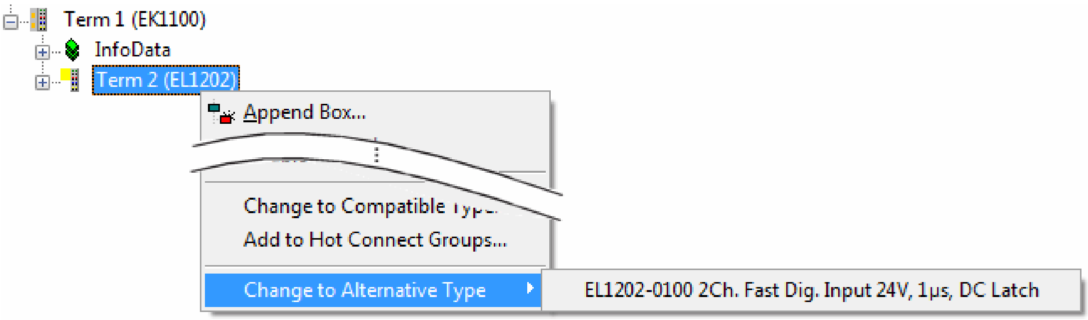


Fig. 123: TwinCAT 2 Dialog Change to Alternative Type

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).

6.2.7 EtherCAT subscriber configuration

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

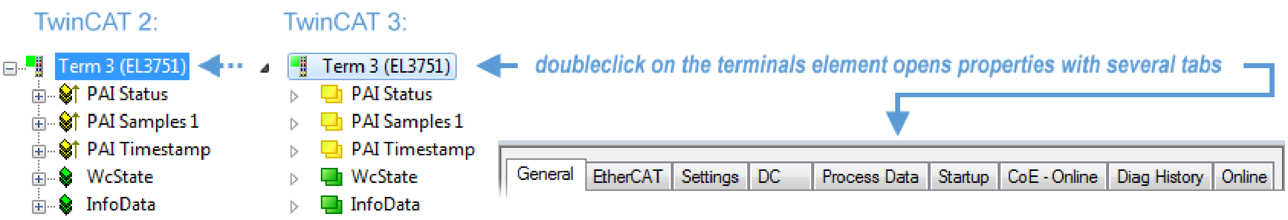


Fig. 124: Branch element as terminal EL3751

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

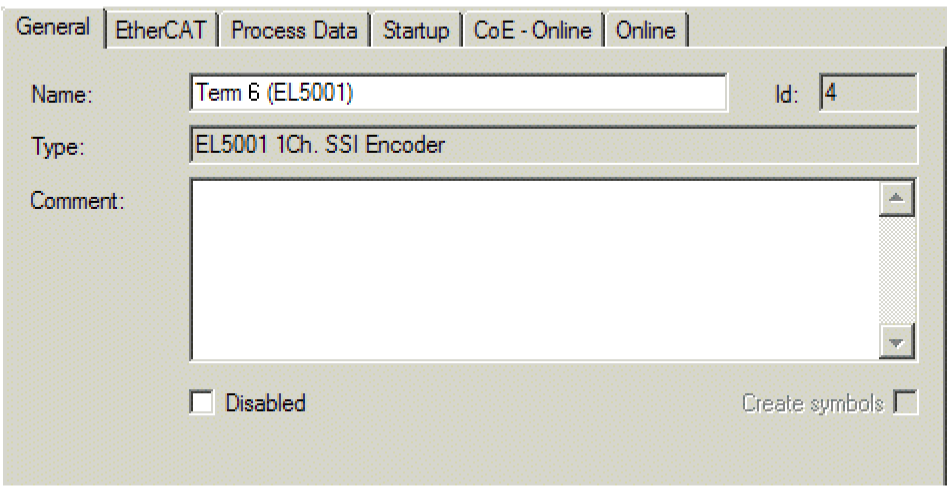


Fig. 125: “General” tab

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

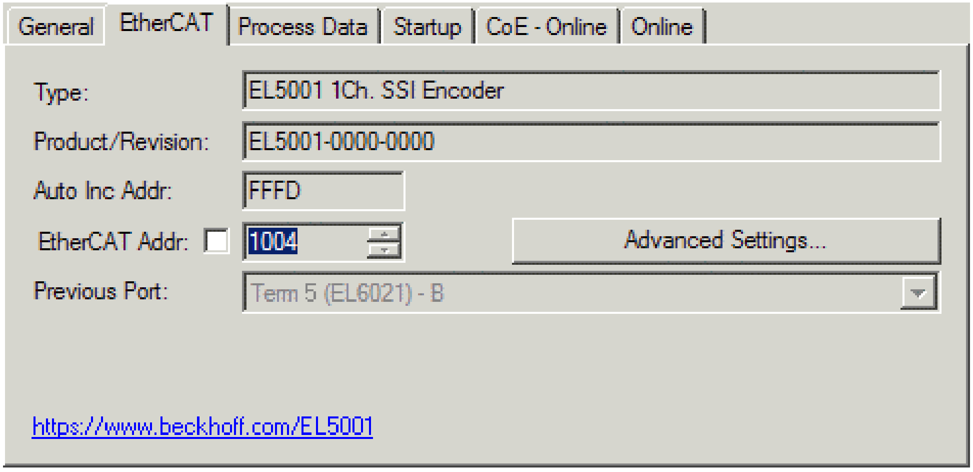


Fig. 126: “EtherCAT” tab

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.

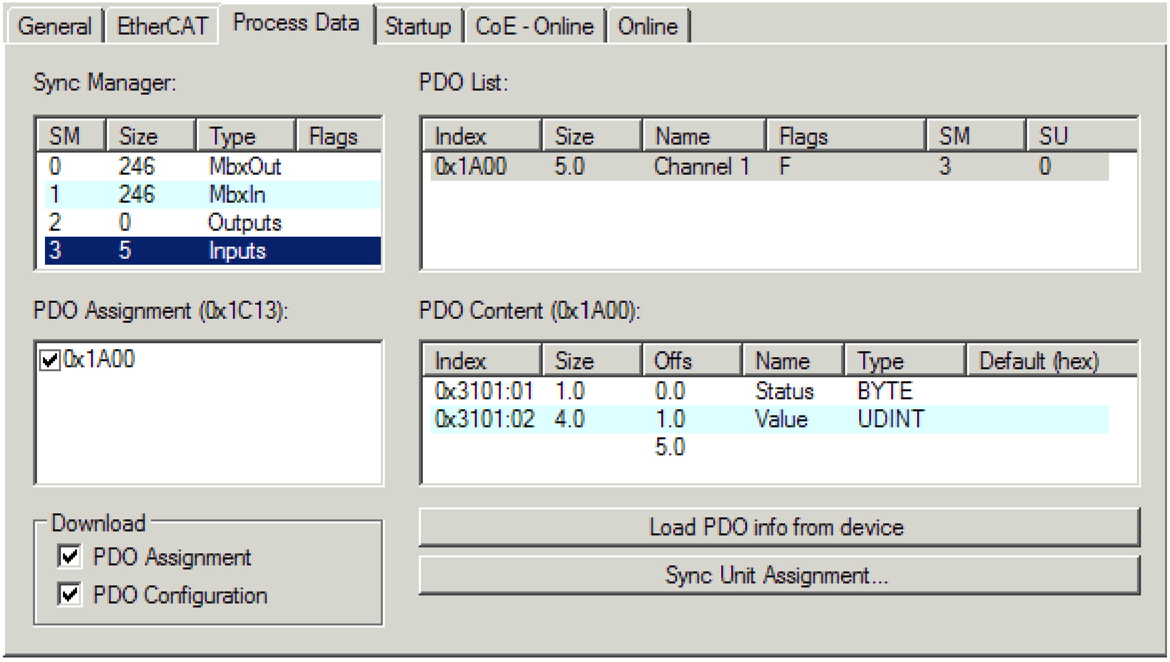


Fig. 127: “Process Data” tab

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”).

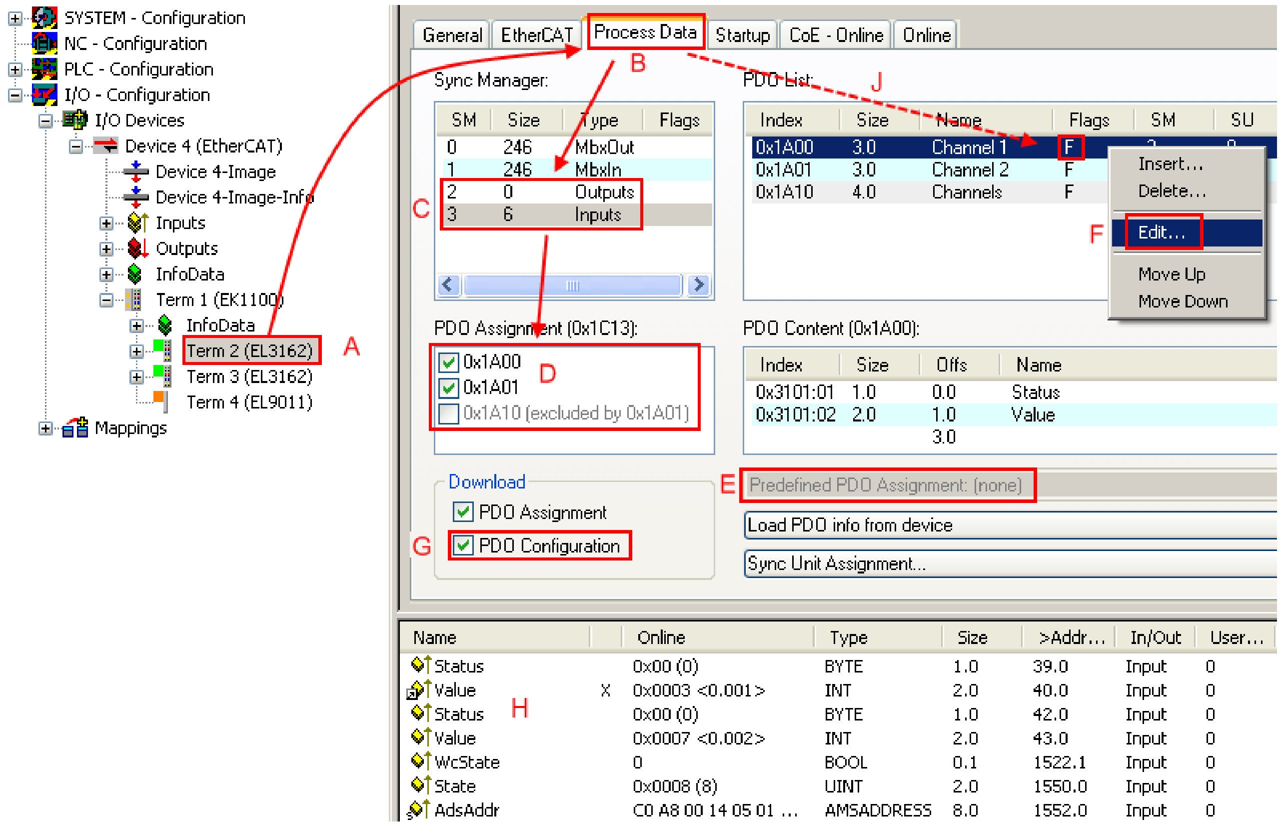


Fig. 128: Configuring the process data

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 [▸ 119] 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.

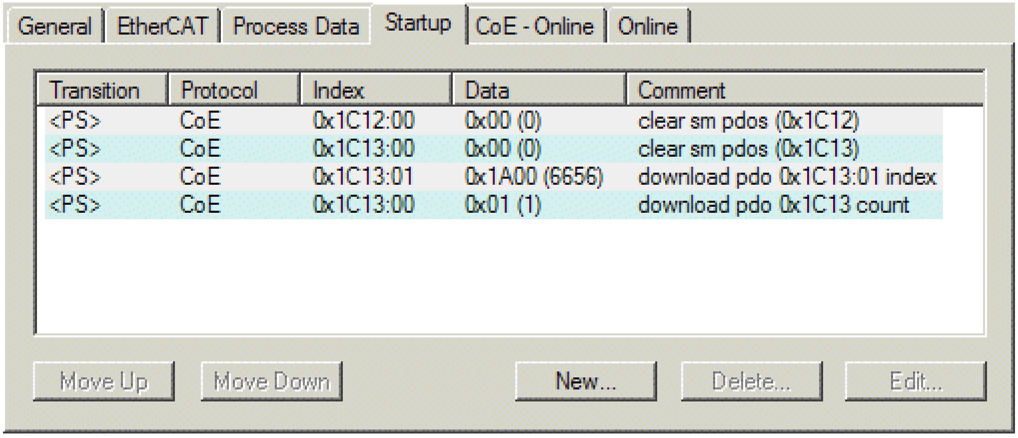


Fig. 129: “Startup” tab

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.

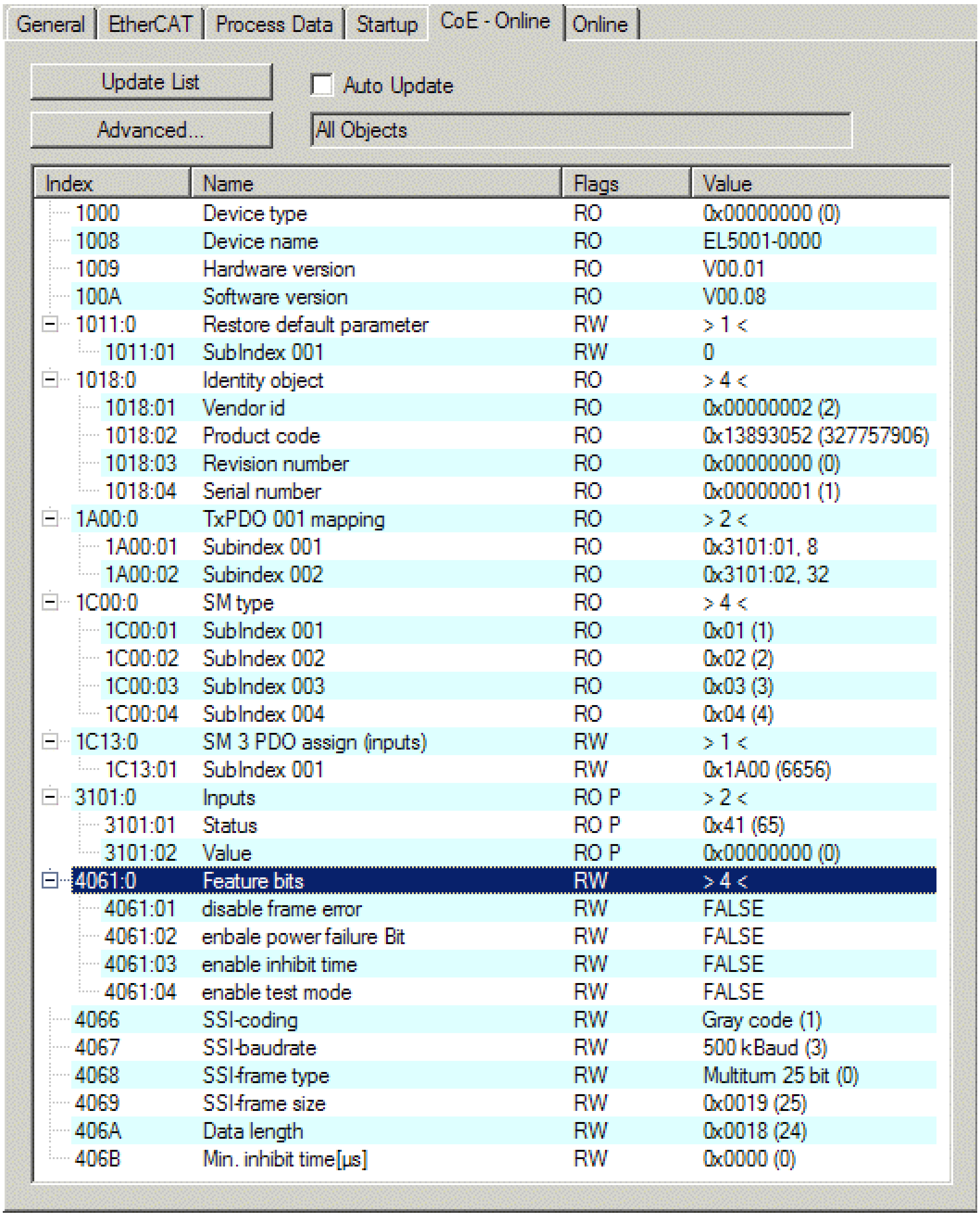


Fig. 130: “CoE - Online” tab

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.

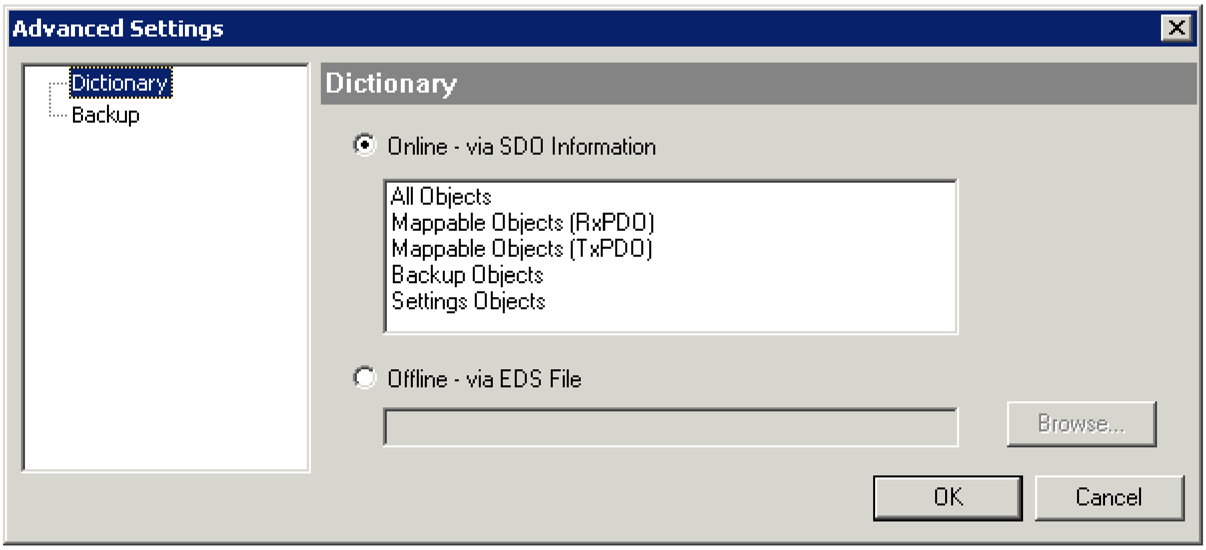


Fig. 131: Dialog “Advanced settings”

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

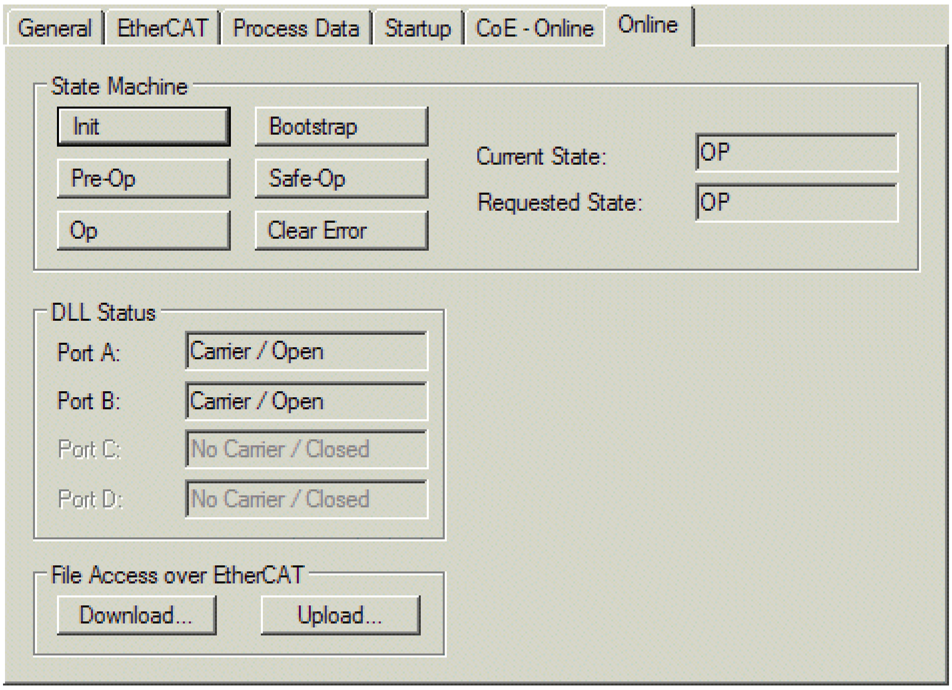


Fig. 132: “Online” tab

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)

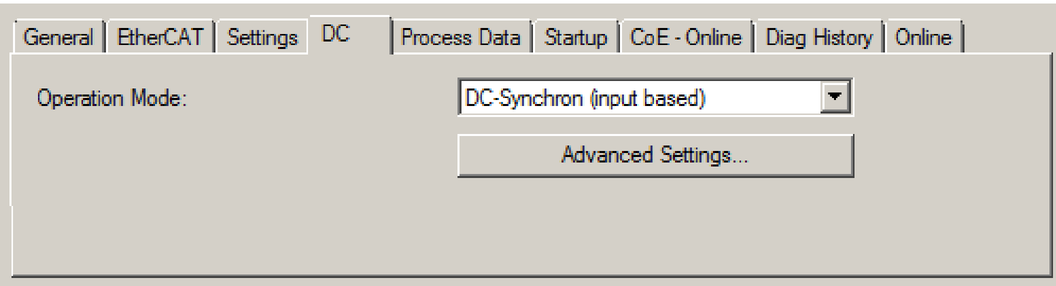


Fig. 133: “DC” tab (Distributed Clocks)

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

6.2.7.1 Detailed description of Process Data tab

Sync Manager

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

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

PDO Assignment

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

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

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

Activation of PDO assignment

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

a) the EtherCAT slave has to run through the PS status transition cycle (from pre-operational to safe-operational) once (see Online tab [▸ 117]),

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 [▶ 114] 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.

6.2.8 Import/Export of EtherCAT devices with SCI and XTI

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

6.2.8.1 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.

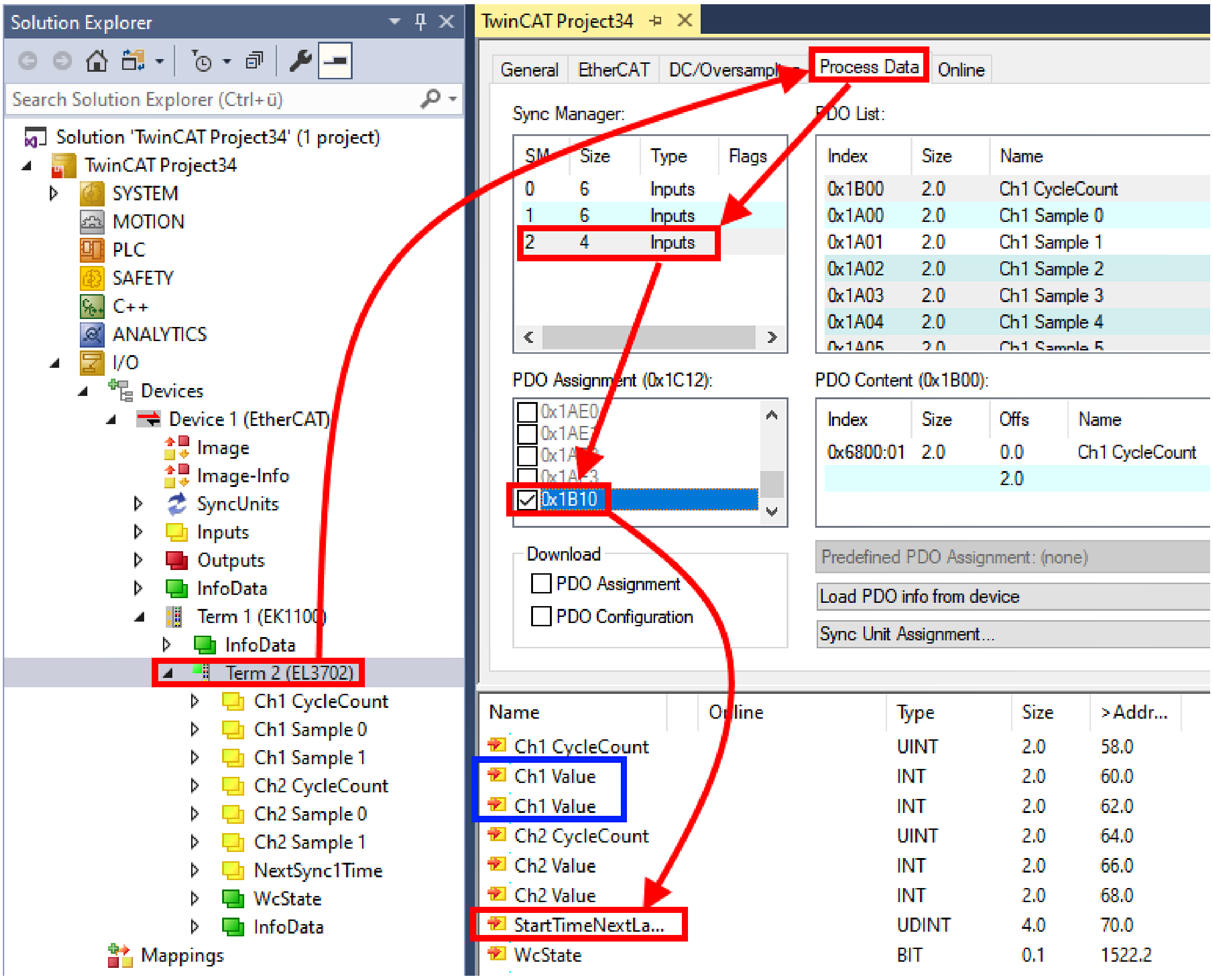
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

- 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.

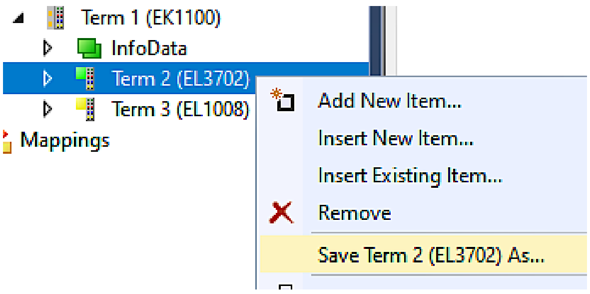
An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):



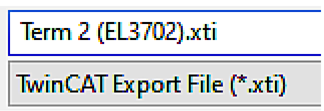
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

6.2.8.2 Procedure within TwinCAT with xti files

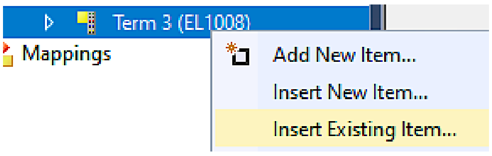
Each IO device can be exported/saved individually:



The xti file can be stored:



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



6.2.8.3 Procedure within and outside TwinCAT with sci file

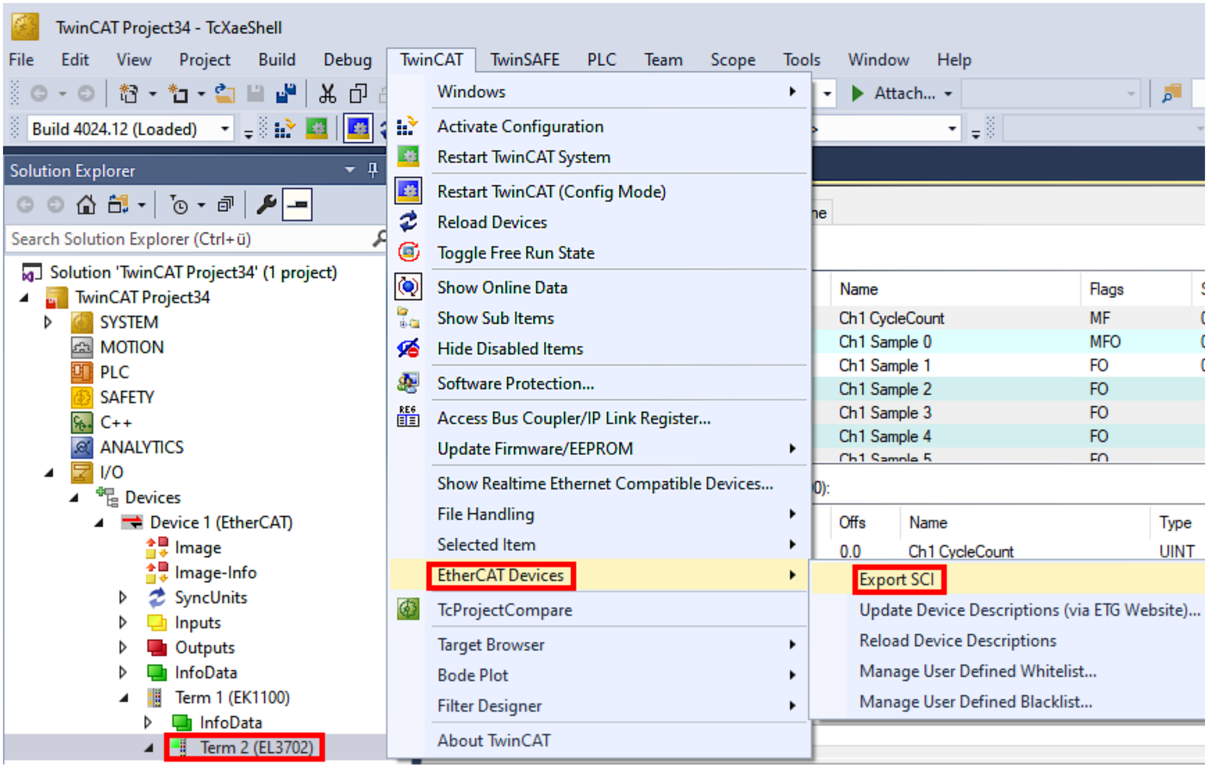
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

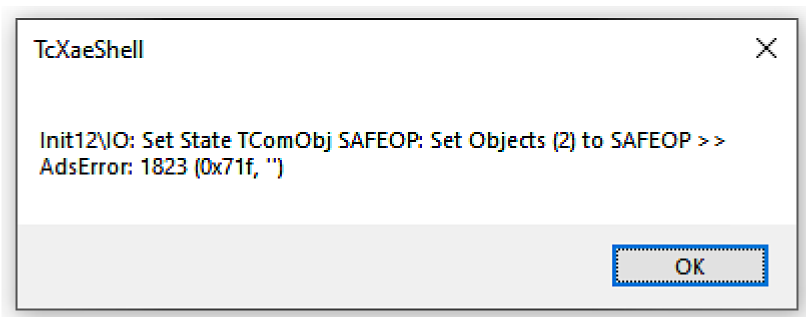
The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

Export:

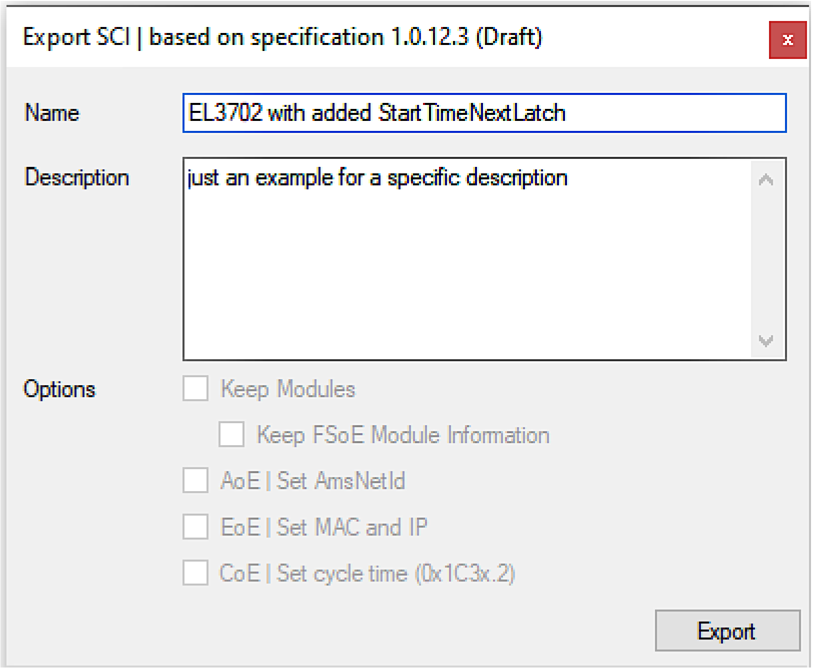
- select a single device via the menu (multiple selection is also possible):
  TwinCAT → EtherCAT Devices → Export SCI.



- 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:



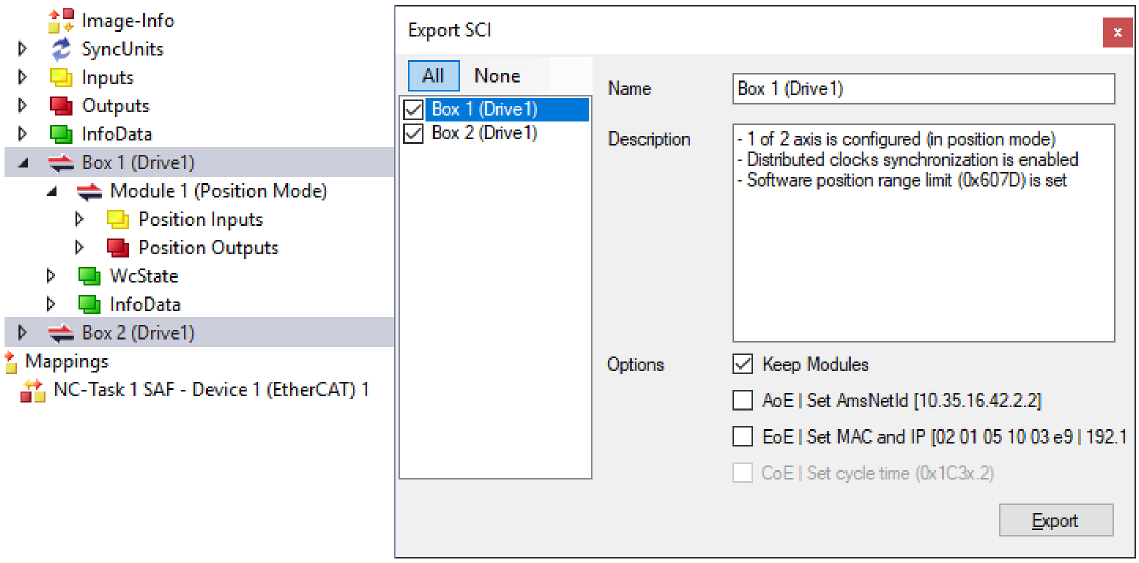
- A description may also be provided:



- Explanation of the dialog box:

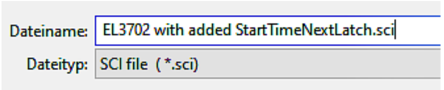
Name		Name of the SCI, assigned by the user.
Description		Description of the slave configuration for the use case, assigned by the user.
Options	Keep modules	If a slave supports modules/slots, the user can decide whether these are to be exported or whether the module and device data are to be combined 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*):

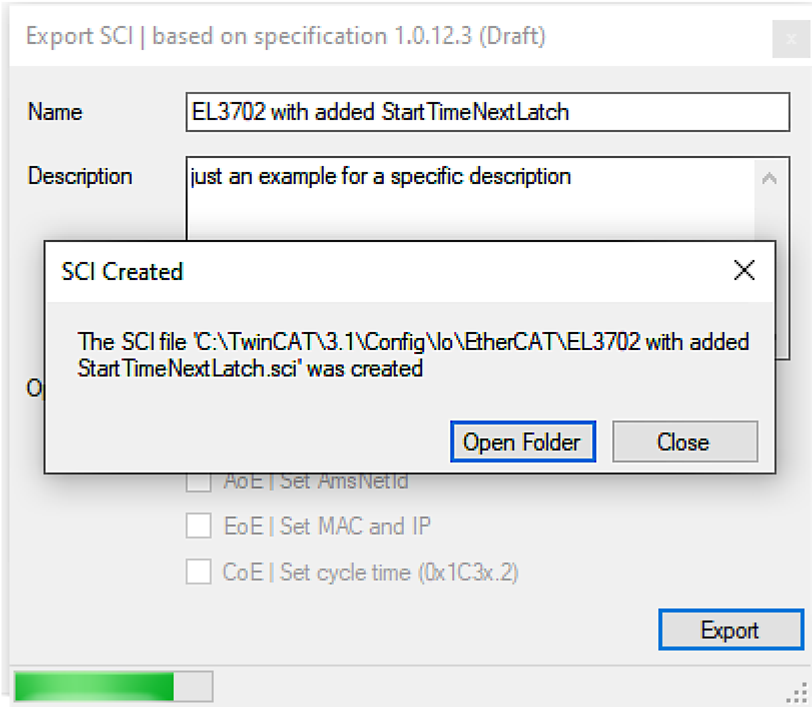


- Selection of the slaves to be exported:
    - All:
      All slaves are selected for export.

- None:
  All slaves are deselected.
- The sci file can be saved locally:

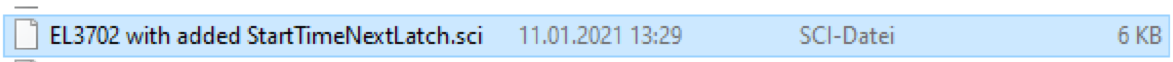


- The export takes place:

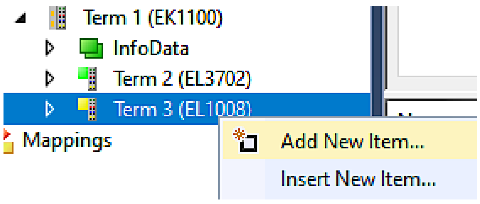


Import

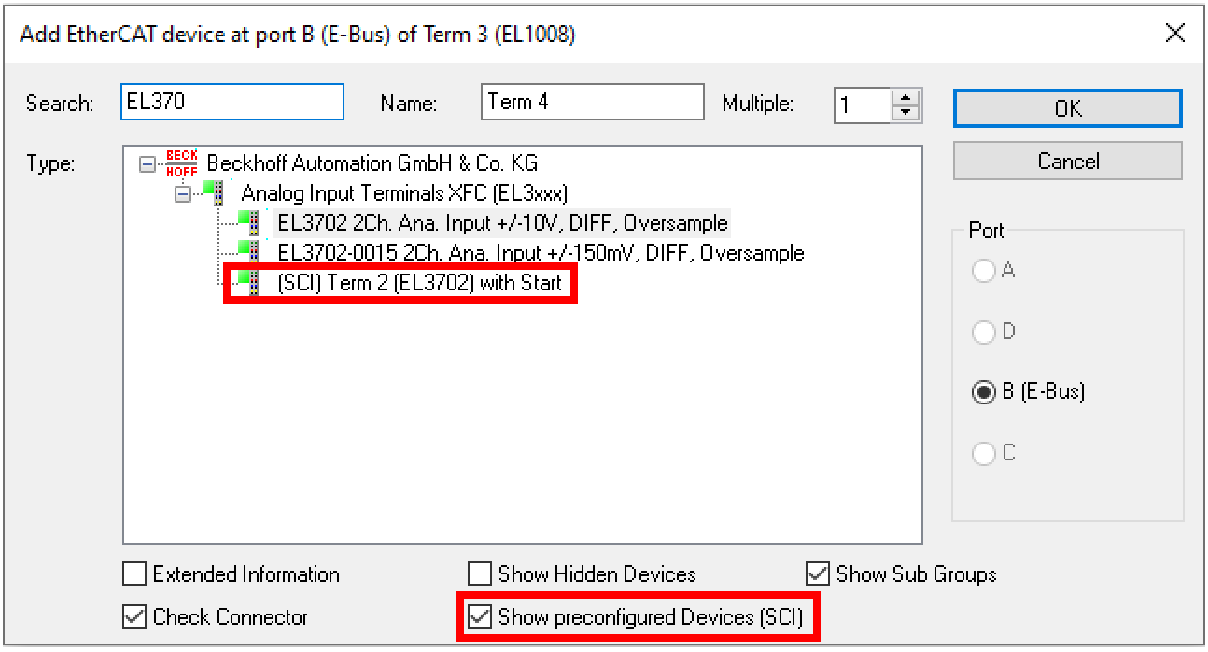
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under:
  C:\TwinCAT\3.1\Config\Io\EtherCAT



- Open the selection dialog:

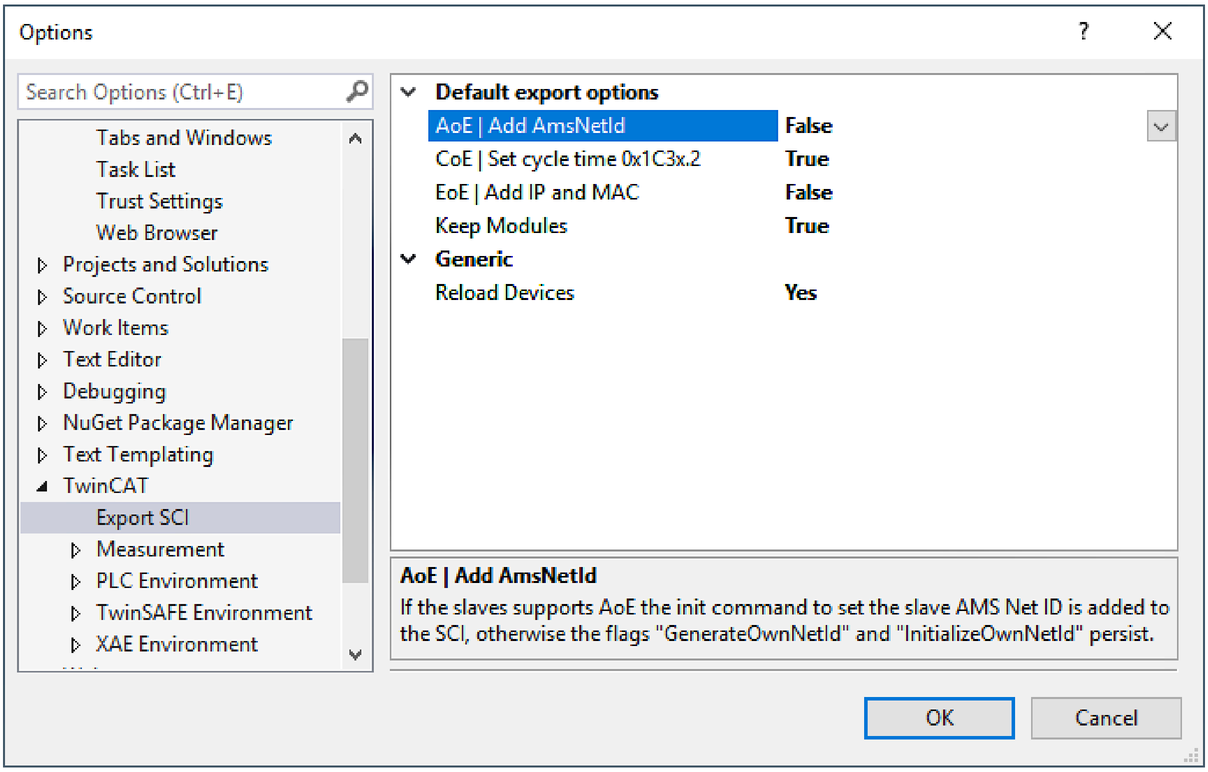


- Display SCI devices and select and insert the desired device:



Additional Notes

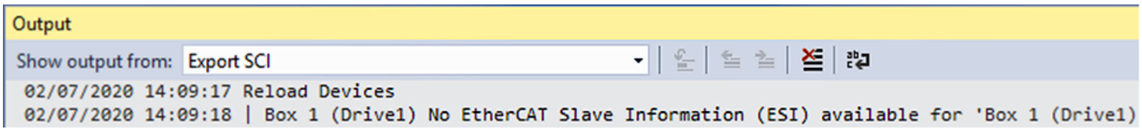
- Settings for the SCI function can be made via the general Options dialog (Tools → Options → TwinCAT → Export SCI):



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:



6.3 General Commissioning Instructions for an EtherCAT Slave

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.

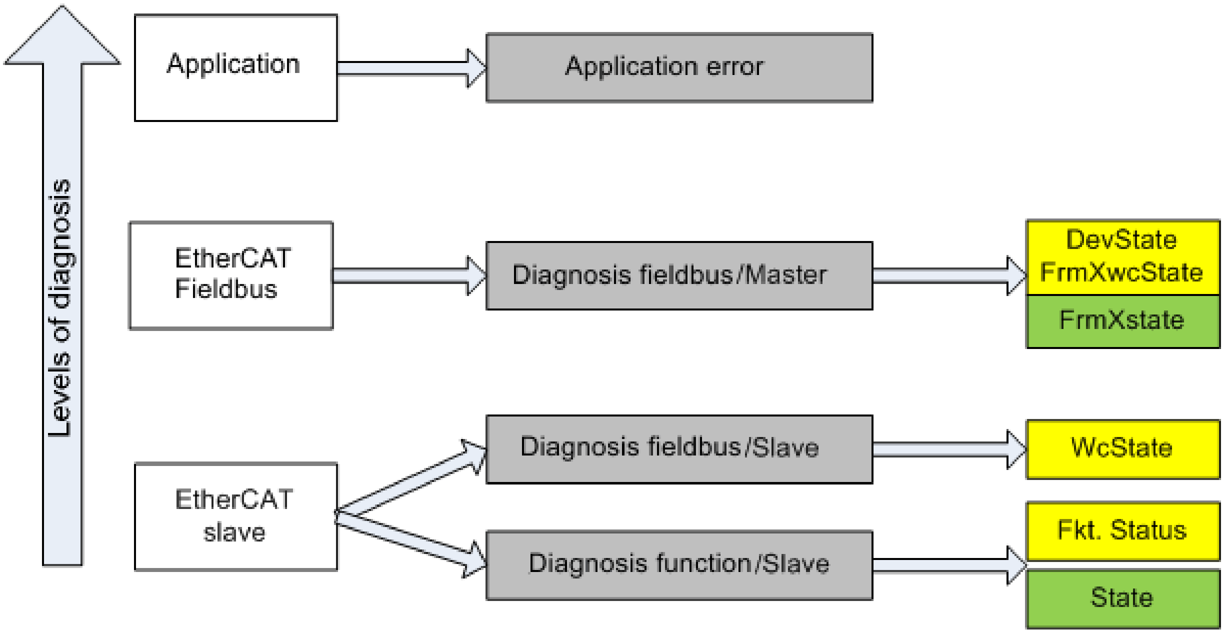


Fig. 134: Selection of the diagnostic information of an EtherCAT Slave

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.

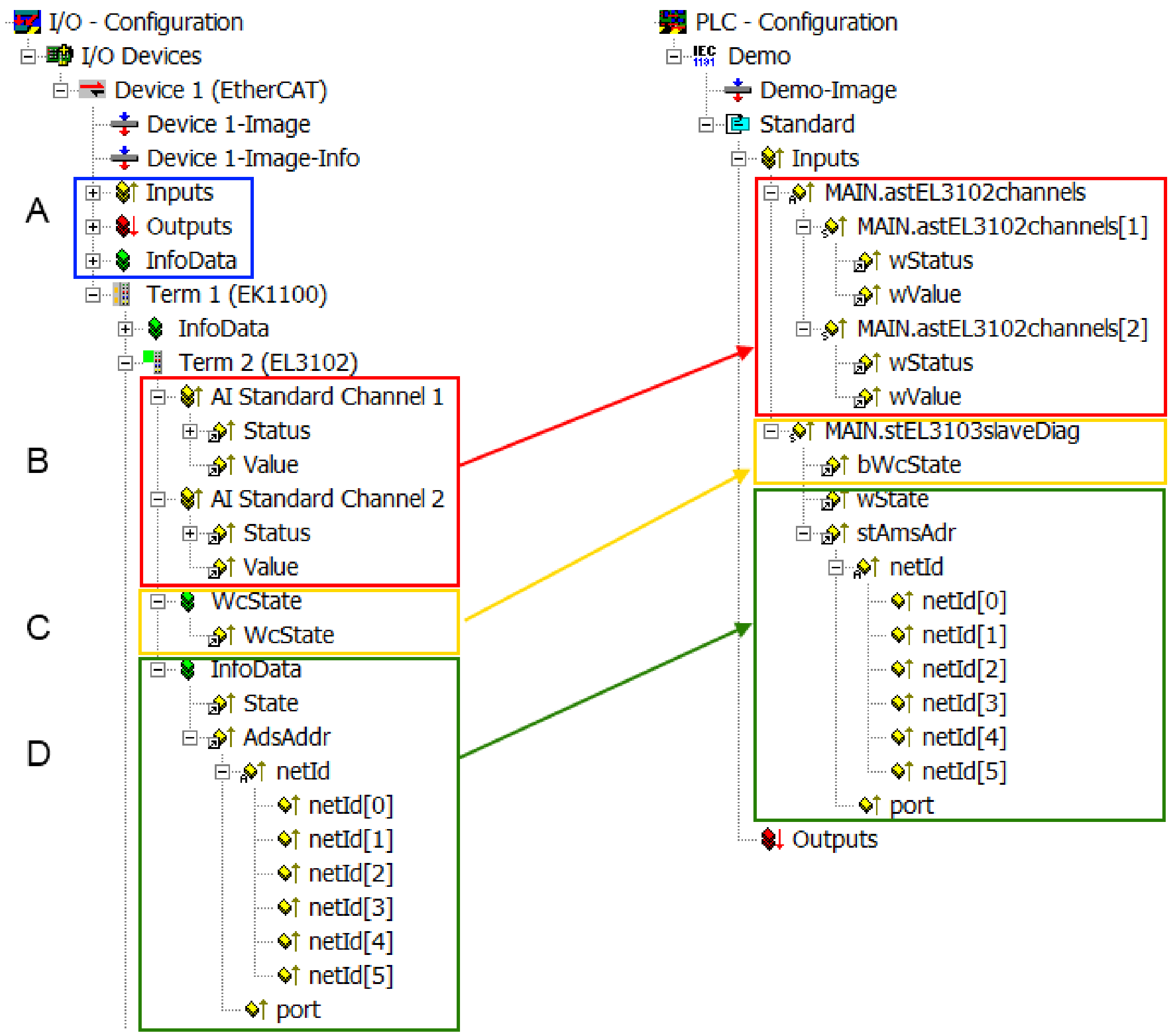


Fig. 135: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated cyclically (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.

NOTICE

Diagnostic information

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

CoE Parameter Directory

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

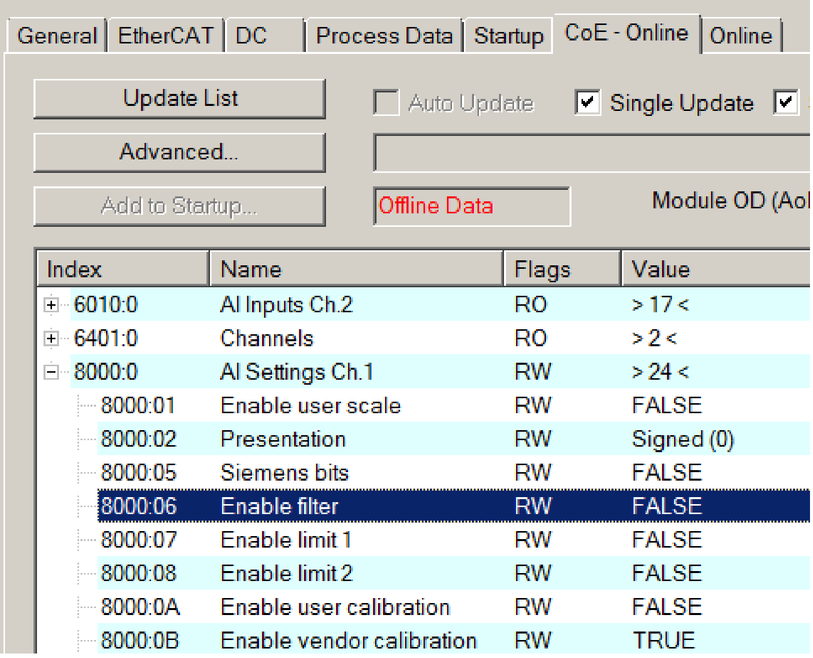


Fig. 136: EL3102, CoE directory

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.

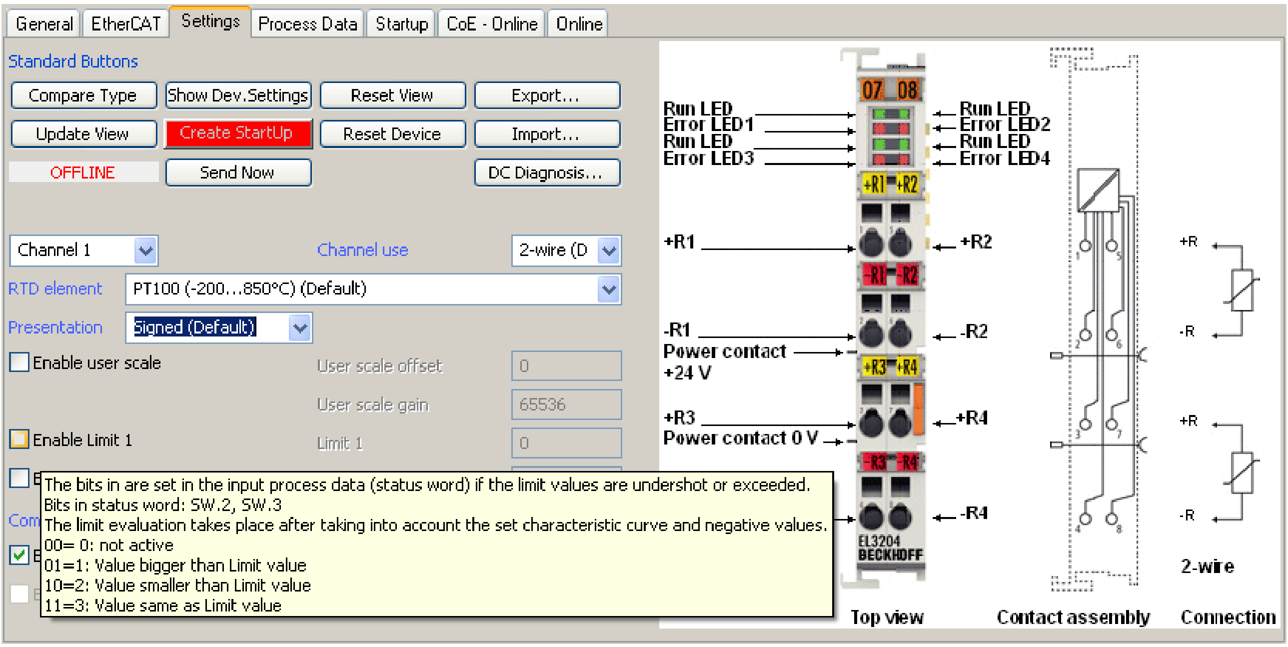


Fig. 137: Example of commissioning aid for a EL3204

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 [▸ 32]" 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.

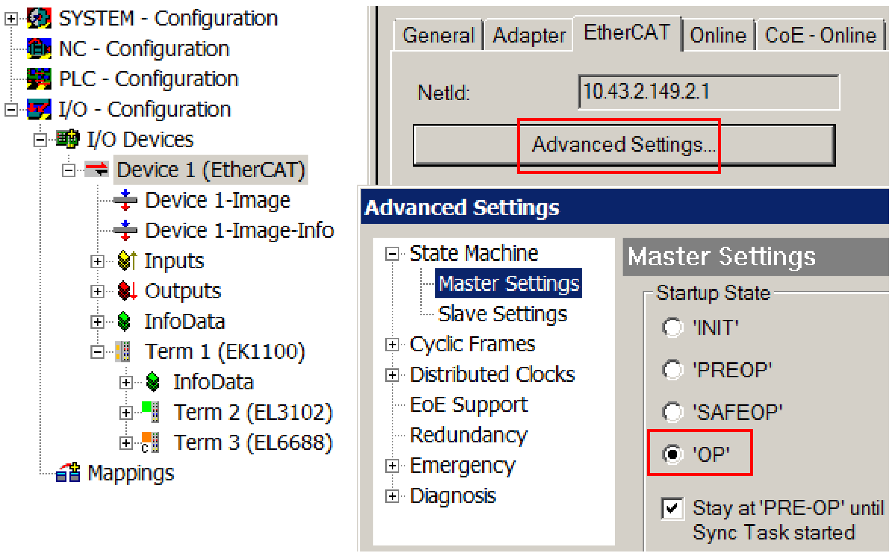


Fig. 138: Default behaviour of the System Manager

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

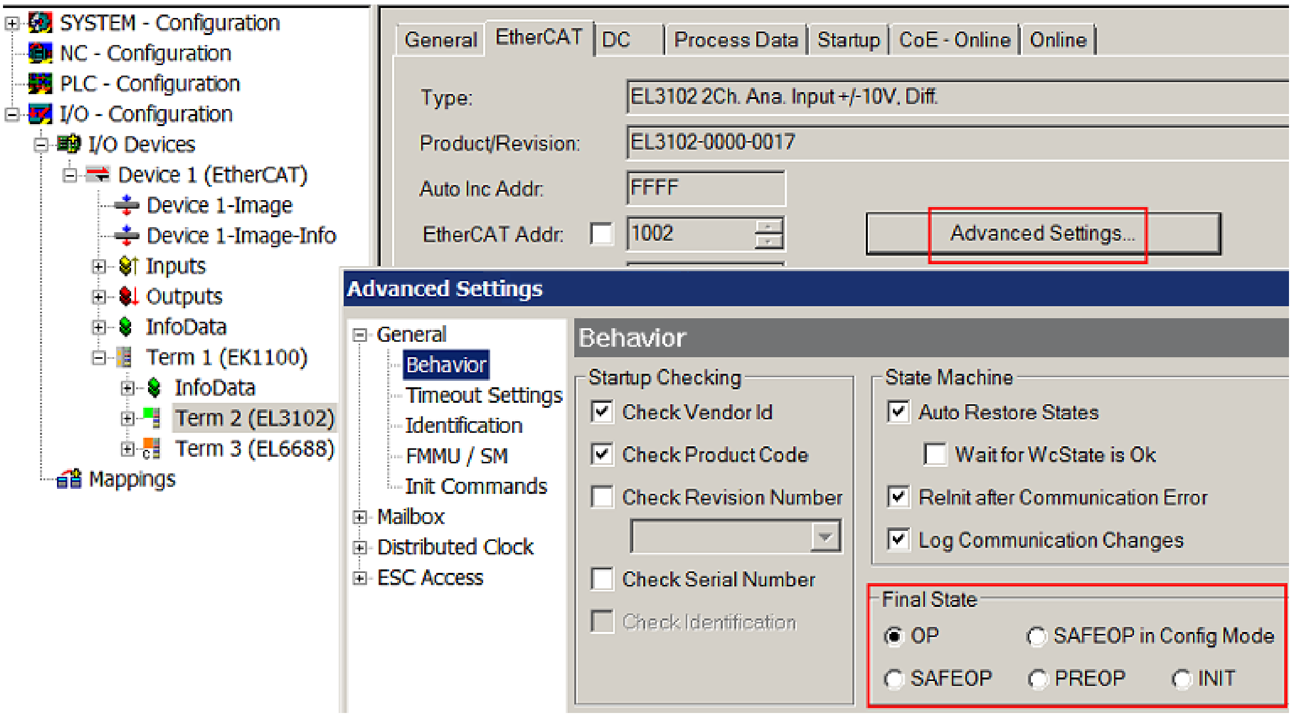


Fig. 139: Default target state in the Slave

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.

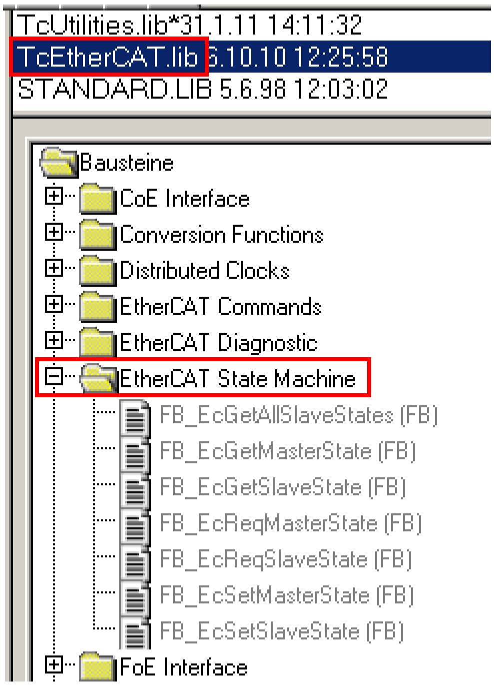
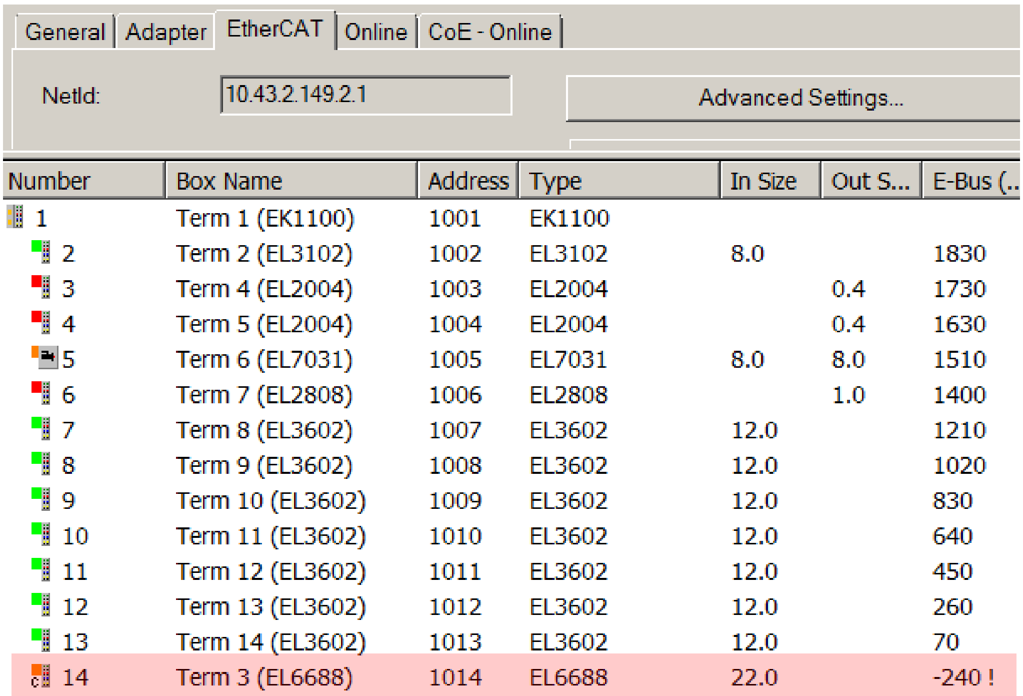


Fig. 140: PLC function blocks

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. 141: 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:

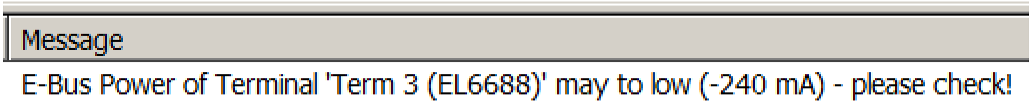


Fig. 142: Warning message for exceeding E-Bus current

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

6.4 Oversampling terminals/box modules and TwinCAT Scope

Generally, input data of a terminal/box could be achieved by the scope either directly (via the activated ADS server) or by creation of a PLC variable which is linked to the PDO of a terminal/box for recording them. Both procedures will be explained for TwinCAT 3 (TC3) at first and for TwinCAT 2 (TC2) respectively.

Oversampling means that an analog or digital input device supplies not only one measured value for each process data cycle/EtherCAT cycle (duration T), but several, which are determined at a constant interval t < T. The ratio T/t is the oversampling factor n.

A channel thus offers not only one PDO for linking in the process data, as in the example here with the EL3102, but n PDOs as in the case of the EL3702 and other oversampling terminals/box modules.

The definition of "oversampling" by the Beckhoff's point of view shouldn't be mixed up with the oversampling process of a deltaSigma ADC:

- **deltaSigma ADC**: the frequency used by the ADC to sample the analogue signal is faster than a multiple times than the frequency of the provided digital data (typically in kHz range). This is called oversampling resulting by the functional principle of this converter type and serve amongst others for anti-aliasing.
- **Beckhoff**: the device/ the terminal/box read of the used ADC (could be a deltaSigma ADC also) digital sample data n-times more than the PLC/ bus cycle time is set and transfers every sample to the control – bundled as an oversampling PDO package.

For example, these both procedures are arranged sequentially by their technical implementation within the EL3751 and can also be present simultaneously.

EL3102

Name	Type	Size
Status	Status_4099	2.0
Value	INT	2.0
Status	Status_4099	2.0
Value	INT	2.0

EL3702

Name	Type	Size
Ch1 CycleCount	UINT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch1 Value	INT	2.0
Ch2 CycleCount	UINT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0
Ch2 Value	INT	2.0

Fig. 143: Oversampling PDO of the EL37xx series and in the comparison with EL31xx

Accordingly, the Scope2 (TC2) or ScopeView (TC3) can read in and display several PDOs per cycle in correct time.

6.4.1 TwinCAT 3 procedure

From TwinCAT 3.1 build 4012 and using the revision as below specified in the configuration, the integrated ScopeView recognizes in its variable browser that the oversampling data is an array package and activates ForceOversampling automatically. The array as a whole must be selected using *AddSymbol* (see description in the next section).The extended PDO name provides the basis for this. Since a specific revision of the respective terminal ScopeView is able to detect the array type of a set of variables autonomous.

Terminal	Revision
EL4732	all
EL4712	all
EL3783	EL3783-0000-0017
EL3773	EL3773-0000-0019
EL3751	all
EL3742	all
EL3702	all
EL3632	all
EL2262	all
EL1262-0050	all
EL1262	all
EP3632-0001	all
EPP3632-0001	all

Recording a PLC Variable with the TwinCAT 3 – ScopeView

By a precondition of an already created TwinCAT 3 – project and a connected PLC with an oversampling able terminal/box within the configuration it will be illustrated how an oversampling variable can be represented by the Scope (as a standard part of the TwinCAT 3 environment). This will be explained by means of several steps based on an example project “SCOPE_with_Oversampling“ as a standard PLC project.

Step 1: Adding a project „Scope YT“

The example project “SCOPE_with_Oversampling“ has to be added a TwinCAT Measurement – project “Scope YT project” (C) by right click (A) and selection (B) “Add” → “New Project..”. Then “Scope for OS” will be entered as name. The new project just appears within the solution explorer (D).

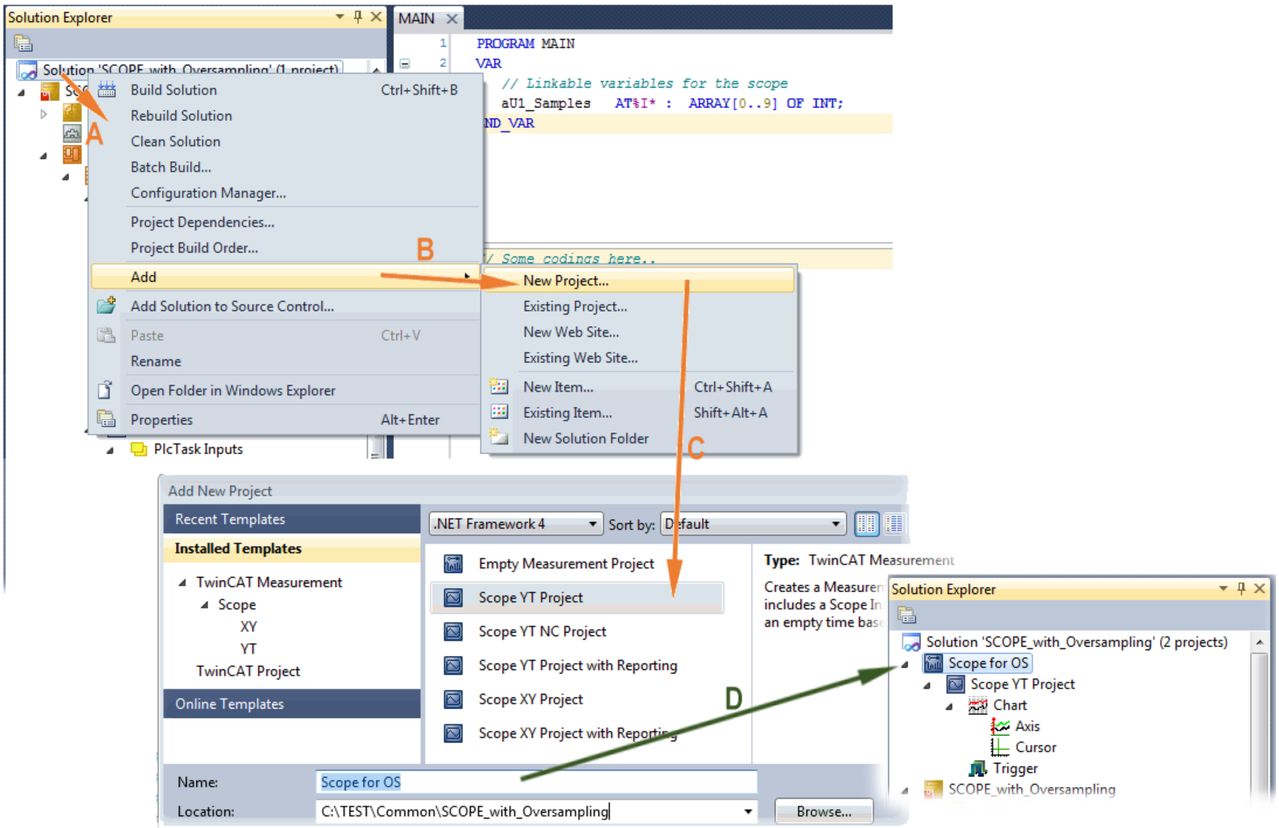


Fig. 144: Adding a Scope project into an already existing project

Step 2a: Creation of a PLC variable within a POU

Within the TwinCAT 3 development environment an input variable as an array with respective amount than is given by the oversampling factor have to be defined at first how it’s illustrated in an example for the POU “MAIN” and an oversampling factor 10 with structured text (ST) as follows:

```
PROGRAM MAIN
VAR
   aU1_Samples AT%I* : ARRAY[0..9] OF INT;
END_VAR
```

The identification “AT%I*” stands for swapping out this array variable to link it with the process data objects (PDOs) of a terminal/box later. Notice that at least the number of elements has to be the same as the oversampling factor so that the indices can be set from 0 to 9 also. As soon as the compiling procedure was started and ended successful (in doing so no program code may be present) the array appears into the solution explorer of the TwinCAT 3 development environment within the section PLC under “..Instance”.

The following illustration shows extracts of the solution explorer on the right. As an example that linking of an array variable to a set of oversampling process data of an EL3773 is represented herewith:

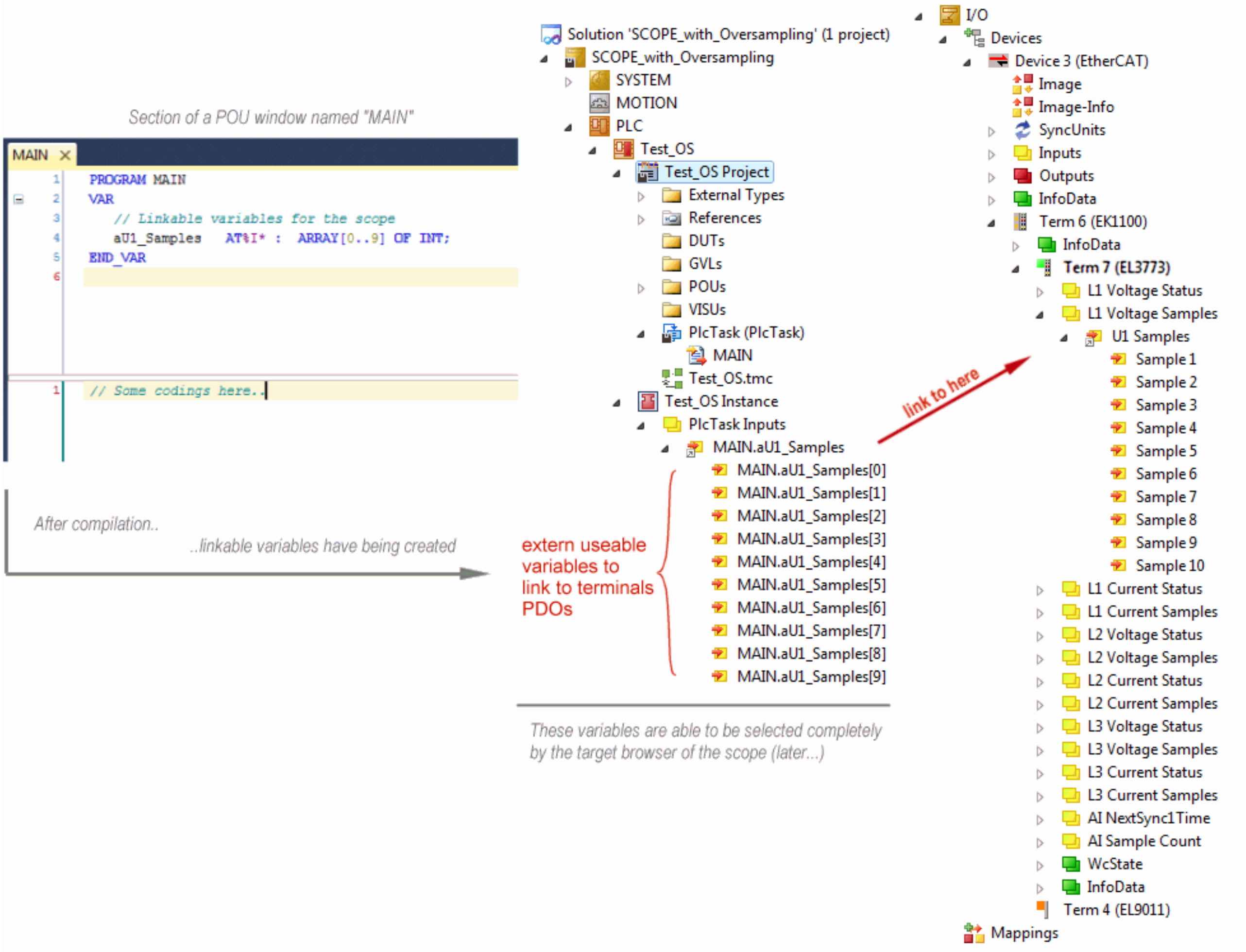


Fig. 145: Representation of a created PLC array variable („aUI_Samples“) to link with oversampling PDOs of EL3773

Step 2b: Creation of a PLC variable via a free task

When a POU is not needed onto the particular system, a referenced variable could be applied via a free task also. If a free task is not existing still yet, it can be created by a right-click to “Task” of the project within SYSTEM with “Add New Item…”.

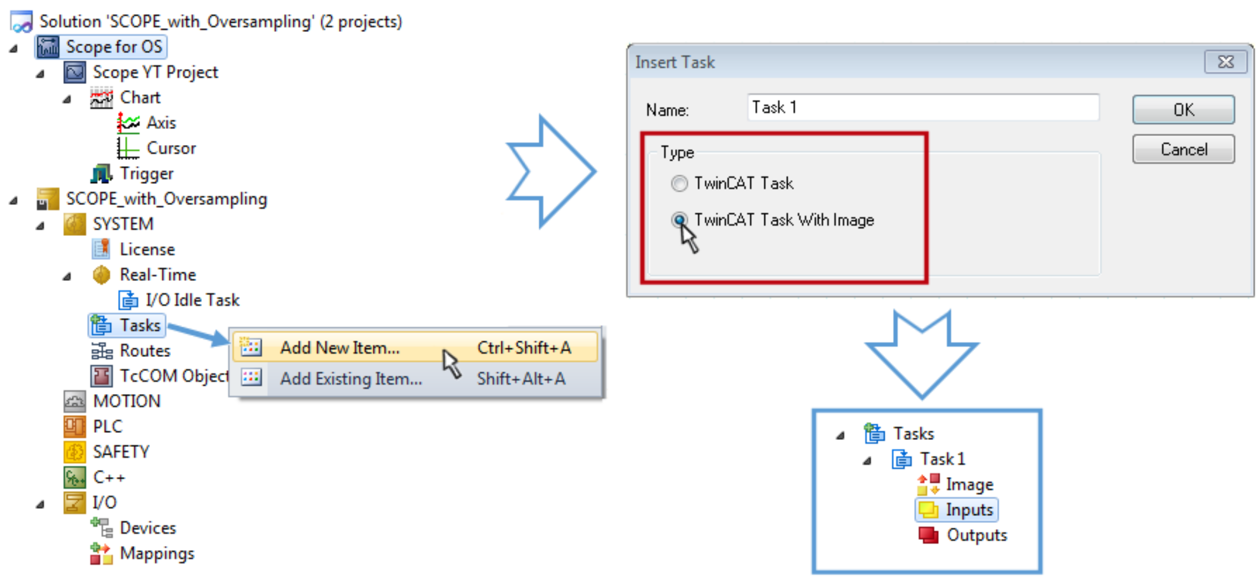


Fig. 146: Insertion of a free task

The Task has to be inserted as “TwinCAT Task With Image” and also creates an “Inputs” and “Outputs” folder therefore. The properties of the new (or as the case may be already existing) task must have activated the attribute “Create symbols” to make them selectable by the “Target Browser” of the Scope later on. The task cycle time has to be changed if so. Then, with10 x Oversampling 1 ms at 100 µs base time, resulting 10 ticks will be set by the usage of the EL3751 for example:

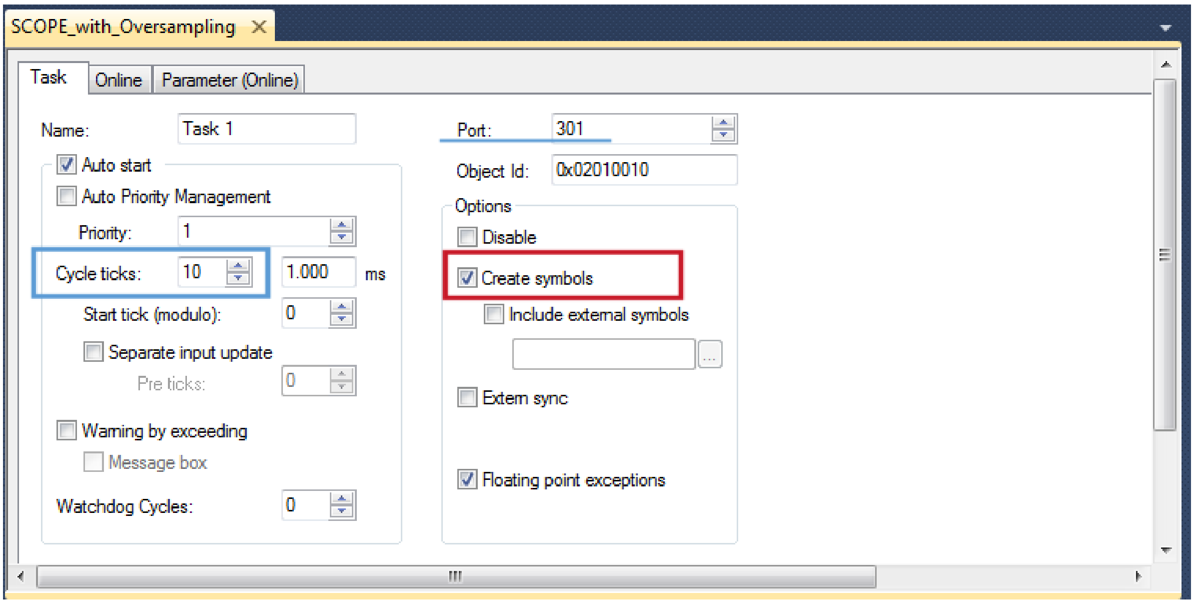


Fig. 147: Task property "Create symbols" must be activated

There’s a default value given for the Port number (301) that should be changed, if necessary. This number has to make acquainted for the Scope, if applicable, later on. By a right click on “Inputs” that oversampling based variable can now be appended with the fitting datatype of an array. „ARRAY [0..9] OF DINT“ referred to as „Var 1“ in this case:

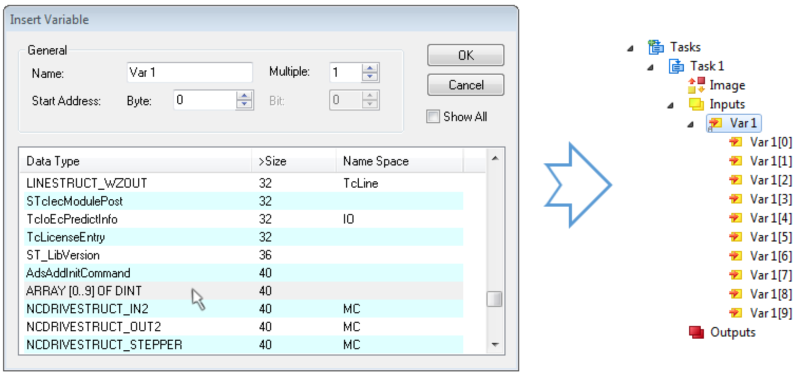


Fig. 148: Insertion of variable "Var 1" fitting to the oversampling (-factor)

Step 3: Linking an array variable with an oversampling PDO

By right click on “MAIN.aUI_Samples” (according to the last preceding paragraph Step 2a) or rather “Var 1” of the free Task 1 (according to the last preceding paragraph Step 2b) within the Solution Explorer a window opens to select the process data:

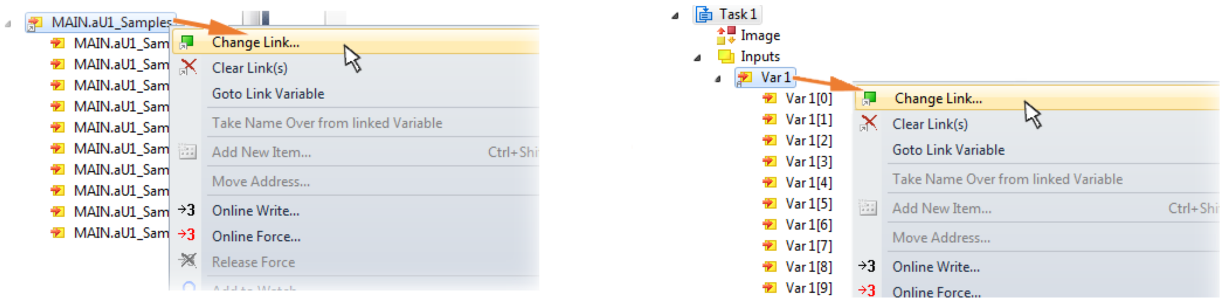


Fig. 149: Set up the link of the PLC array variable (left: for the last preceding paragraph Step 2a, right: for the last preceding paragraph Step 2b)

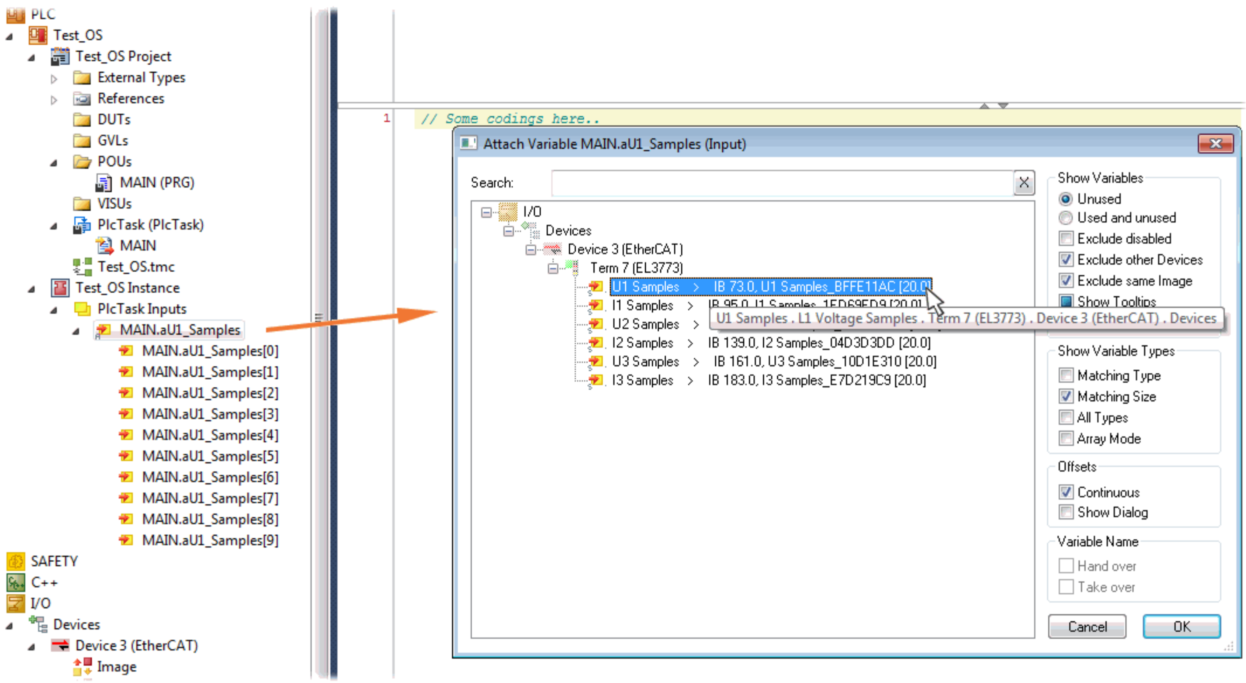


Fig. 150: Select the EL3773 PDO "L1 Voltage Samples" to create a link to the PLC array variable „aUI_Samples“

The selection of PDO "U1 Samples" of the EL3773 for “MAIN.aUI_Samples” based by the last preceding paragraph Step 2a as illustrated above have to be done in the same way for “Var 1” accordingly.

Step 4: Selection of the PLC array variable for the Y-axis of the scope

Now the configuration will be activated () and logged in the PLC (), so the array variable will be visible for the target browser of the scope for being selected.

Thereby the drop down menu will be opened by right clicking on “Axis” (A) for selection of the scope features (B):

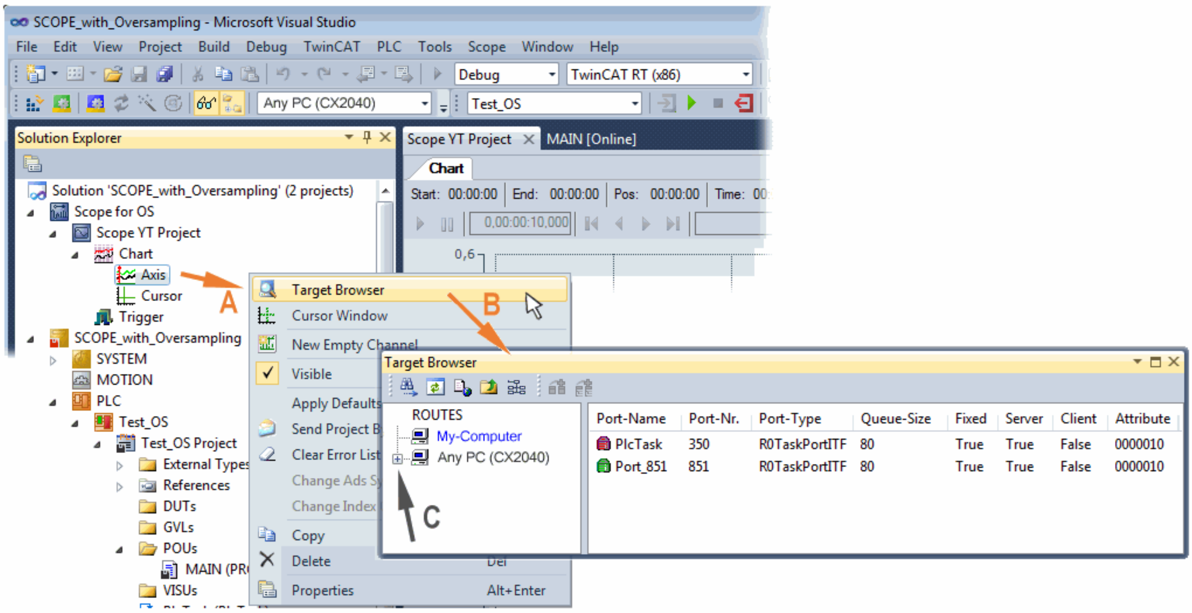


Fig. 151: Selection of the oversampling variable with the target browser

By addressing the corresponding system that represents the PLC containing the array variable ("Any PC (CX2040)" in this case) navigation up to the variable "aUI_Samples" (C) have to be done.

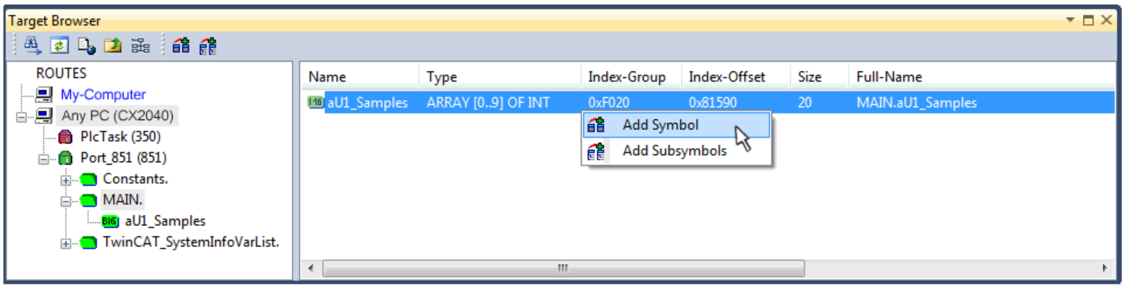
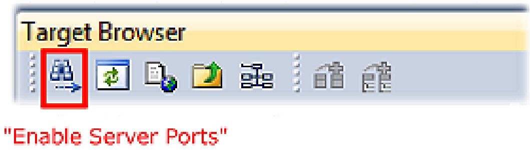


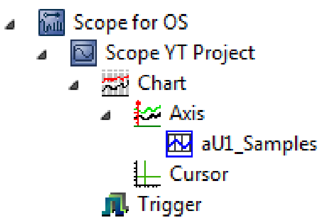
Fig. 152: Appending the variable "aUI_Samples" below "axis" within the scope project of the solution explorer

Variable don't appears into the target browser

If „ROUTES" don't offer a possibility for selection of the provided variables, the corresponding port should be declared for the target browser:



Using "Add symbol" displays the variable "aUI_Samples" below "axis" within the scope project of the solution explorer directly.



Now the program start has to be done with formally although there's no program still yet. Using "Start Recording" the process data value of the oversampling PDO "L1 Voltage Samples " via the linked PLC array variable can be recorded time dependent now.

As an example a sine wave input measurement value (204.5 Hz) will be illustrated below:

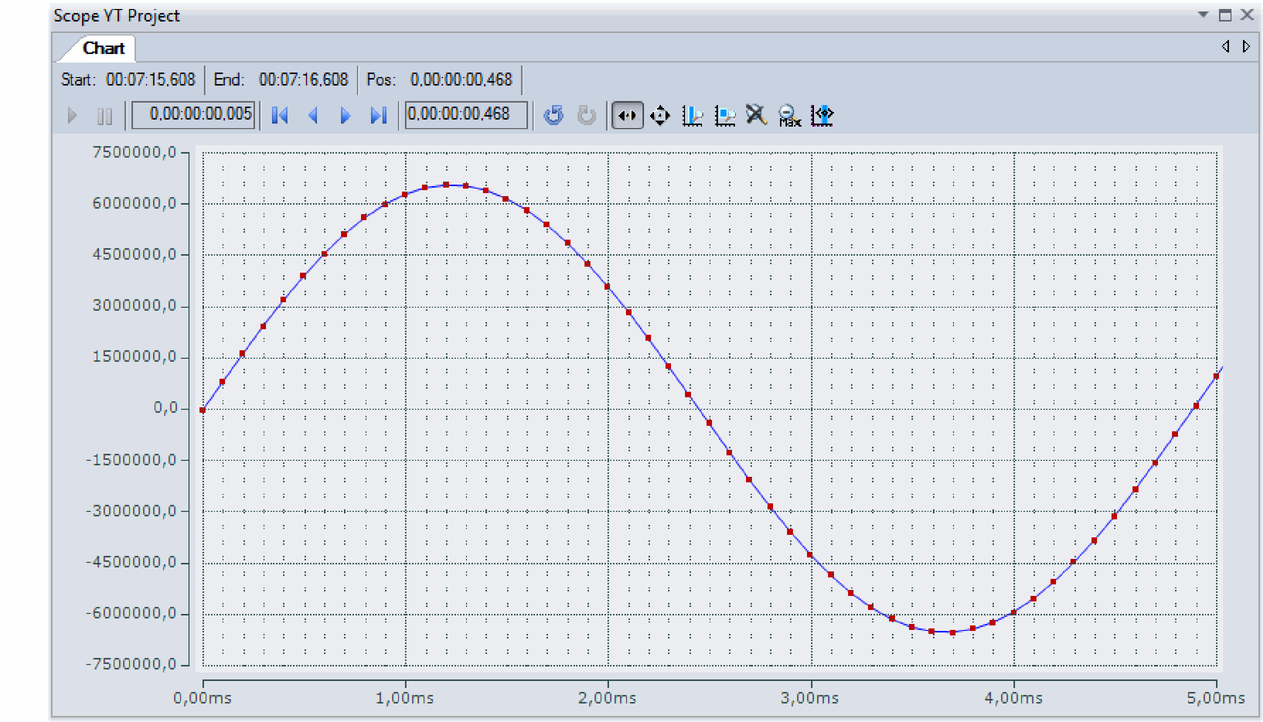


Fig. 153: Example of recording a sine signal with 10 x oversampling at 1 ms measurement cycle time

The X-axis view was fitted properly by using "Panning X" after the recording was stopped . Following the "Chart" property "Use X-Axis SubGrid" was set to true with 10 divisions as well as the "ChannelNodeProperties" attribute "Marks" was set to "On" with the colors "Line Color" blue and "Mark Color" red. Therefore the latter indicates that 10 oversampling measurement points by the red marks.

Proceeding with / via ADS alternatively

In former TwinCAT 3 versions (or a lower revision as specified in the table [▶ 137] above) the oversampling PDO of the respective oversampling able terminal/box can be made visible for the ScopeView by activation of the ADS server.

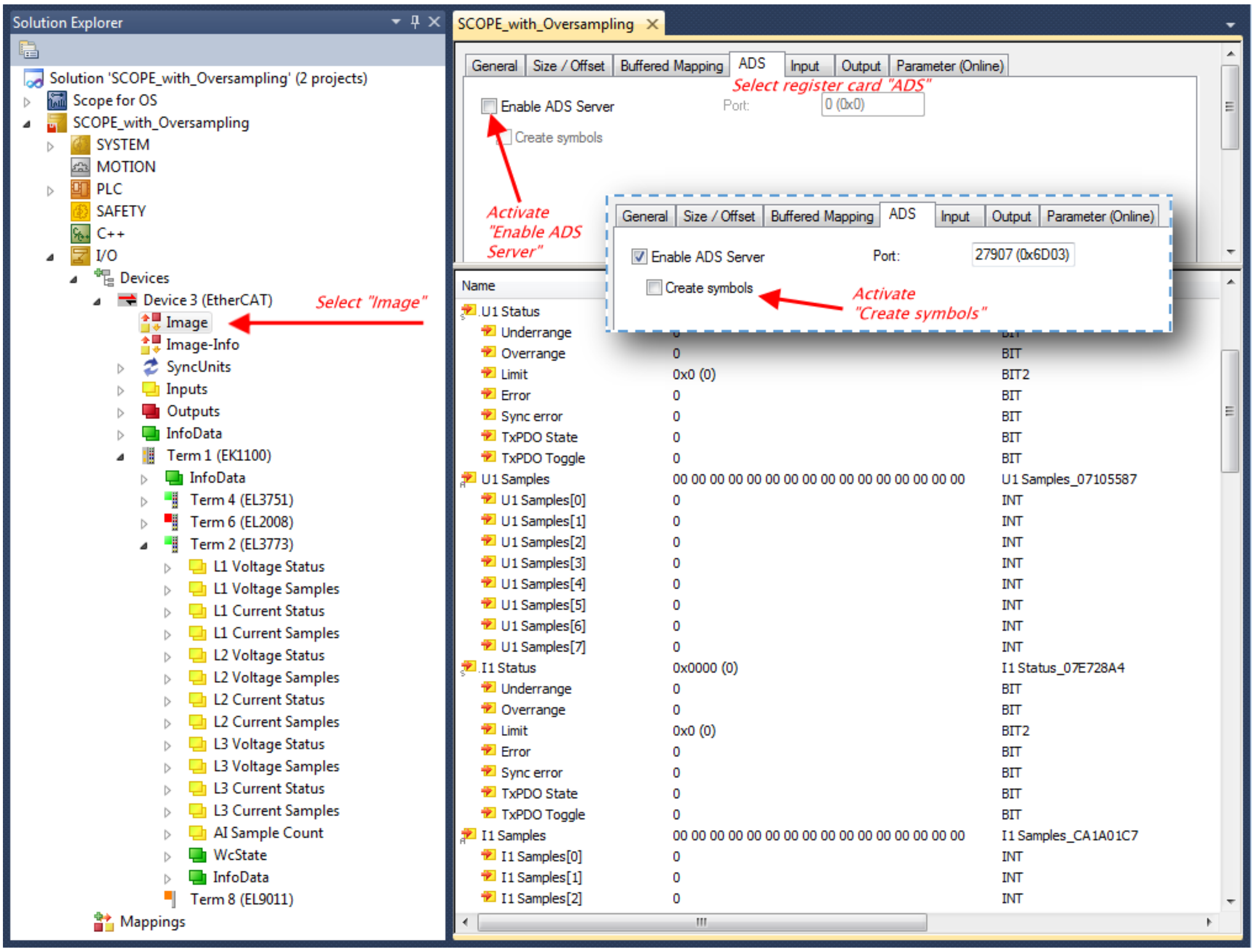


Fig. 154: Activation of the ADS server of the EtherCAT device (TwinCAT 3)

The activation of the server can be carried out by selection of “Image” within the left sided solution explorer: „I/O → Devices → Device .. (EtherCAT) → Image“.

Next the register card “ADS” have to be selected to activate each checkbox „Enable ADS Server“ and „Create symbols“ then (the port entry is done automatically).

Thereby it is possible to access process data without an embedded POU and accordingly without a linked variable:

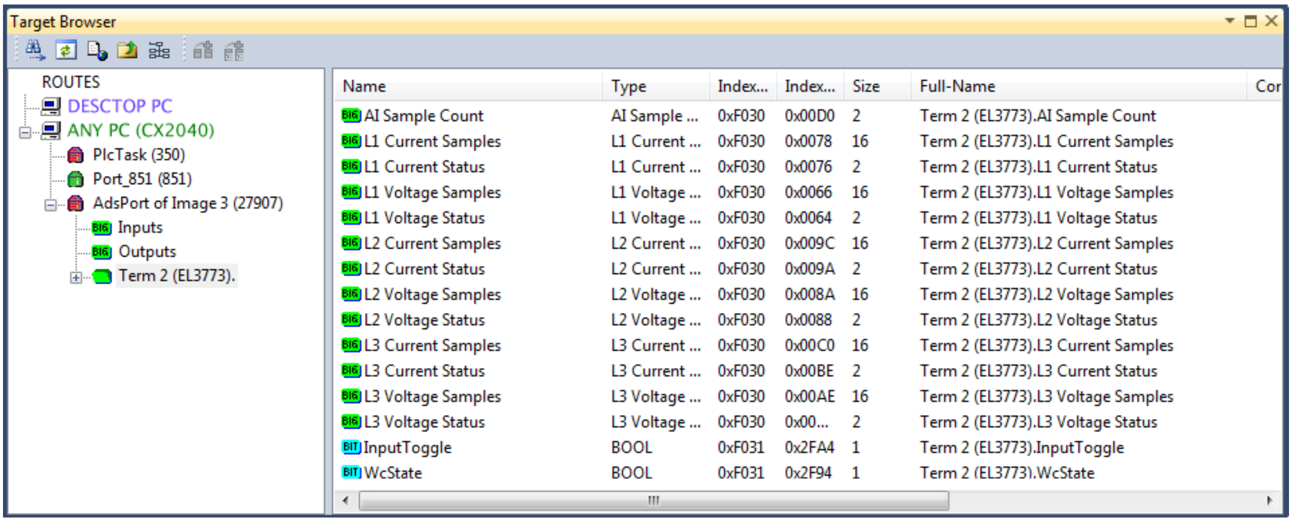
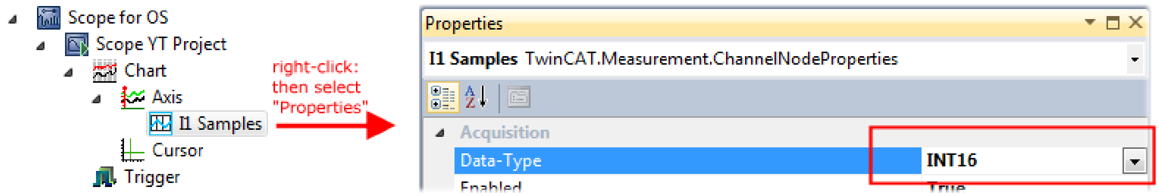


Fig. 155: Direct access to PDOs of the terminal by ScopeView

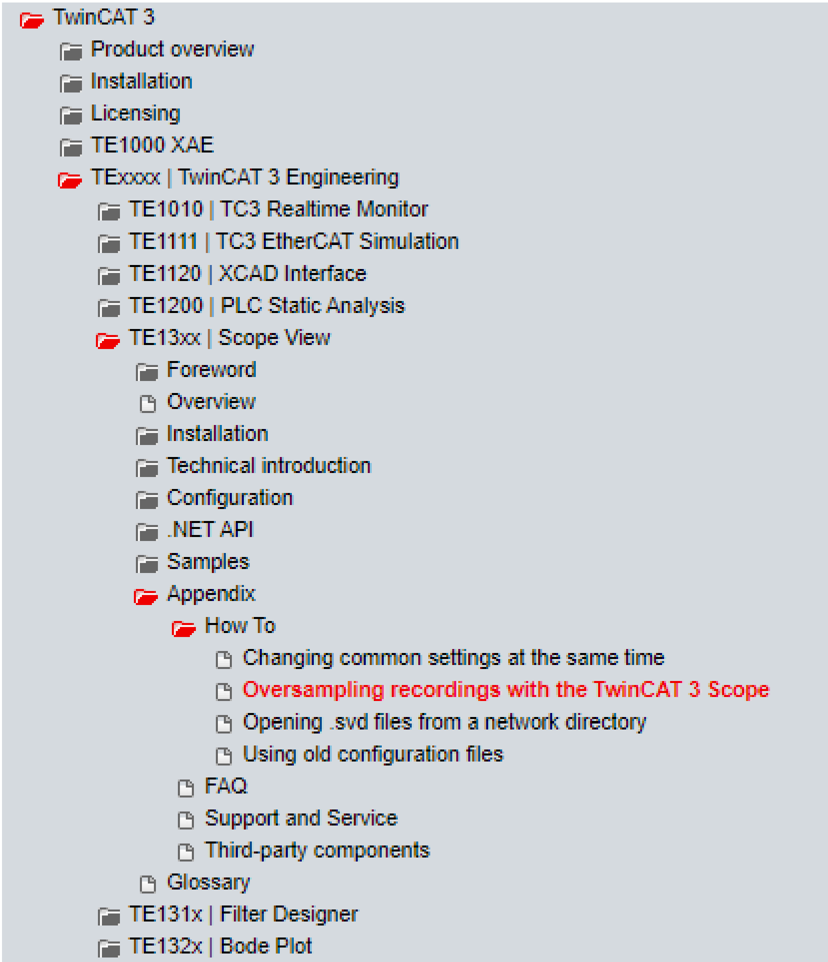
Data type not valid

It may happen that the target browser is unable to determine the data type after insertion of the oversampling PDO (according to an array variable usually). In this case it can be changed by the channel properties:



TwinCAT 3: Activate the ADS Server of an EtherCAT device

Also see Beckhoff Information System:
infosys.beckhoff.com → TwinCAT 3 → TExxxx | TwinCAT 3 Engineering → TE13xx | ScopeView → Appendix → How To → Oversampling recordings with the TwinCAT 3 Scope



6.4.2 TwinCAT 2 procedure

The TwinCAT Scope2 supports the import and display of oversampling process data such as is used by oversampling-able terminals/box modules.

System requirements

A TwinCAT Scope2 must be installed on the system.
An oversampling-able terminal must be present in the configuration.

The data type of the variables is also conveyed to the TwinCAT Scope2 via the ADS data. Therefore, the array variable must be created

- in the PLC, see step 1a [▶ 146]
- or directly in the System Manager if only one free task is present, see step 1b [▶ 146]

The same settings are to be made in the Scope2 for both cases, see step 2 [▶ 148]

Recording of a PLC variable with the TwinCAT 2 – Scope2

Step 1a: TwinCAT 2 PLC

Since the channel data are to be used in the PLC, a linkable ARRAY variable must be created there, as shown in the following example:

```
VAR
    aiEL3773_Ch1_DataIn AT%I*: ARRAY[1..10] OF INT;
END_VAR
```

Fig. 156: PLC declaration

This then appears in the list in the System Manager; as a rule it can also be reached via ADS without further measures since PLC variables are always created as ADS symbols in the background.

Fig. 157: PLC in the System Manager

Note: The Scope2 can only "see" such variables in the variable browser if TwinCAT and the PLC are in RUN mode.

Step 1b: TwinCAT 2 - free task

The array variable required for Scope2 can alternatively be defined and created manually in the System Manager.

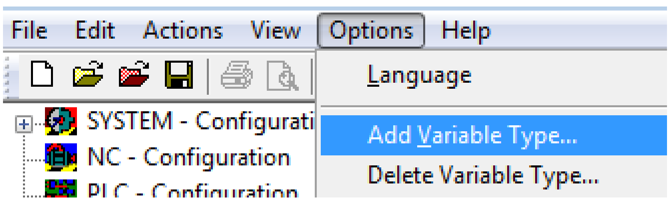


Fig. 158: Add Variable Type

As before in the program (POU "Main"), an ARRAY variable of the same type as from the oversampling PDO of the respective terminal/box must be created. In this example an array of 0..9 of the type INT, i.e. with 10 fields.

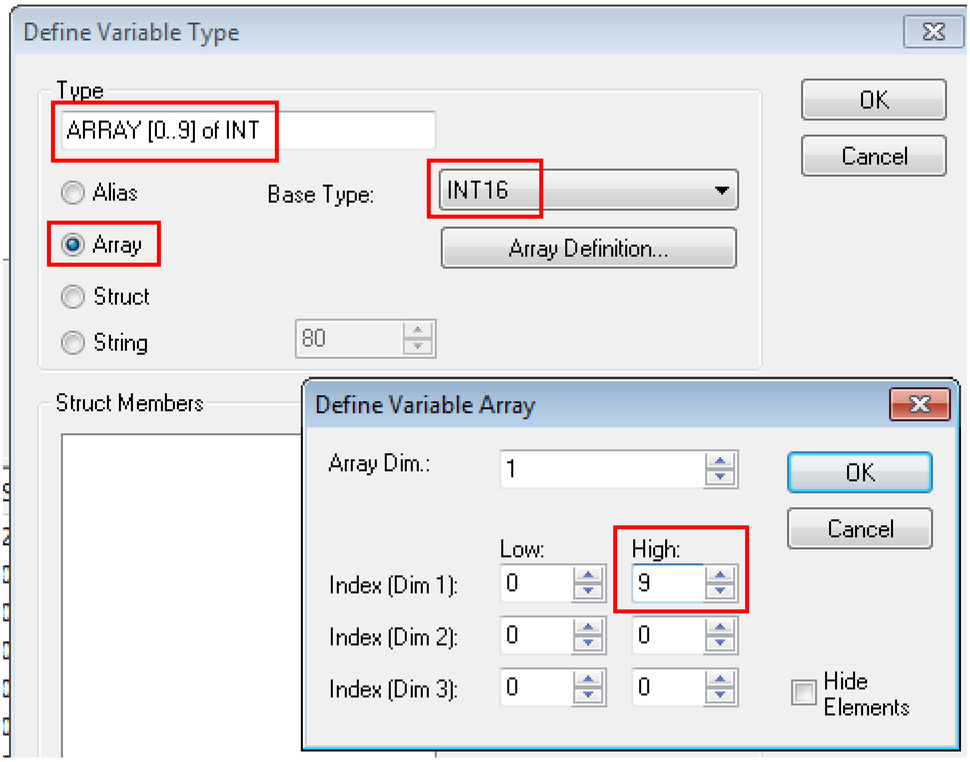


Fig. 159: Definition of the variable type

If this variable is known to the System Manager, an instance of it can be assigned to an additional task with a right-click. It appears in the overview, sorted according to bit size.

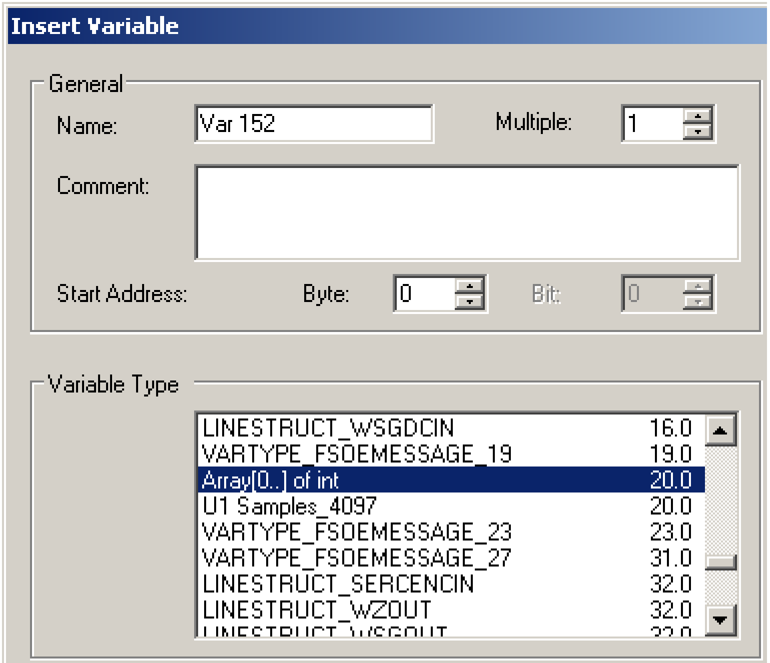


Fig. 160: Overview of declared types

In this example the variable *Var152* is created. It can now be linked with the PDO-Array of the respective channel of the terminal/box.

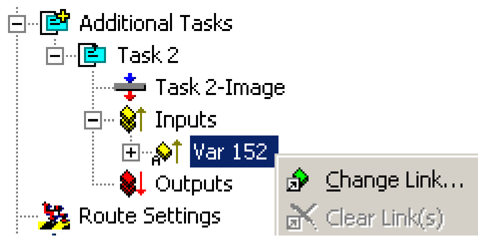


Fig. 161: Linking

If *MatchingSize* is activated in the dialog, the individual channels are offered directly.

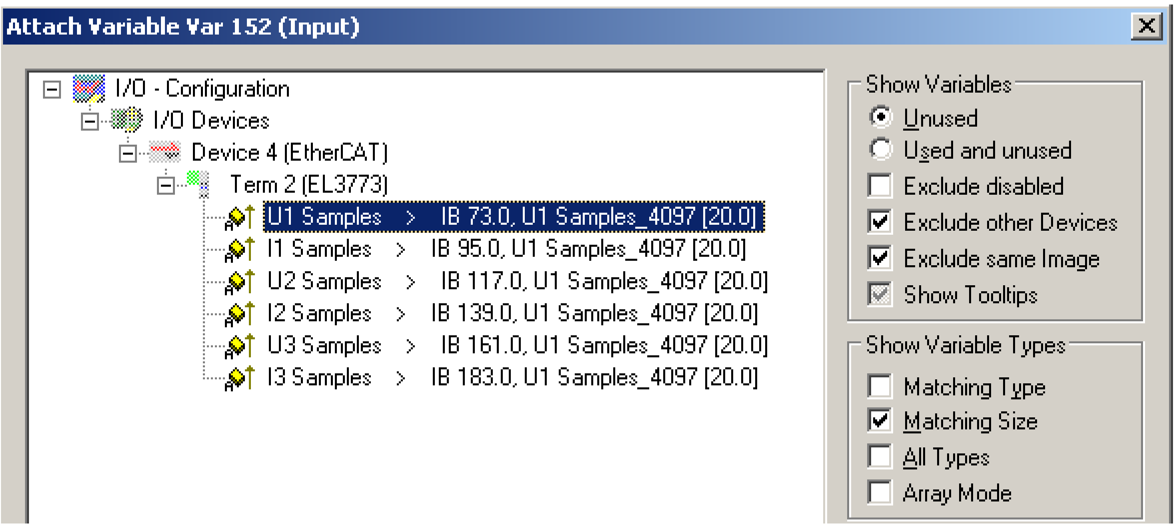


Fig. 162: Array variables of an oversampling terminal

So that the variables can also be found via ADS in the Scope2, the ADS symbols must be activated as well as the Enable Auto-Start, otherwise the task will not run automatically. ADS symbol tables are then created for all variables that have this task in their process data images.

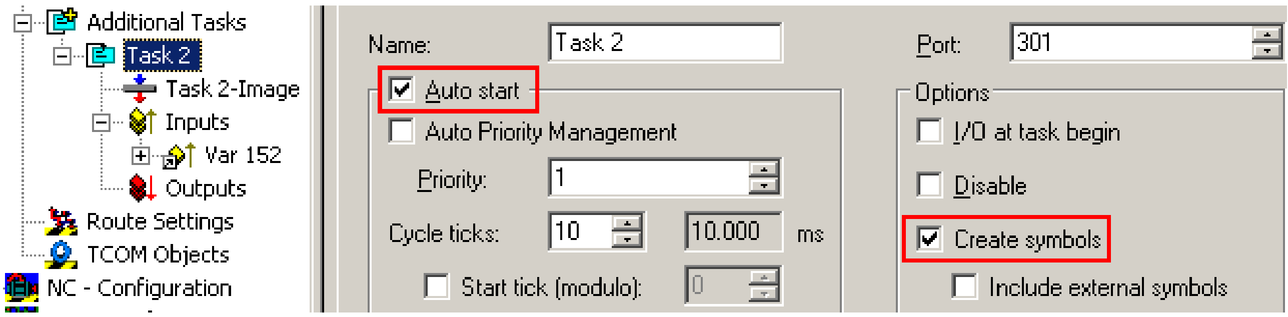


Fig. 163: Settings in the additional task

Step 2: Configuration in the Scope2

So that the linking works, an array variable with the channel data of the respective terminal/box must be present in the system manager; i.e. each oversampling data package must be present in an array. This array variable must be defined and created manually; see above [▶ 146].

You can now browse to the variable concerned in the Scope2.

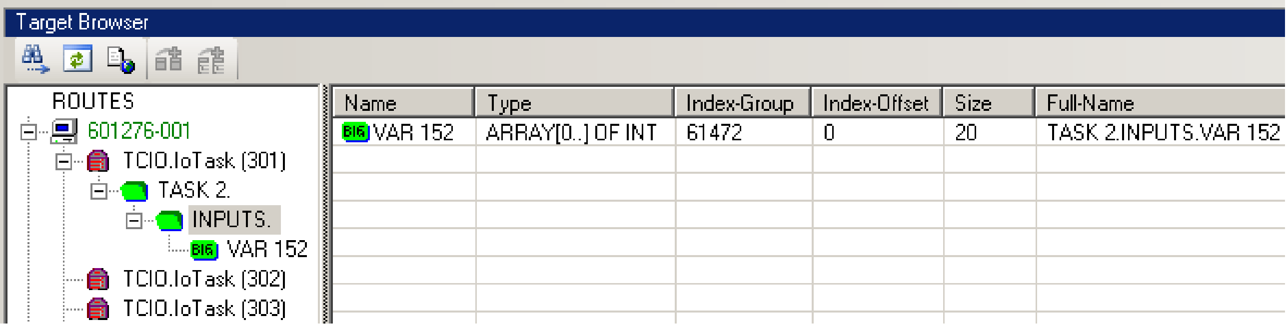


Fig. 164: Variable browser up to the array VAR152

The array is then not to be opened; instead the array symbol is to be selected by right-clicking on *AddSymbol*.

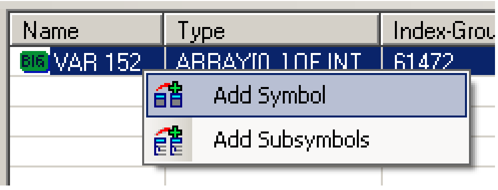


Fig. 165: AddSymbol on the array

ForceOversampling and *DataType* INT16 must be set in the channel which has now been created. If necessary *SymbolBased* must be temporarily deactivated in addition.

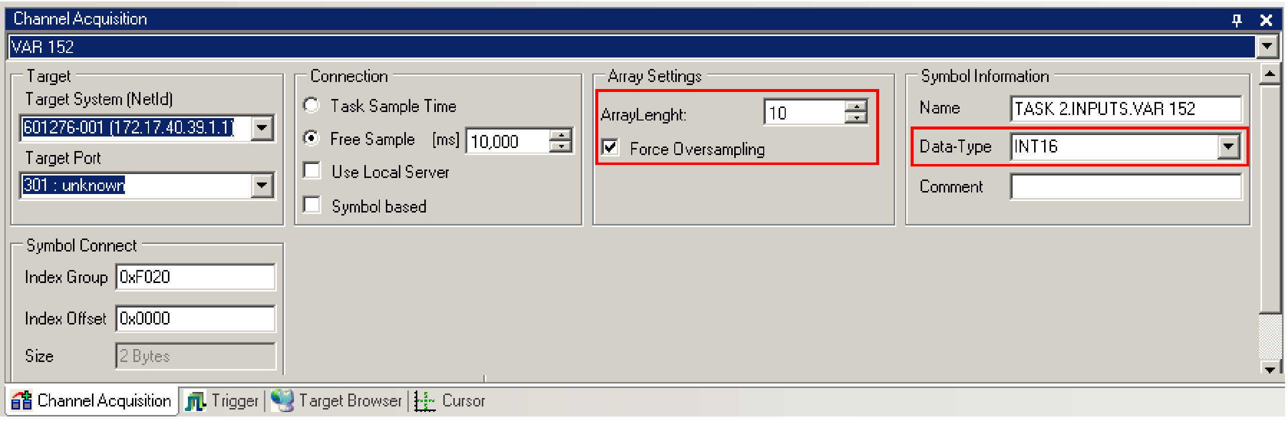


Fig. 166: Channel settings

In order to check that individual oversampling values are really being logged, the *Marks* can be activated in the Scope2. Please observe the interrelationships between task cycle time, sampling time of the Scope2 channel and oversampling factor.

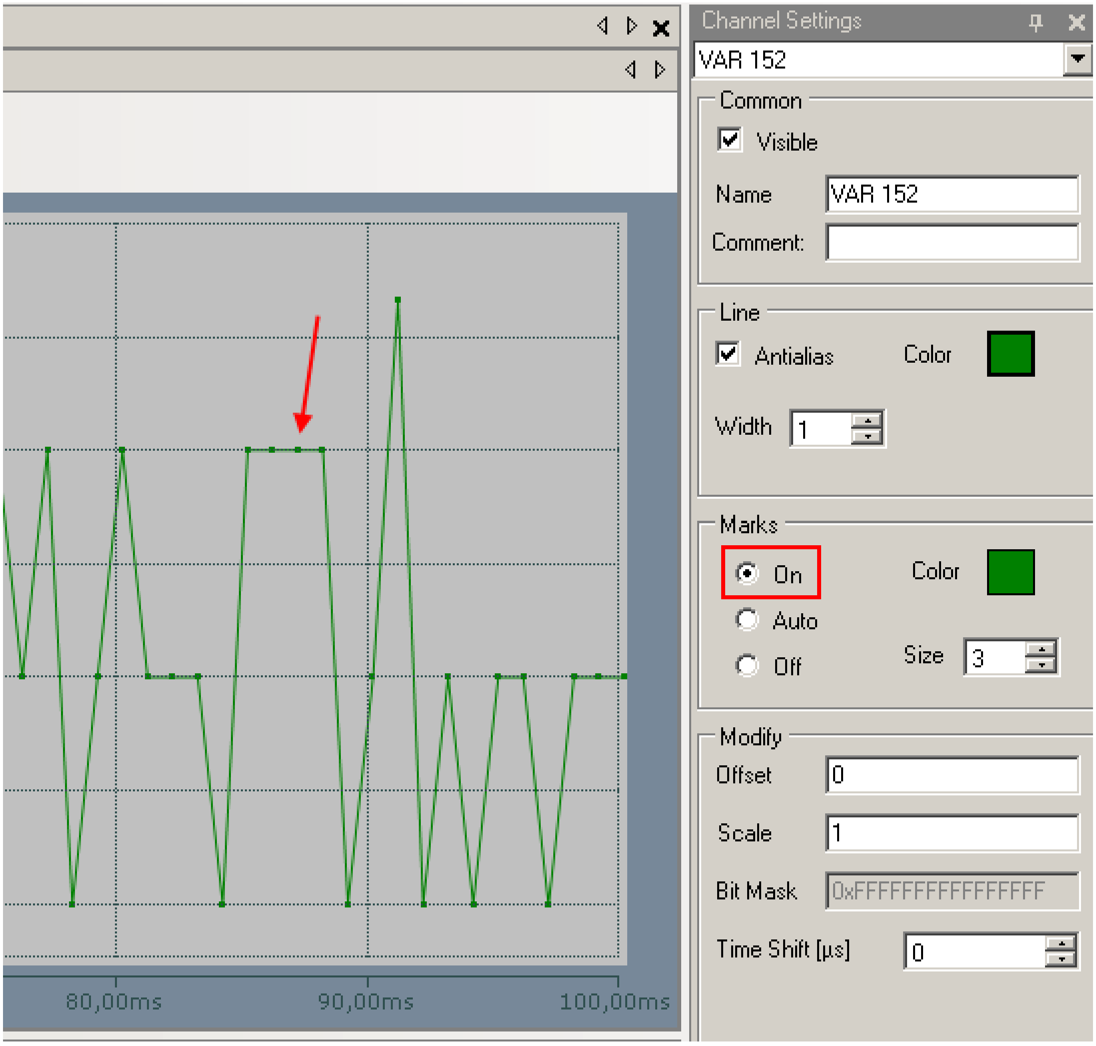


Fig. 167: Activation of the marks

An additional example illustrates the following image by representation of an oversampling – variable from the EL3751 with 10 x oversampling:

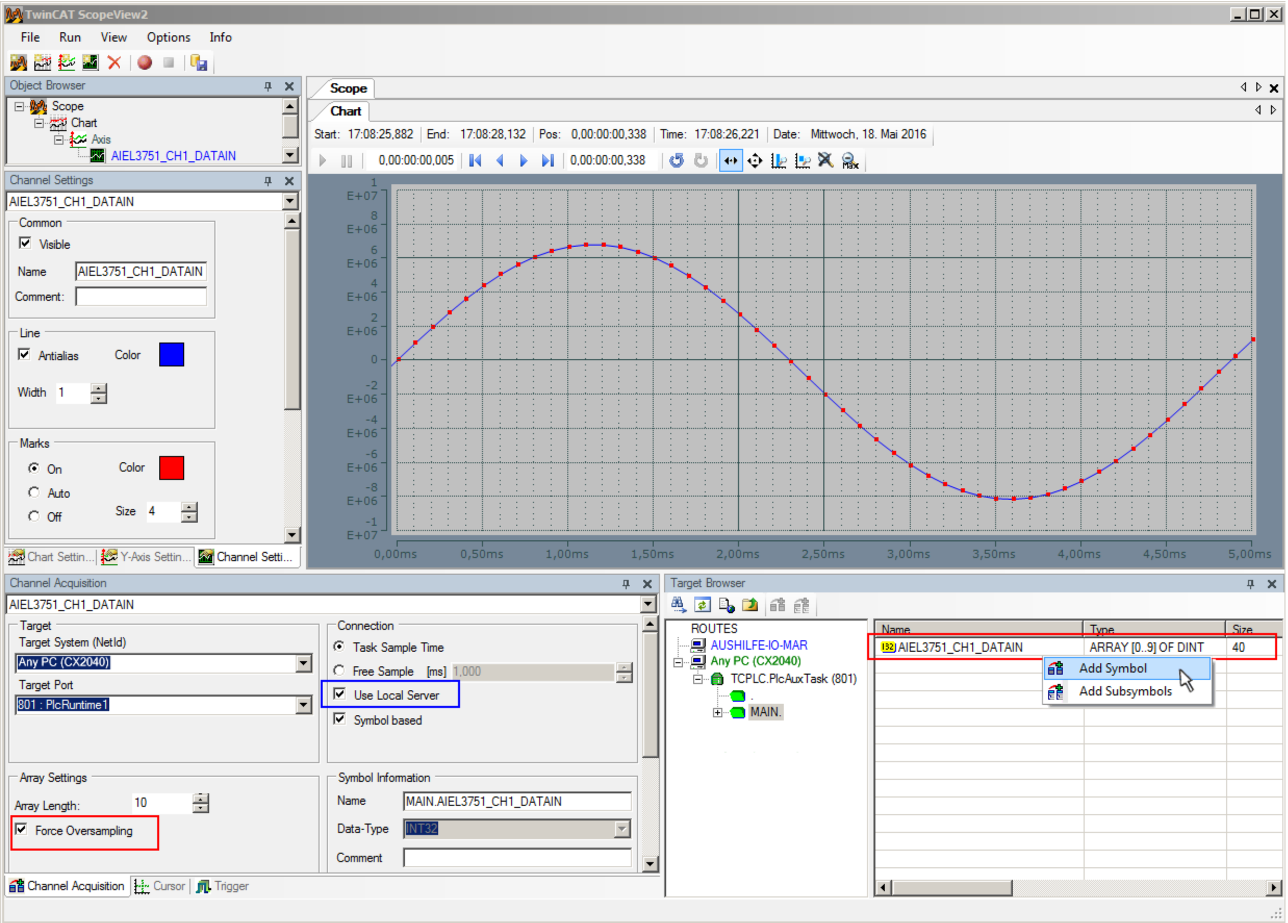


Fig. 168: Illustration of a 10 x oversampling variable of the EL3751 by the Scope2

Within the image was marked subsequently that the oversampling variable originated by the PLC was just added to the Y-axis (observe selection of the PLC-POU name “MAIN” within the “ROUTES” tree). Herewith “Force Oversampling“ was activated due to the oversampling variable is not provided by the terminal/box.

Proceeding with TwinCAT 2/ alternatively via ADS

In former TwinCAT 2 versions (or a lower revision as specified in the table [▶ 137] above) the oversampling PDO of the respective oversampling able terminal/box can be made visible for the Scope2 by activation of the ADS server.

So, the creation of a PLC variable can be disclaimed as well. Therefore, the ADS server of the EtherCAT Device where the oversampling able terminal/box is connected with have to be activated.

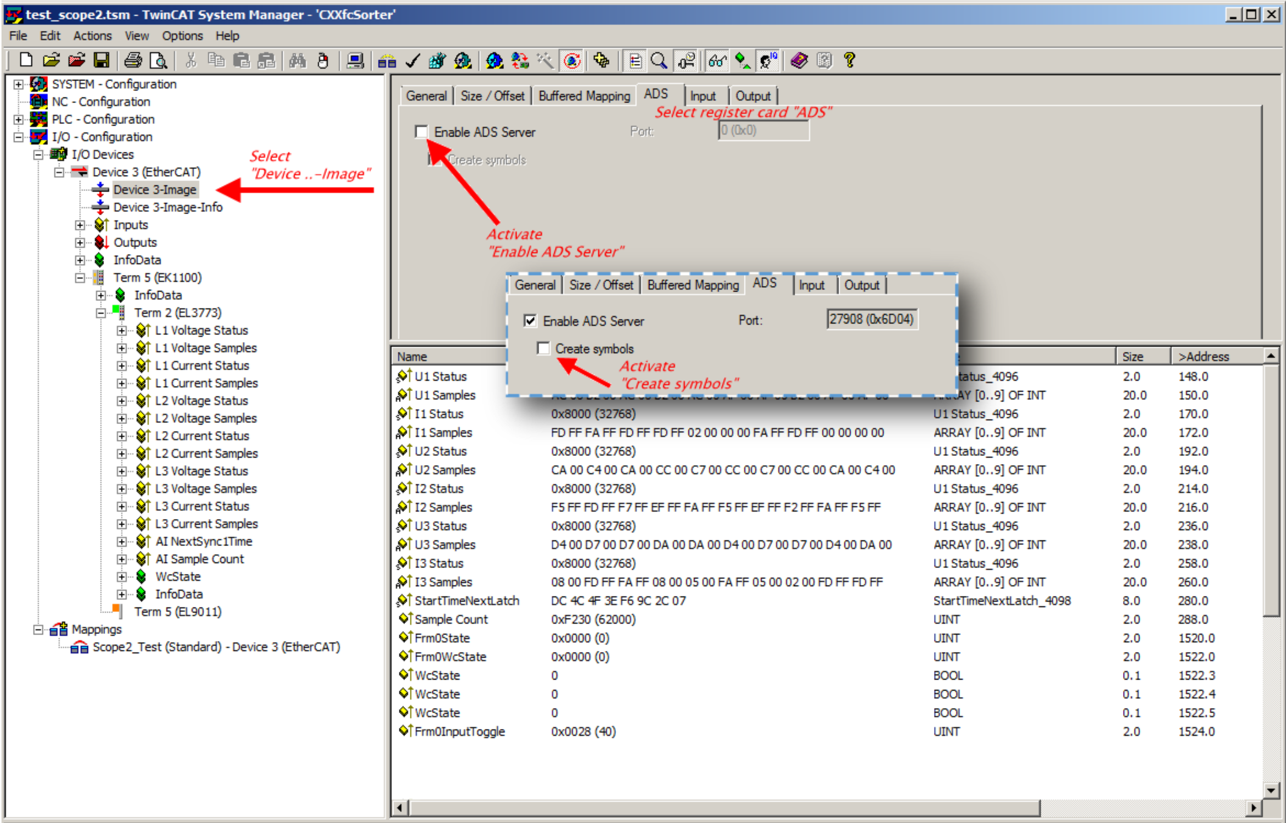


Fig. 169: Activation of the ADS server of the EtherCAT Device (TwinCAT 2)

The activation of the ADS server have to be carried out by selection of the “Device – Image” on the left sided configuration tree:
„I/O – Configuration → I/O Devices → Device .. (EtherCAT) → Device .. – Image“.

Next the register card “ADS” have to be selected to activate each checkbox „Enable ADS Server“ and „Create symbols“ then (the port entry is done automatically).

Thus, with the Scope2 process data can be accessed via the target browser without an embedded POU and without a variable reference respectively.

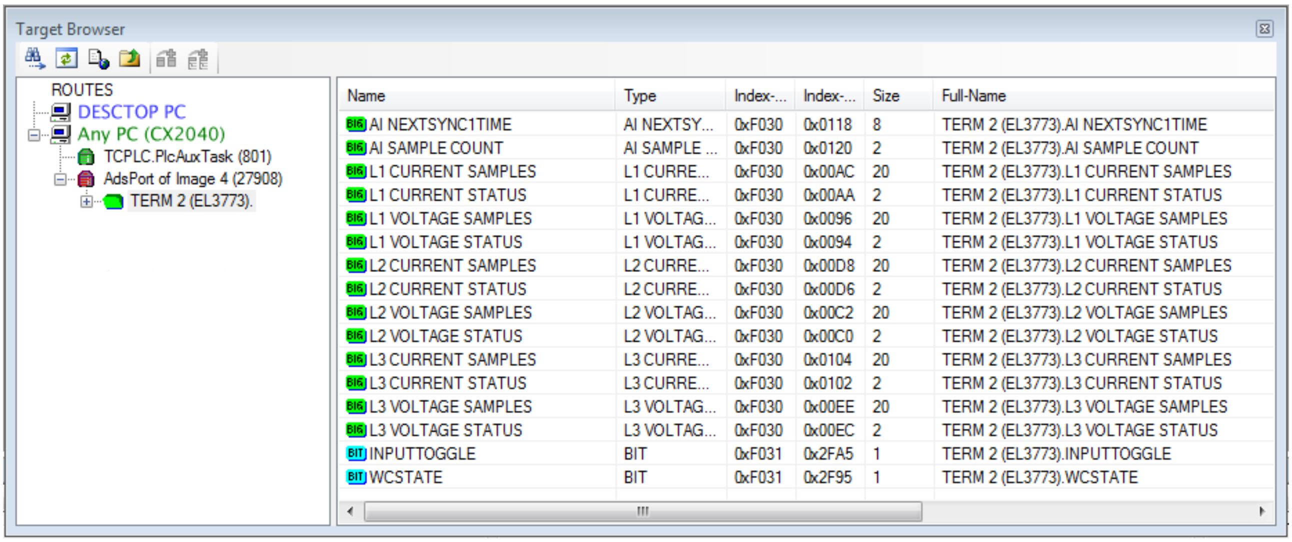
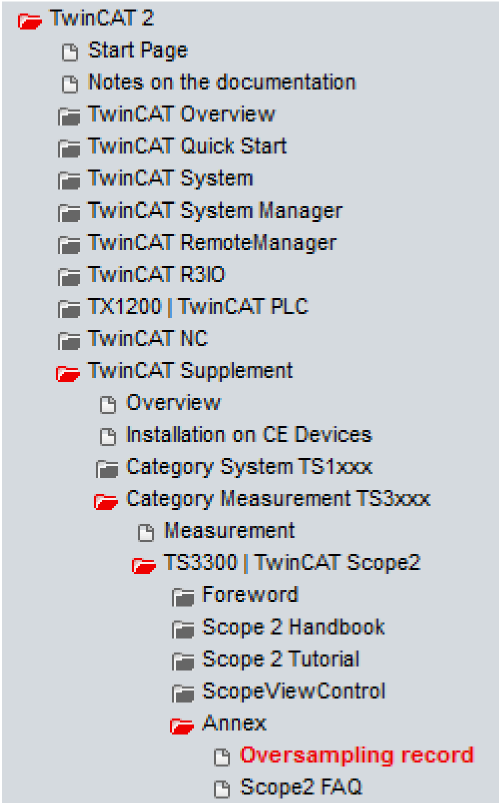


Fig. 170: Direct access of the Scope2 to the terminal's PDOs

Also see Beckhoff Information System

infosys.beckhoff.com → TwinCAT 2 → TwinCAT Supplement → Category Measurement TS3xxx → TS3300 | TwinCAT Scope 2 → Annex → Oversampling record:



Beckhoff TwinCAT supports the Scope2 with some oversampling devices in a special way by automatically calculating a special ADS array symbol in the background, which appears in the Scope2 in the variable browser. This can be then linked as a variable and automatically brings along the array information.

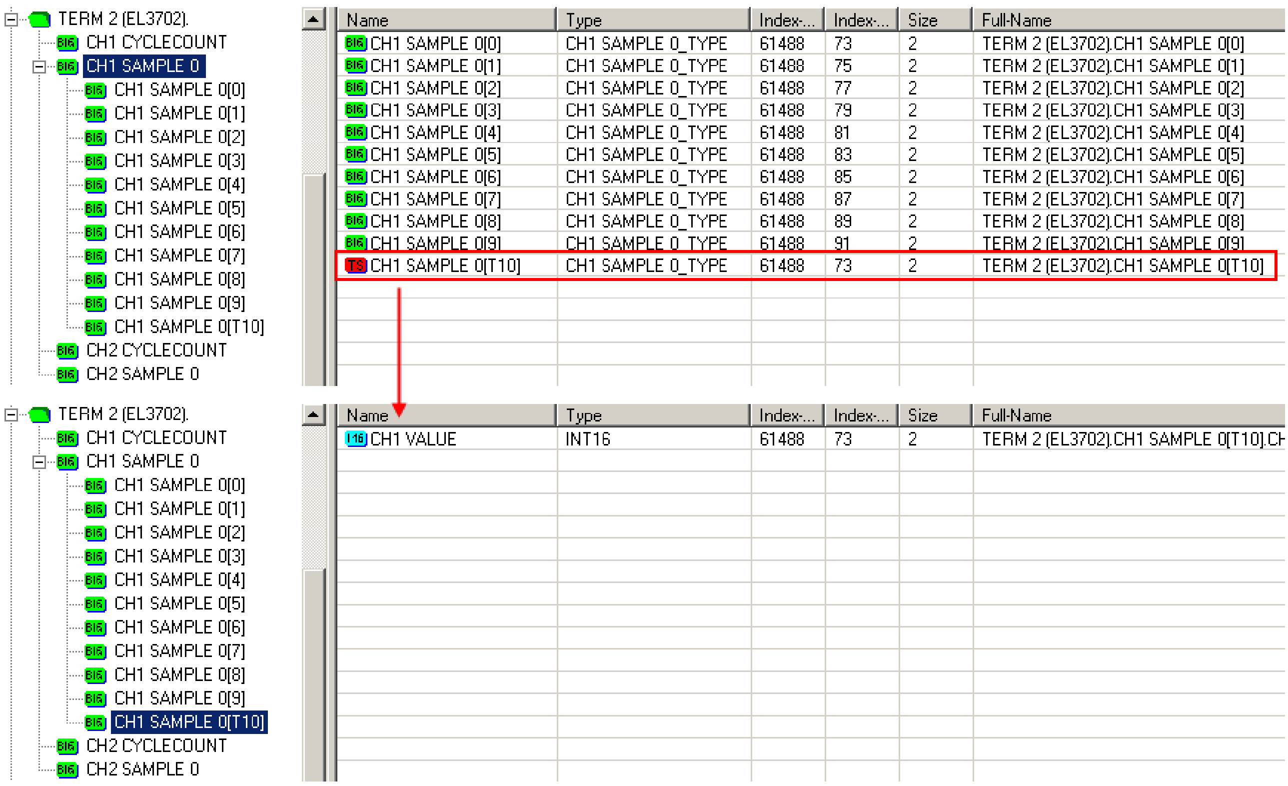


Fig. 171: Automatically calculated array variable (red) in the Scope2

Summary: an array variable has to be provided which is reachable via ADS. This can be a PLC variable of a POU or a defined array variable by the system manager or alternatively the ADS server of the device of the terminal or box is just activated. This is then detected by Scope2.

6.5 Notes on operation EL1262-0000, EL1262-0050, EL1264

Table of contents

Delivery state

No special settings are required when commissioning the EL126x for the first time. The EL126x operates as a normal 2/4-channel digital input terminal.

XML Device Description

If the XML description of the EL126x is not available in your system you can download the latest XML file from the download area of the Beckhoff website and install it according to the installation instructions.

Operating principle

The EL126x is a digital input terminal with 2/4 channels. It can read the voltage level not only cyclically with the EtherCAT cycle, but also several times in between. The distributed clocks support of the EL126x is used for this purpose. In the EL126x the ESC (EtherCAT Slave Controller) handles the data transmission to the EtherCAT fieldbus and supports the distributed clock functionality. This enables the ESC to read the inputs of the EL126x cyclically and equidistantly with high precision and store the values in the memory. When the EtherCAT frame fetches the data from the EL126x, a whole set of process data is ready for transfer. The inputs can be sampled with significantly higher frequency than the fieldbus cycle. Hence the term oversampling.

Distributed Clock

Distributed Clock Oversampling requires a clock generator in the terminal that triggers the individual data sampling events. The local clock in the terminal, referred to as distributed clock, is used for this purpose.

The distributed clock represents a local clock in the 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.

The EL126x only offers 32-bit support.

EtherCAT and distributed clocks

A basic introduction into EtherCAT and distributed clocks is available for download from the Beckhoff website.

Example:

The fieldbus/EtherCAT master is operated with a cycle time of 1 ms to match the higher-level PLC cycle time of 1 ms, for example. This means that every 1 ms an EtherCAT frame is sent to collect the process data from the EL126x. The local terminal clock therefore triggers an interrupt in the ESC every 1 ms (1 kHz), in order to make the process data available in time for collection by the EtherCAT frame. This first interrupt is called SYNC1.

As an example, the EL126x is set to oversampling n = 1000 in the TwinCAT System Manager, see *Oversampling factor selection dialog for the EL1262 in the TwinCAT System Manager*. This causes the ESC to generate a second interrupt in the terminal with an n-times higher frequency, in this case 1 MHz or 1 µs period. This interrupt is called SYNC0. With each SYNC0 signal the voltage is sampled as a digital value (0/1) and the corresponding values are sequentially stored in a buffer.

Generation of the SYNC0 pulse from the local synchronized clock within the distributed clock network ensures that the input values are sampled at highly equidistant intervals with the period of the SYNC1 pulse.

The maximum oversampling factor depends on the memory size of the used ESC and in the KKYY0200 version of the EL126x, it is n = 1000.

The values accumulated in the buffer are sent as a packet to the higher-level controller. For example, with 2 channels (EL1262) and n = 1000, 2 x 1000 bits = 2000 bits = 250 bytes of process data are transferred per EtherCAT cycle.

The oversampling factor of the EL126x can be set from 1 to 1000 in predefined values.

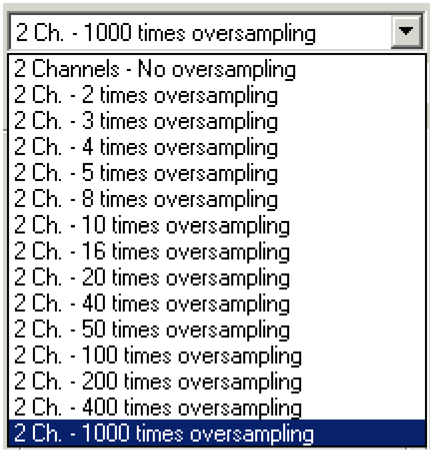


Fig. 172: *EL1262 oversampling factor selection dialog in the TwinCAT System Manager*

Please note that the EL126x process image characteristics change depending on the oversampling factor, see Process data [▶ 157] description.

Oversampling factor

Regarding the calculation of SYNC0 from the SYNC1 pulse based on manual specification of an oversampling factor, please note that for SYNC0 only integer values are calculated at nanosecond intervals. Example: 187.500 µs is permitted, 333.333 333 is not. Values other than those offered in the dialog are not possible. If implausible values are used, the terminal will reach the OP state, but its behavior will correspond to an oversampling factor of 1, and only the first bit will contain valid data. Example: For SYNC1/EtherCAT cycle = 1 ms oversampling factors such as 1, 2, 5 or 100 are permitted, but not 3.

Input characteristics

The input characteristics of the EL126x meet the requirements of EN61131-2:2003 Type 1.

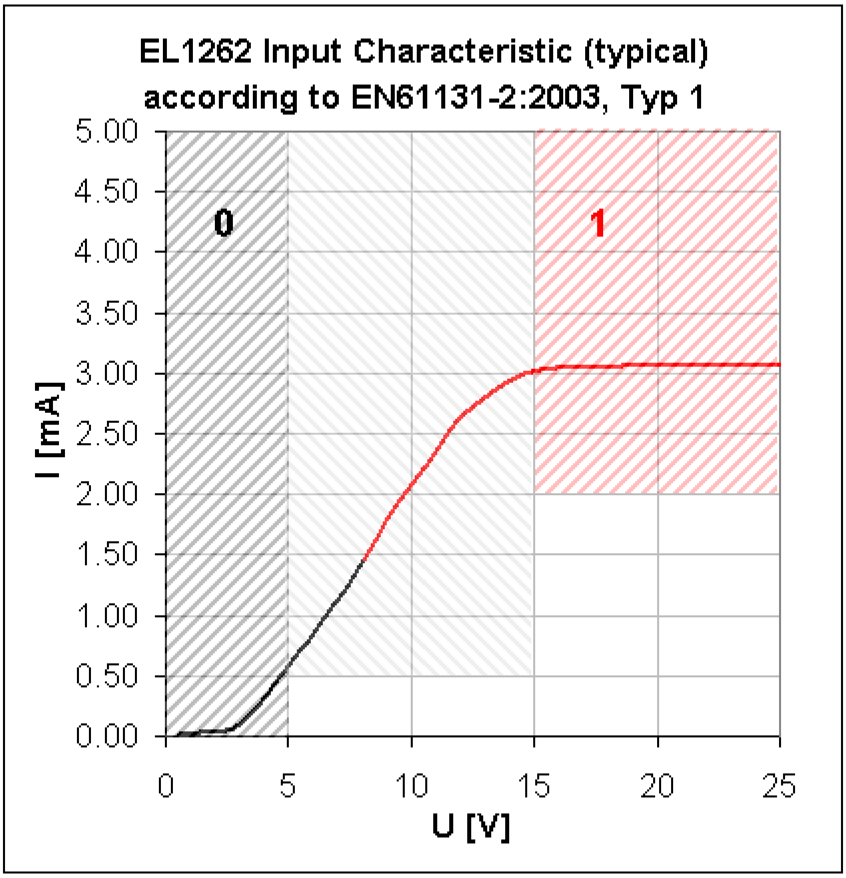


Fig. 173: Typical EL1262 input characteristics. Beckhoff reserves the right to make unannounced changes.

The input circuit of the EL12xx is optimized for fast signal changes and for the fastest possible signal acquisition. The duration required by a signal change (a rising or falling edge) to propagate from the terminal point at the front through to the logic of the central evaluation unit (ESC) is specified for the EL12xx series as T_{on}/T_{off} < 1 µs, for both rising (T_{on}) and falling edges (T_{off}). Because of this low absolute lead time, the temperature drift of the lead time is also very low.

It should be borne in mind that the input circuit does *not* include any filtering. It has been optimized for the fastest possible signal transmission from the input to the evaluation unit. Fast level changes or pulses in the µs range therefore reach the evaluation unit unfiltered or unattenuated. It may be necessary to use shielded cables in order to eliminate interference from the surroundings.

The sensor/signal transducer must be able to generate sufficiently steep signal edges. The power supply used should have sufficient buffer reserves to ensure that the signal reaches the terminal with a sufficiently steep edge in spite of capacitive or inductive cable losses.

Start-up behavior

From the start of the EtherCAT fieldbus, the EL126x requires around 60 bus cycles until it supplies continuous process data for the first time in the OP state.

Process data

The EL126x offers a range of process data for transmission,

Example EL1262: in the default state, the terminal is displayed in the System Manager as in *EL1262 default state*.

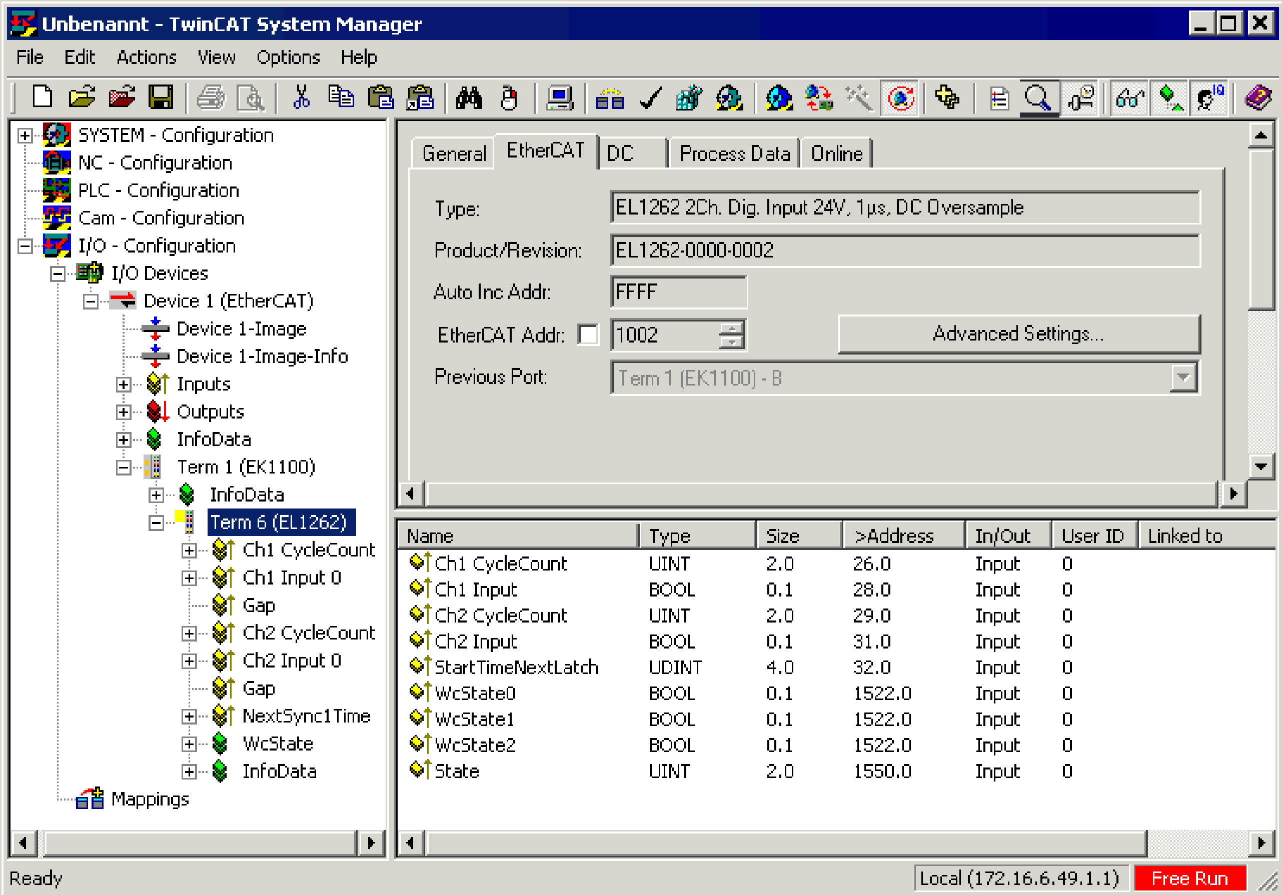


Fig. 174: *EL1262 default state*

- **Chx Cycle Count**
  EL1262 cycle counter: during each cycle the EL1262 increments this 16-bit counter by 1. The counter can be used to check the EL1262 for lost frames or data repetitions. The cycle counters for both channels show the same value.
- **Chx Input 0**
  Depending on the selected oversampling factor the digital input values are listed here, from 1 bit to 125 bytes per channel. The byte sequence corresponds chronologically to the ascending array index and the bits of a byte are read chronologically from right to left, i.e. from the least significant (.0) to the most significant (.7) bit
- **Gap**
  This variable is only used as a placeholder and does not represent a usable process data
- **NextSync1Time**
  As mentioned above the SYNC1 interrupt triggers provision of the accumulated process data in the EL1262 in synchrony with the fieldbus. The time of the SYNC1 interrupt is the same as the first SYNC0 interrupt, which determines sampling of the inputs. The NextSync1Time value transferred by the EL1262 during an EtherCAT cycle is the start value for the *next* SYNC1 interrupt with a resolution of 32 bit (see Distributed Clocks [▸ 155]). The NextSync1Time process data can be deactivated in the ProcessData tab. NextSync1Time can be used to specify the read time for each individual sample within the distributed clock accuracy.

Chx Input presentation

The Chx Input process data must cover a large range of values from 1 to 1000 bits. In order to maintain a clear display of the configuration tree and the task variable links, the Chx input variables are shown either as bit or byte. Oversampling factor <= 100: individual bits are displayed.
Oversampling factor > 100: bits are consolidated as bytes.
The task receiving the EL126x process data therefore has to offer bit or byte arrays as appropriate.

Tips for operation

Distributed Clocks settings

In the advanced settings of the EL126x for the distributed clocks, the time of the SYNC1 interrupt can be shifted forward slightly, see example *advanced settings EL1262, Distributed Clocks*. By activating the "Based on Input Reference" checkbox the SYNC1 interrupt is shifted forward by a few µs. For further information please refer to the Distributed Clock system description [▶ 155].

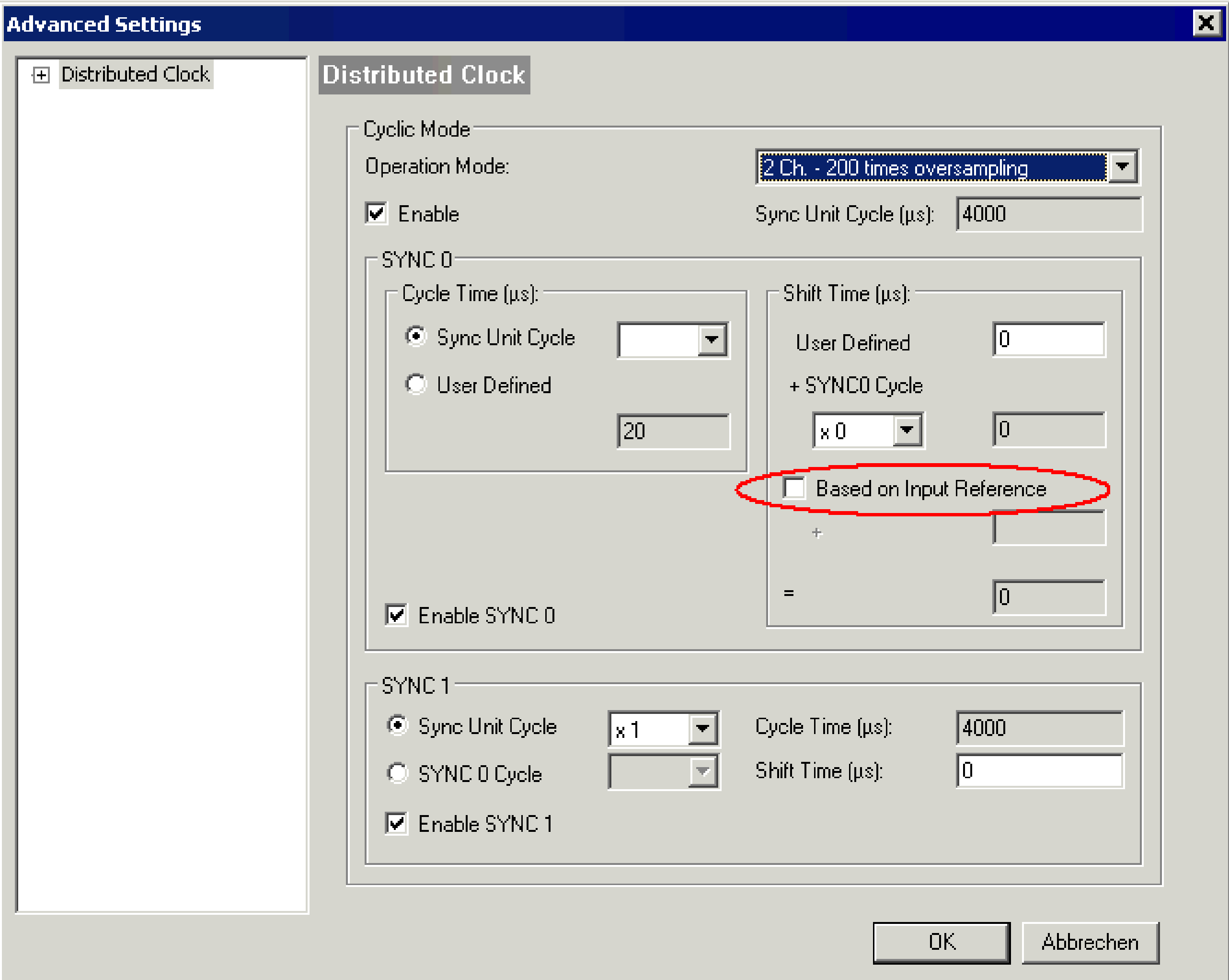


Fig. 175: EL1262 Advanced Settings, Distributed Clocks

Linking large variables

The option “Change Multi Link” can be used to link larger memory areas with continuous variables. Proceed as follows:

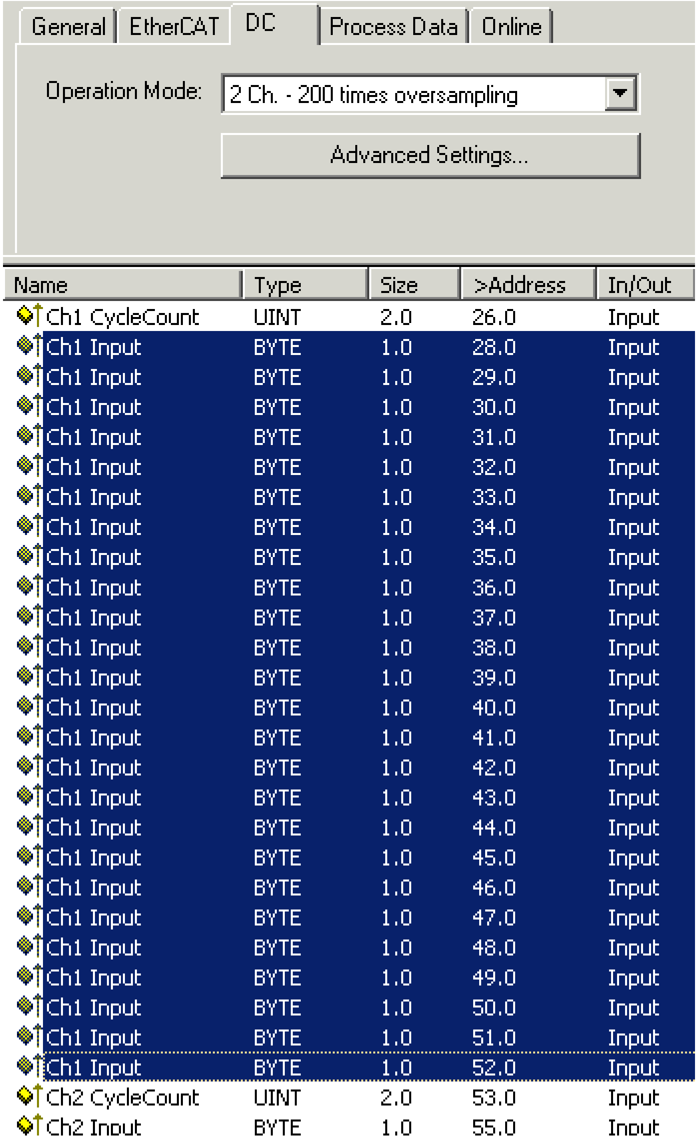


Fig. 176: *Select the variables in the terminal with the mouse*

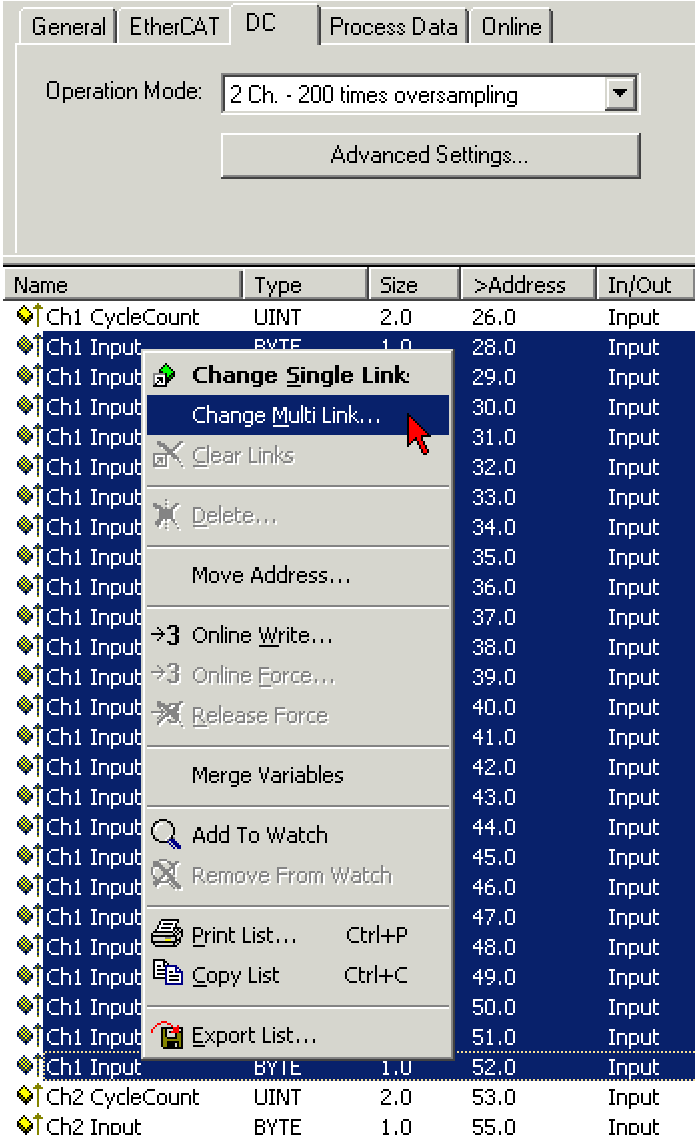


Fig. 177: *Right-click, Change Multi Link*

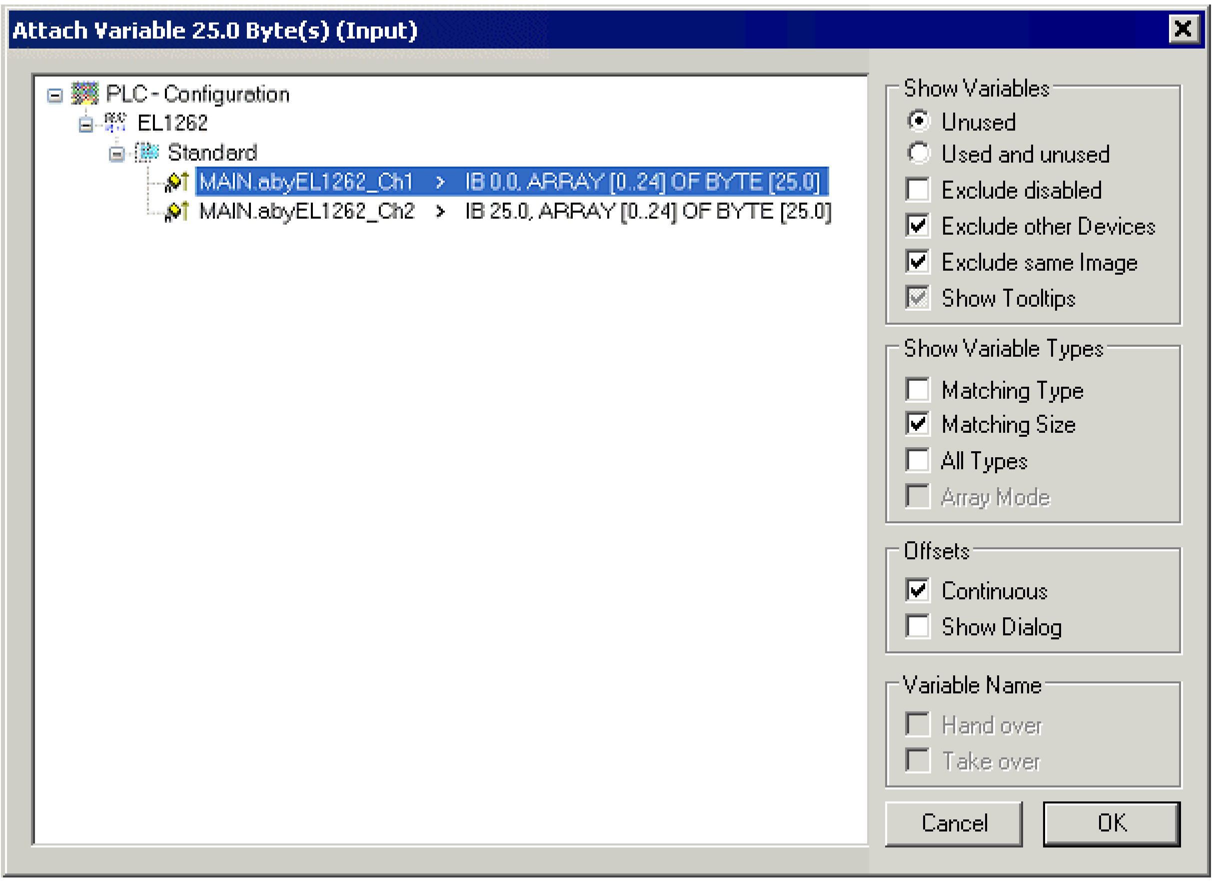


Fig. 178: *Select the variable range from the task*

6.6 Notes on operation EL1262-0010

Basic instructions

- Wording: "Input" in the following means the digital input of the terminal, i.e. as input from the controller's point of view. In the following, "output" means the digital output of the terminal, i.e. the output from the controller's point of view.
- The EL1262-0010 can work with up to 10,000,000 samples (bits) per second and channel. This corresponds to an effective 20 Mbit per second per terminal and thus to a calculated 20% utilization of a 100 Mbit EtherCAT segment.
- Conditions of use
    - TwinCAT 3.1 Build 4024.55 or higher, e.g. Build 4025/26
      Note: Observe build setting
    - from Firmware FW02
    - from Revision -0001

Fig. 179: TwinCAT 3.1 Build 4024.55 or higher

After scanning or manually inserting the terminal in TwinCAT 3.1, it responds with

- 80x oversampling on channel 1+2

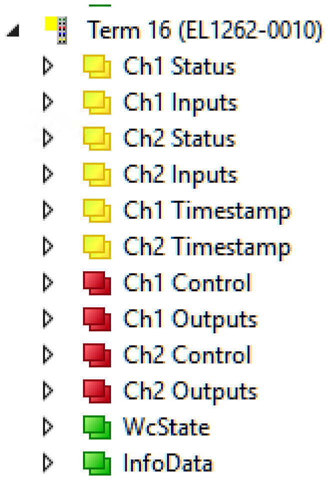


Fig. 180: EL1262-0010 in the TwinCAT tree

- RS422 setting in CoE on both channels

8000:0	Settings Ch.1	RW	> 2 <	
8000:01	Setting	RW	RS422, Enddev., term. and Rx Bias/Failsafe (0)	
8000:02	Timestamp correction	RW	0	ns
8001:0	Manual settings Ch.1	RW	> 12 <	
8010:0	Settings Ch.2	RW	> 2 <	
8010:01	Setting	RW	RS422, Enddev., term. and Rx Bias/Failsafe (0)	
8010:02	Timestamp correction	RW	0	ns
8011:0	Manual settings Ch.2	RW	> 12 <	

Fig. 181: RS422 setting in the CoE

6.6.1 Configuration/ electrical settings

The first step is to make the electrical setting in CoE 0x80n0:01 for each channel according to the intended use (channel 1: n=0, channel 2: n=1).
Each channel can operate in either 5 V single-ended or RS mode.

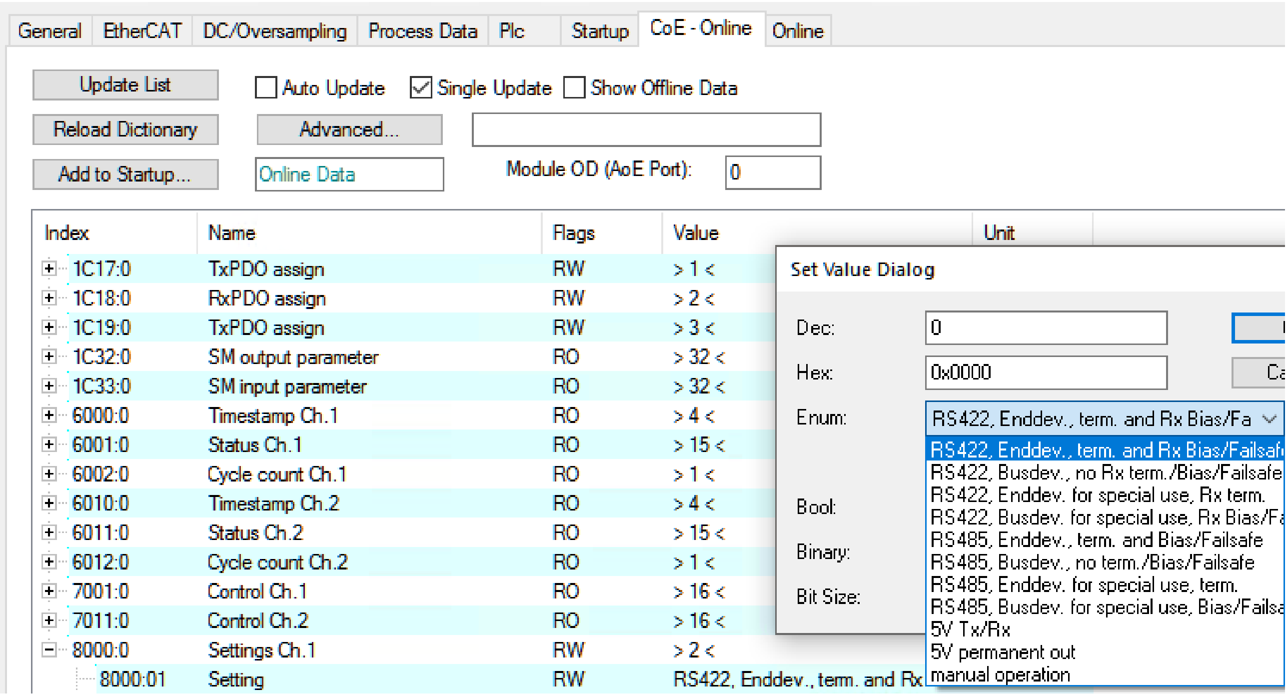


Fig. 182: Setting "CoE- Online" tab, RS or single-ended operation

1. RS422/RS485 operation mode: the RS422/485 driver is electrically active on this channel. Termination resistor and bias resistors (also called: failsafe) are switchable per channel. Four common configurations can be selected here. It should be noted that RS422/485 is a differential signal between Tx+/Tx- or Rx+/Rx-. In general, operation with "Enddev., term. and Rx Bias/Failsafe" is successful with short cable lengths and only one remote terminal. Checking the electrical signal quality (crosstalk, rise times) by means of a suitable measuring instrument (e.g. oscilloscope) is recommended.

- RS422: bidirectional operation, input and output can be used independently of each other

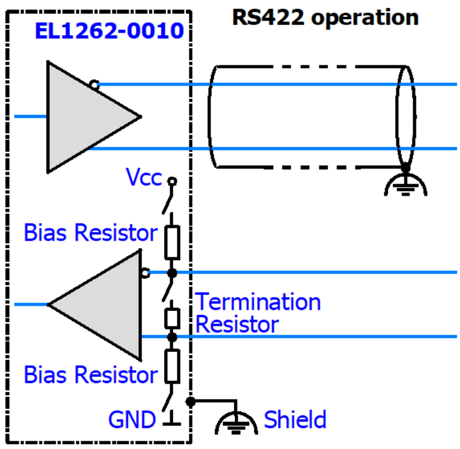


Fig. 183: RS422: bidirectional operation

- RS485: unidirectional operation e.g. for serial communication, input and output are internally already connected

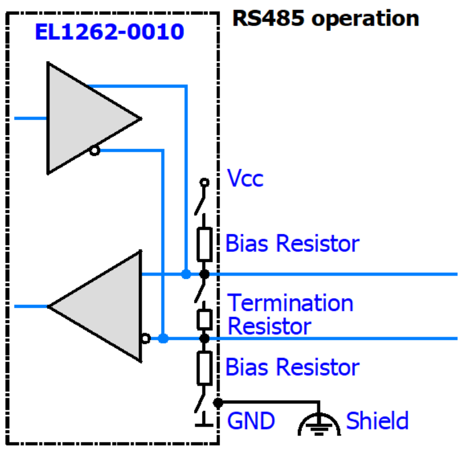


Fig. 184: RS485: unidirectional operation

2. Operation mode "5 V Tx/Rx": the 5 V input/output is electrically active on this channel.

3. Operation mode "5 V permanent out": useful for test purposes or for supplying a remote terminal. A "5 V Tx/Rx" operation is then not possible.
If the 5 V output is to be used permanently for device supply simultaneously with RS operation, this must be set via "manual operation".

4. Operation mode "manual operation": if selected, the individual functions of the terminal can be enabled individually via 0x80n1

8001:0	Manual Settings Ch.1	RW	> 12 <
8001:01	Differential Rx enabled	RW	TRUE
8001:02	Differential FullDuplex enabled	RW	TRUE
8001:03	Differential enabled	RW	TRUE
8001:04	Differential Termination enabled	RW	TRUE
8001:05	Differential Termination Tx enabled	RW	FALSE
8001:06	Differential Tx Filter enabled	RW	FALSE
8001:07	Differential Rx/Tx Shutdown enable	RW	FALSE
8001:08	Differential Enable Bias/Failsafe Tx	RW	FALSE
8001:09	Differential Enable Bias/Failsafe Rx	RW	TRUE
8001:0A	5V SingleEnded Rx enable	RW	FALSE
8001:0B	5V SingleEnded Tx enable	RW	FALSE
8001:0C	5V permanent enable	RW	FALSE

Fig. 185: Activation of functions in CoE object 8001 in operation mode "manual operation"

6.6.2 Process data (functions and oversampling)

1. Functions

- **Input function**: each channel of the terminal samples its input (RS or 5 V) with the set oversampling factor n (i.e. n times faster than the EtherCAT cycle time) and delivers it as bit data stream in the process image e.g. 0011001001001... via EtherCAT to the controller/PLC. The bits are combined to bytes for handling reasons.
- **Output function**: each channel of the terminal expects a bit data stream e.g. 01001000101... from the controller/PLC and outputs it via its output (RS or 5 V) after the set oversampling factor n (i.e. n times faster than the EtherCAT cycle time).

2. The oversampling and thus the extent of process data must be set with the TwinCAT DC dialog (see also chapter Make oversampling setting [▶ 171]).

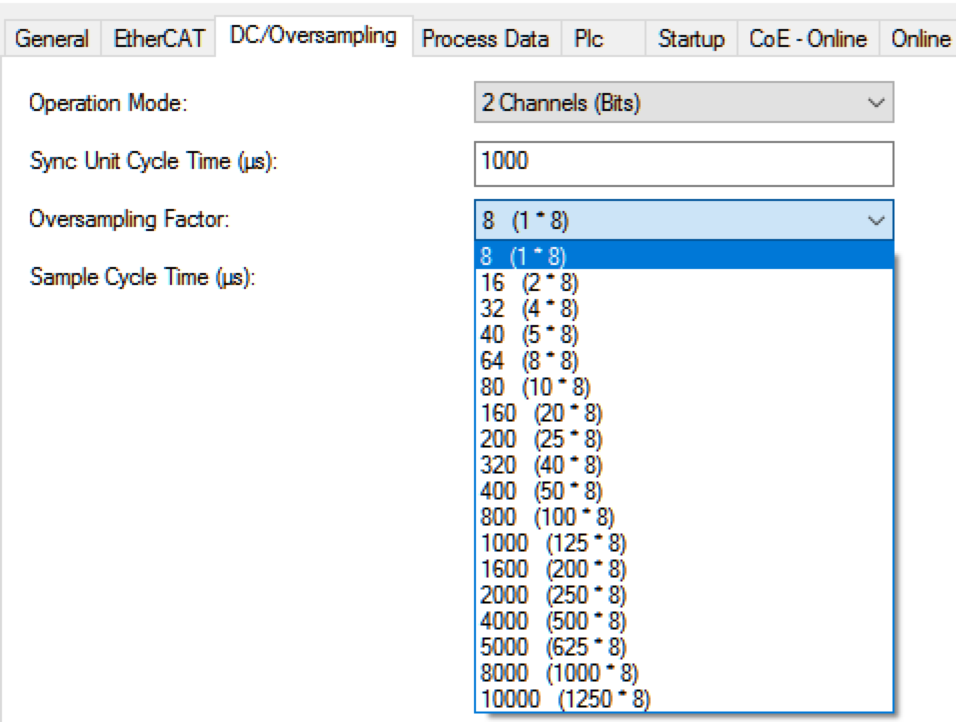


Fig. 186: Settings "DC/Oversampling" tab

Rules:

- The oversampling factor must be the same for input and output (if both are used)
- The oversampling factor must be the same for channel 1 and 2 (if both are used)
- The resulting sampling time from EtherCAT cycle time and oversampling factor must correspond to the following formula: 100 ns + n * 20 ns ; the sampling times from the minimum are therefore 100 ns, 120 ns, 140 ns ...

For easy linking of the individual PDO variables, it is recommended to activate the *Show Sub items* function, then the PDOs are "expanded" and can be linked individually:

Fig. 187: Show sub items" button

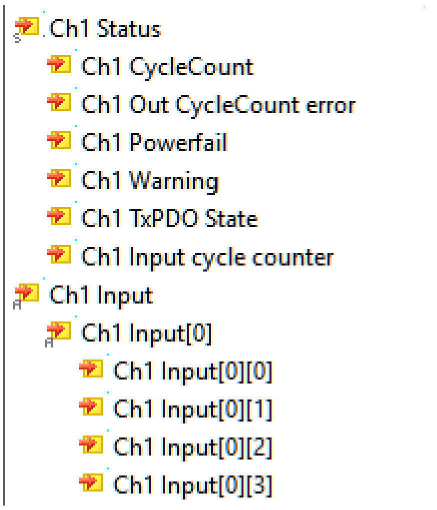


Fig. 188: PDO tree

3. Manual PDO configuration

An individual PDO configuration, e.g. operation with input channels only or with channel 1 only, can be set manually via the *PDO Assignment* dialog:

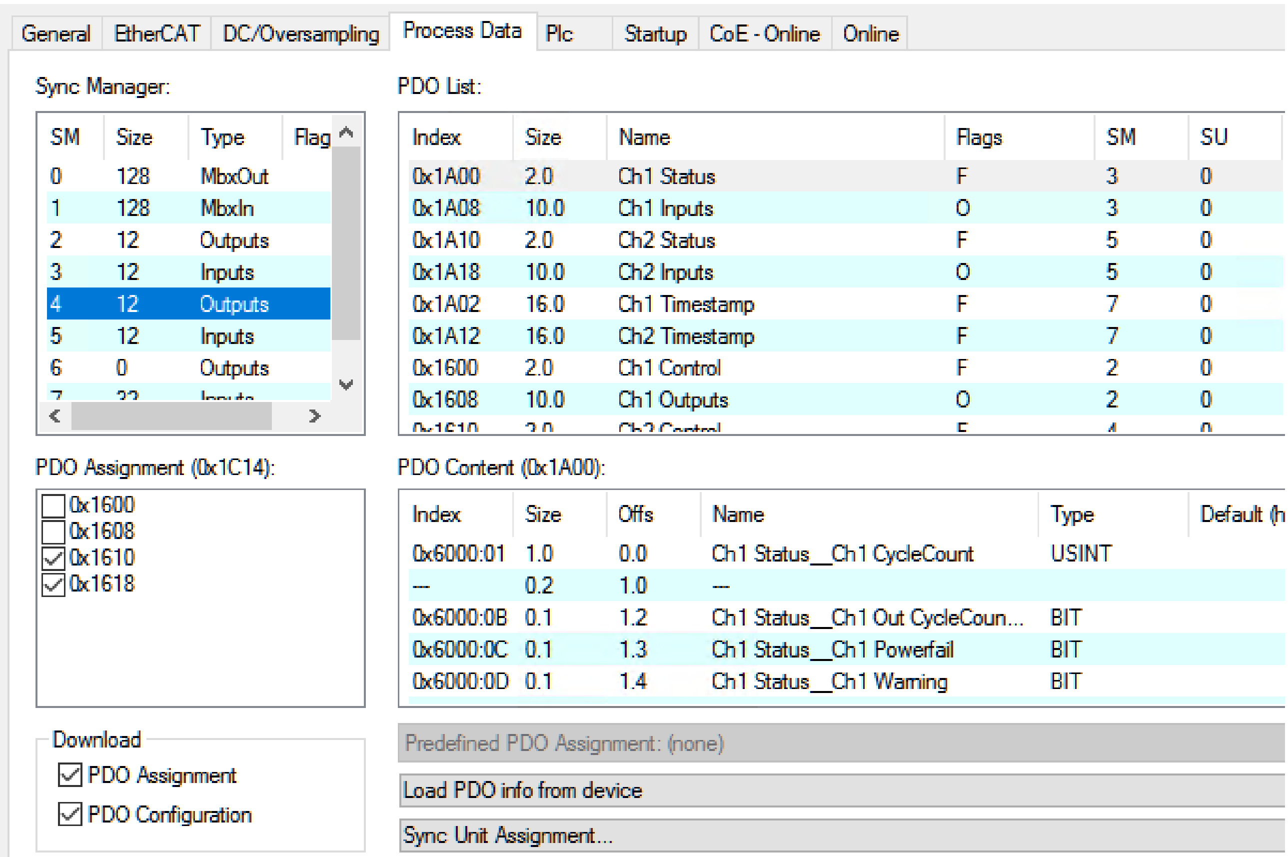


Fig. 189: "Process data" tab, PDO assignment

Procedure as follows:

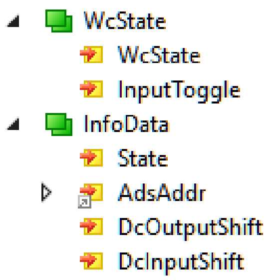
- set the desired oversampling factor in the *DC/Oversampling* dialog, all PDOs are then activated
- in the *SyncManager* dialog, deselect the unwanted PDOs:

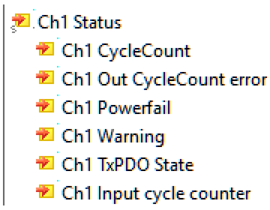
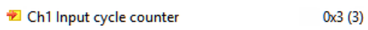
0,1: Mailbox, no change possible
2: Outputs channel 1
3: Inputs channel 1
4: Outputs channel 2

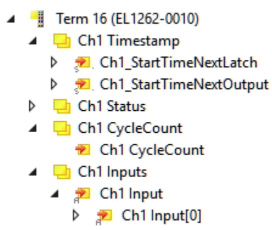
5: Inputs channel 1
6: empty, not to be used
7: Timestamp channel 1+2

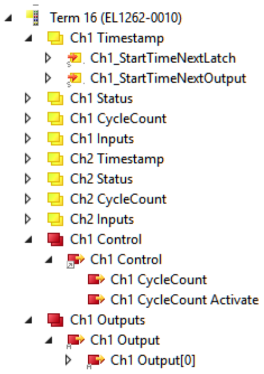
6.6.3 Interpretation of the process data (PDO)

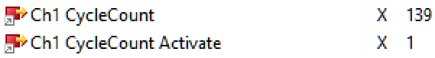
The process data of the terminal can be assigned to four areas (device information, channel information, use as digital input, use as digital output):

Device information  	WcState  	"0": proper operation "1" : the terminal has not exchanged any process data in this cycle
	InputToggle  	Changes cyclically 0/1/0/1...: proper operation Changes acyclically 0/1/1/0/... : the terminal has not exchanged any process data in this cycle.
	State  	Bit 3 =TRUE: "0": proper operation
	AdsAddr	AMS NetId and port number of the terminal, important for acyclic communication from the controller with the CoE and registers.
	DcOutputShift	offset of the output terminals set by TwinCAT in [ns] to the current task tick. TwinCAT generally displays this PDO, regardless of whether the device has input or output functions. Note: the EL1262-0010 operates as output terminal in the sense of this setting.
	DcInputShift	offset of the input terminals set by TwinCAT in [ns] to the current task tick. TwinCAT generally displays this PDO, regardless of whether the device has input or output functions. Not relevant for the El1262-0010.

Channel information, each for channel 1 and 2  	Ch CycleCount	Byte	iterating per cycle [0..255], can be used on the controller side to check that the terminal supplies new data in each EtherCAT cycle, expected value is therefore +1
	Ch Out CycleCount error	Bool	Feedback channel of the corresponding output. "0": no transmission error "1": Output channel has detected count > +1
	Powerfail		"0": proper operation "1": internal error, contact Service [▸ 211]
	Warning		"0": proper operation "1": internal warning, possibly overheating
	TxPDO		"0": proper operation "1": the EtherCAT data exchange with the channel has failed in this cycle
	Input cycle Counter  		iterating per cycle 0→1→2→3→0...: proper operation not '+1'-iterating, e.g. 0→2→2→3→0…: The EtherCAT frame has not exchanged data with the slave in time

Use as digital input  The process data are structured channel-oriented for control-side linking with a specific FB, therefore the DC timestamp is offered twice	StartTimeNextLatch	64-bit DC timestamp [ns] of the first sample of the next oversampling package
	CycleCount  	counts cyclically +1: proper operation cyclic change +0 or +2: if necessary, the sync time of the terminal must be shifted into a jitter-free range by changing the ShiftTime, where the data transfer from/to the EtherCAT frame is safely possible
	Input[n]	Bytewise input process data for control; Process data size depending on the set oversampling

<table>
<tr><td>Use as digital output


The process data are structured channel-oriented for control-side linking with a specific FB, therefore the DC timestamp is offered twice</td>
<td>StartTimeNextOutput</td>
<td>64-bit DC timestamp [ns] of the first sample of the next oversampling package</td></tr>
<tr><td></td><td>CycleCount

</td>
<td>8-bit counter that can be used cyclically by the controller with +1. In case of operation, both channel counters must be used!
See Explanation of CycleCounter monitoring [▶ 175]</td></tr>
<tr><td></td><td>CycleCount Activate</td>
<td>activates the monitoring on '+1' of the CycleCount in the terminal.
See Explanation of CycleCounter monitoring [▶ 175]</td></tr>
<tr><td></td><td>Output[n]</td>
<td>Byte-wise output process data from the controller; the process data size depends on the set oversampling</td></tr>
</table>

6.6.4 Make oversampling setting

1. The terminal supports oversampling factors from 8-fold to 10,000-fold for both inputs and both outputs. Conditions: The oversampling factor must be the same for both channels and for input and output.

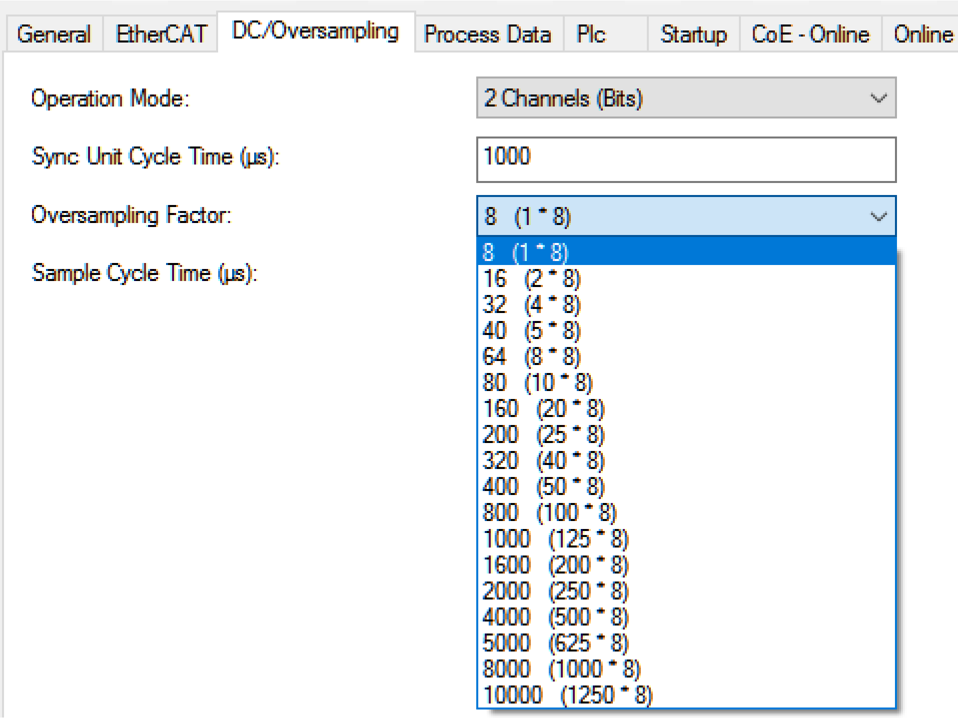


Fig. 190: DC/oversampling setting

Note: the oversampling factor must be selected at the EtherCAT cycle time so that the divisor in [1 ns] remains an integer.

2. Examples:

- Cycle 100 µs, OVS = 32 → 3.125 µs = OK
- Cycle 666.6 µs, OVS = 32→ 20.0909... µs = NOK

TwinCAT usually does not display non-executable constellations (here: OVS = 32) at all:

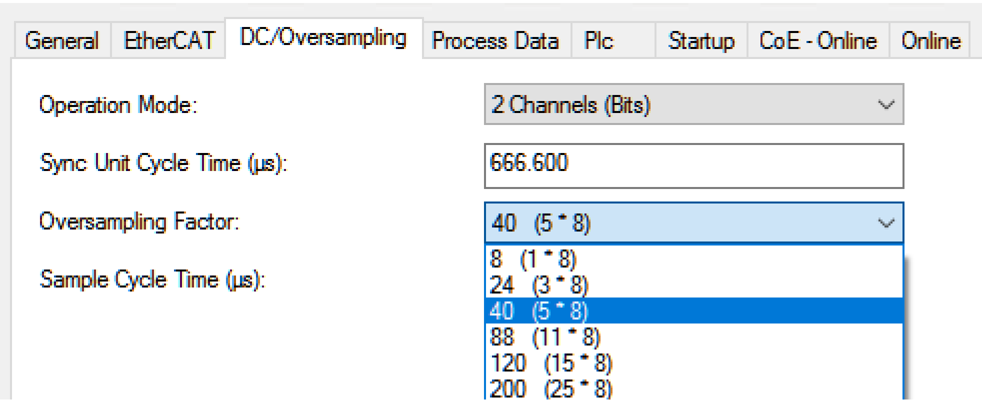


Fig. 191: Suitable representation Oversampling factor

3. For linking with the PLC a byte array is suitable. For example, a byte array of size 1250 for OVS = 10,000:

```
aMassiveOutput AT %Q*:ARRAY[1..1250] OF BYTE;
```

The linking in TwinCAT 3 can take place, for example, under the selection of "Matching Size":

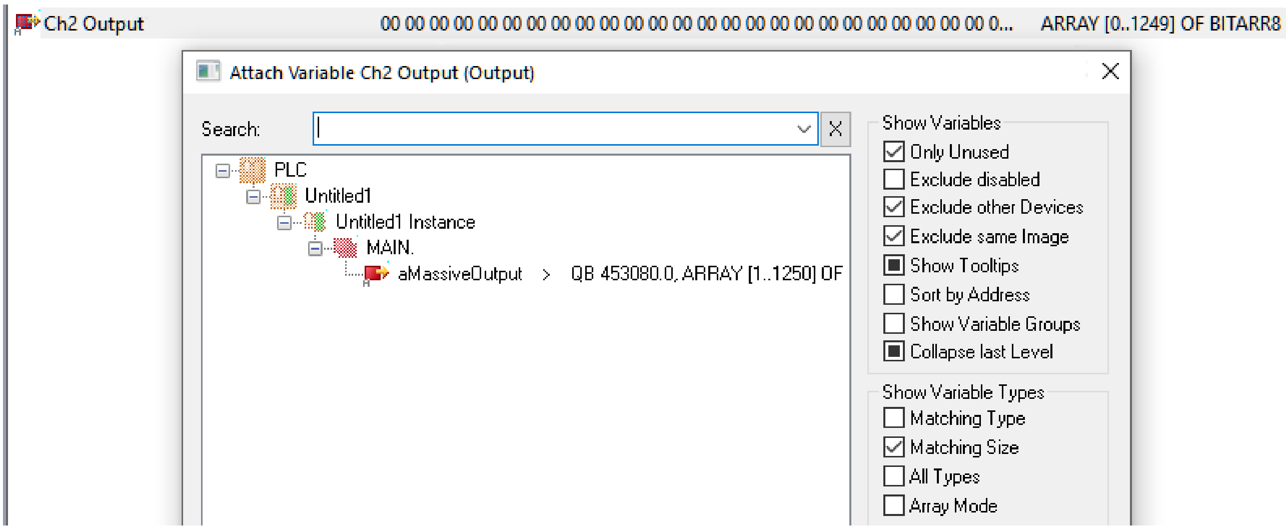


Fig. 192: Linking by "Matching Size" variable type

It should be noted that the byte sequence corresponds to the ascending array index in *terms of time* and the bits of a byte are output from *right to left*, i.e. from the least significant (.0) to the most significant (.7) bit: the byte 2#10100001 would therefore be visible on the oscilloscope (1 V/digit, 1 µs/digit, time axis from left to right) as 10000101 (blue line):

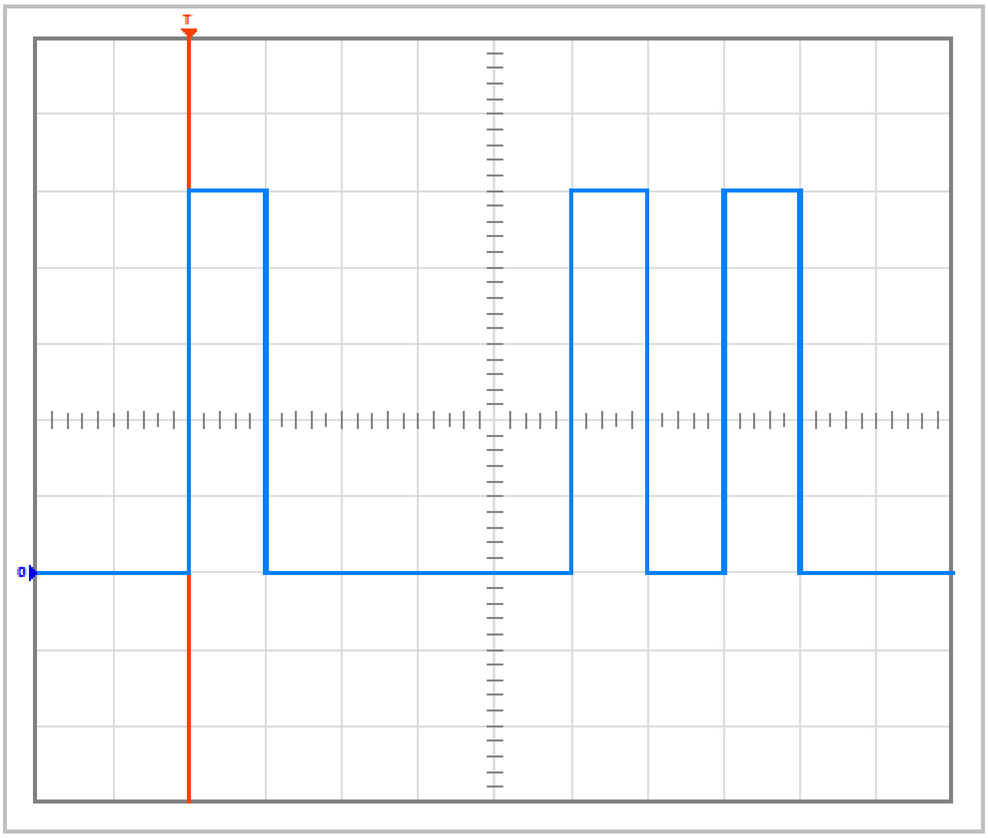


Fig. 193: Oscilloscope display

6.6.5 Adjust DistributedClocks if necessary

The EL1262-0010 controls the input and output operation (reading or writing the PDO) via a single internal trigger, the ESC Sync0, so they cannot be controlled independently in terms of time. The ShiftTime setting (both via the EtherCAT master and via the slave's own dialog) always affects the entire terminal (input and output).

Note for the EtherCAT master DC-ShiftTime dialog:

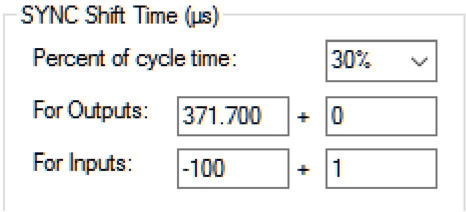


Fig. 194: Example values DC-ShiftTime dialog

The EL1252-0010 is configured as a "DC Output Device" via ESI, "For Outputs" must be used if necessary.

The time shift of the input/output process of the EL1262-0010 by DcShiftTime is of practical importance in 2 cases:

- The temporal cooperation "simultaneity" of the EL1262-0010 with other DC devices is to be changed (in relative terms, i.e. output/read earlier or later)
- there are data transmission losses (recognizable e.g. in jumps <> + 1 in the CycleCounter) because the time of passage of the EtherCAT frame in the EL1262-0010 overlaps with the PDO data transfer in the ESC

For this reason, it is necessary to proceed as follows:

1. The terminal is only operated with inputs

- Change DcShiftTime
    - for all devices in the EtherCAT segment (rather unusual):
      Enter the desired value for "For Outputs" in the EtherCAT master dialog:

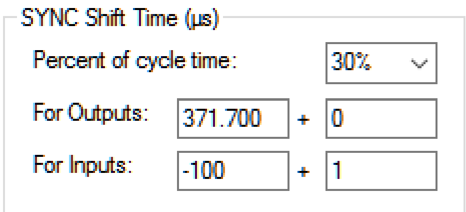


Fig. 195: Change SYNC shift time

    - - For the individual terminal:

"User Defined ShiftTime" for SYNC0 in [µs] with (decimal values possible) change as required -> this has a direct effect on the read time, i.e. the start of sample = and thus the value in PDO *StartTimeNextLatch-*

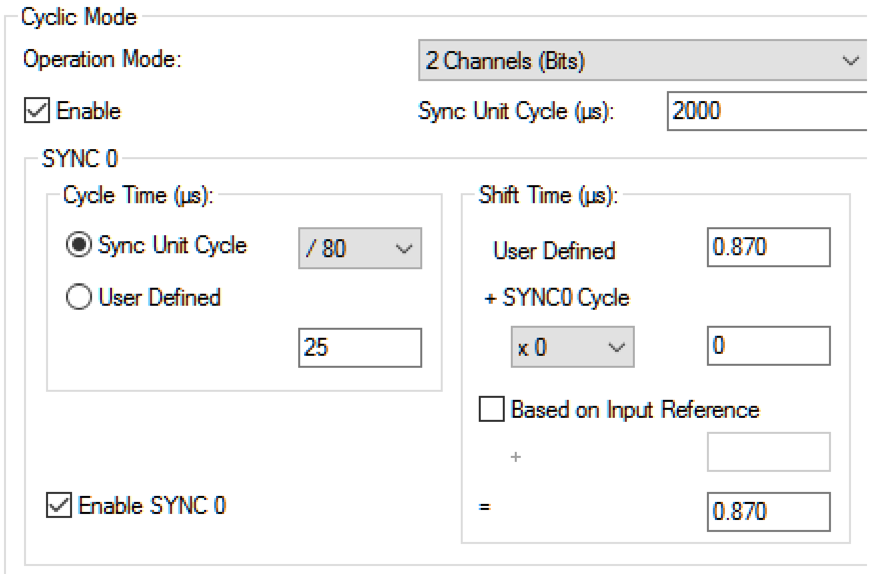


Fig. 196: Change User Defined Shift time

- If external delays are to be compensated, C0E x80n0:02 "Timestamp correction" can also be written in [ns], in which case the value in PDO *StartTimeNextLatch* is offset (added or subtracted) against this entry in each cycle . The electrical read time does not change, only the transmitted timestamp is manipulated.

2. The terminal is only operated with outputs

- Change DcShiftTime
  - For all devices in the EtherCAT segment (rather unusual): see for inputs
  - For the individual terminal: see for inputs
- Separate manipulation of the output time is not possible

3. The terminal is operated with inputs and outputs

First set the output as above, then set the input timestamp using "Timestamp correction".

Transfer of the DC timestamp

The transmission of the DC timestamp StartTimeNextLatch/StartTimeNextOuput offers the possibility of processing the process data on a "real" time basis rather than a cycle basis. This requires additional code on the controller side, which has already been created by Beckhoff in the case of the TwinCAT 3 XFC library or NC/cam controller and must otherwise be created on the application side.

Also see about this

- Explanation of CycleCounter monitoring [▸ 175]
- Explanation of the output behavior in case of error (Watchdog, CycleCounter) [▸ 177]
- Interpretation of the process data (PDO) [▸ 169]

6.6.6 Output behavior in the event of an error

To set the output behavior in the event of an error (no cycle-current data or watchdog), see "Explanation of the output behavior in case of error (Watchdog, CycleCounter) [▸ 177]".

6.6.7 Recommended diagnostics in the controller/PLC

For this high performance terminal it is particularly important that it is supplied with current output data or can deliver its input data in every EtherCAT cycle. Therefore it is recommended to check the following PDO continuously (in each real-time cycle) after TwinCAT has been started in RUN to ensure proper operation:

- Device
  - State
  - WcState
  - InputToggle
- Channel
  - Powerfail
  - Warning
  - TxPDO State
  - Input cycle Counter
- Related to the inputs:
  - CycleCount:
- Related to the outputs: see "Explanation of CycleCounter monitoring [▸ 175]"

For the respective meaning/setpoints, see the explanations of the PDOs in the previous chapter Interpretation of the process data (PDO) [▸ 169].

In the event of a problem, suitable settings may need to be adjusted, e.g. PLC real-time load, DC ShiftTime, oversampling factor.

6.7 Sensitivity of the input

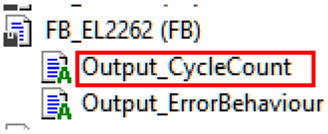
The input circuit of the EL12xx is optimized for fast signal changes and the shortest possible signal detection. The time required for a signal change as a rising/falling edge from the terminal point at the front of the terminal to the logic of the central processing unit (ESC) is specified for the EL12xx series at T_{ON}/T_{OFF} < 1 µs, both for rising (T_{ON}) and falling edge (T_{OFF}). Due to this low absolute cycle time, the temperature drift of the cycle time is also very low.

It should be considered that the input circuitry has little or no filtering, depending on the type. It is optimized for fastest signal transmission from the input to the evaluation unit. Fast level changes/pulses in the µs range, e.g. due to possible EMC influences, thus arrive at the evaluation unit unfiltered/undamped and are possibly visible as a change of state of the input.
If necessary, shielded cables should be used to exclude environmental influences.

6.8 Explanation of CycleCounter monitoring

The terminal must be enabled to recognize whether it has been cyclically supplied with current data. For this PDO CycleCount Activate = TRUE and the PDO CycleCount are to be used cyclically +1. In case of a missing data packet (e.g. by LostFrame) or in case of a frame repetition the terminal increments (+1) a local 8-bit error counter (0...255) in the ESC register 0x030D. By occasionally reading this counter (and setting it to zero if necessary), the application can thus determine, e.g. every second, whether the terminal has detected transmission errors. The reading and writing of the registers must be done by ADS (acyclic communication). If necessary, the sync time of the terminal is to be shifted into a jitter-free range by changing the terminal shift time, where the data transfer from/to the EtherCAT frame is safely possible.

See the TC3 **sample program** https://infosys.beckhoff.com/content/1033/el126x/Resources/13740301195.zip in FB_EL2252 in the action *Output_CycleCount*, it demonstrates the regular reading of register 0x030D and cumulating the values.



6.9 Explanation of the output behavior in case of error (Watchdog, CycleCounter)

Behavior in case of error - Watchdog and CycleCounter

This output terminal depends on regular supply of cycle updated data from the controller. If this is not the case, two phases are to be distinguished and the following tools take effect

- Short absence (one cycle and more) → CycleCounter monitoring
  In CycleCounter monitoring, the device wants to be assigned cyclically incremented values (+1) from the controller in the CycleCounter process data variables. Thus, it can determine whether a frame repetition or a LostFrame has occurred based on the consecutive values. If CycleCounter monitoring is not activated, the firmware is unable to detect the non-arrival of data and repeats the data of the last sample until the watchdog has run out.

  Diagnosis: If no increment +1 is detected by the device, at least the internal register $030D_{hex}$ "PDI-Error" counts up by 1. A counter overflow 255 → 0 is not interpreted as an error. If the counter increment is greater than 1 once and subsequently 1 again, only one PDI error is output, since all the counter values after the event have an increment of 1. This register can be read out acyclically, e.g. by the PLC, see "Explanations on CycleCounter monitoring".

  Output behavior: Furthermore, in this case the firmware continues to use the output according to the ESC register setting:

  - for channel 1: Register 0x0F00
    Bit 0 = TRUE → Parameterization is activated
    Bit 1 to 3: Default value for behavior at CycleCounterError, see below.
  - for channel 2: Register 0x0F01
    Bit 0 = TRUE → Parameterization is activated
    Bit 1 to 3: Default value for behavior at CycleCounterError, see below.
  - Note: The two byte registers 0x0F00 and 0x0F01 should be written simultaneously as 1 word access.
  - Possible register values for the CycleCounter behavior are (names are taken from the demo program below):
    ZERO: "000": A logical zero "0" is output (default)
    ONE: "001": A logical one "1" is output
    HOLD: "010": The value of the last bit in the previous cycle is output
    CONTINUE: "011": A PDI error is output, but the data currently in the buffer is output. This can also be outdated data.
    ALT: "100": zero and one are output alternately
    OFF: "101" (EL2262 only): the output stage is switched to high-resistance. No signal level is driven.

bCh1_ErrCtrl_CycCnt_ONE	BOOL	TRUE
bCh1_ErrCtrl_CycCnt_HOLD	BOOL	FALSE
bCh1_ErrCtrl_CycCnt_CONTI...	BOOL	FALSE
bCh1_ErrCtrl_CycCnt_ALT	BOOL	FALSE
bCh1_ErrCtrl_CycCnt_OFF	BOOL	FALSE
bCh1_ErrCtrl_CycCnt_ZERO	BOOL	FALSE

  Figure from TC3 sample program

- Prolonged absence (watchdog time and longer) → SM-Watchdog
  The device has a parameterizable **SM-Watchdog** (SyncManager watchdog) as output terminal. In each successful EtherCAT cycle this is 'wound up' again or reset. If it is not used for a certain time (default: 100 ms), the outputs are set to a definable state. For the time-based parameterization of the SM watchdog, see here [▶ 31].

  The output behavior after the watchdog has been triggered is parameterized via the ESC registers as follows:

  - For channel 1: Register 0x0F00
    Bit 0 = TRUE → Parameterization is activated
    Bit 4 to 6: Default value for watchdog behavior, see below.

- Channel 2: Register 0x0F01
  Bit 0 = TRUE → Parameterization is activated
  Bit 4 to 6: Default value for watchdog behavior, see below.
- The default behavior for watchdog monitoring is output=FALSE
- Note: the two byte registers 0x0F00 and 0x0F01 must be written as 1 word access at the same time The default values for the watchdog behavior are:
- Possible register values for the watchdog behavior are (names are taken from the demo program below):
  ZERO: " "000": A logical zero is output (default)
  ONE: "001": A logical one is output
  HOLD: "010": The value of the last bit in the previous cycle is output
  REP: "011": The data of the last cycle is output repeatedly
  ALT: "100": Zero and one are output alternately
  OFF: "101" (EL2262 only): The output stage is switched to high-resistance. No signal level is driven

Since CycleCounter and Watchdog behavior are controlled by the same register, the two registers are to be described completely in 1 operation, see sample program.

Using the ESC registers

If settings are loaded into ESC registers (in this case 0x0F00, for example), they are retained until they are overwritten or until the system is de-energized. If the system was de-energized, the required values have to be re-loaded into the registers.

The following example about the terminal function in case of a communication interruption illustrates the resulting 3 phases.

- Invented signals (as specified by the PLC) are output on channel 1 and 2 of a EL2262 and observed with an oscilloscope.
- For demonstration purposes, channel 1 and 2 are parameterized differently
- The example also applies to the EL1262-0010 (outputs).

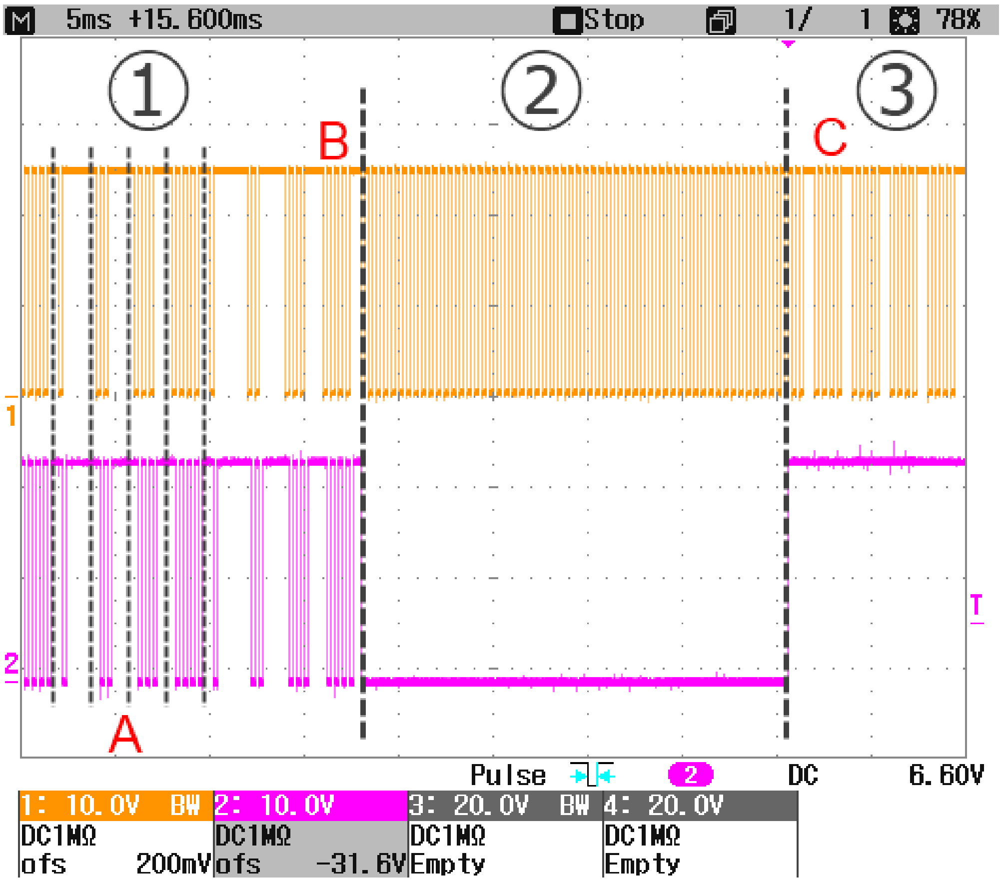


Fig. 197: Example: error in EL2262 from FW09

- **Phase 1: normal operation**: cyclic process data are sent to the terminal in time, the terminal outputs the data.
  In this example, channel 1 and 2 output the same signal curve, 10x oversampling, cycle time 2 ms, with a repeating pattern
  - general output = TRUE, 1 sample = FALSE
  - general output = TRUE, 2 samples = FALSE, in the next cycle then
  - general output = TRUE, 3 samples = FALSE, in the next cycle then
  - generally output = TRUE, 4 samples = FALSE, then starting again.
    In addition, the CycleCounterVariables are used. The data come from the sample program for watchdog parameterization.
- **Phase 2: CycleCounter monitoring:** the terminal does not receive any new process data, the CycleCounter is no longer operated from the terminal's point of view.
  This can occur for a short time (1 cycle) due to delayed data delivery or over a longer period due to communication interruption.
  The watchdog is also no longer used because the SyncManager events do not occur and starts to expire, but has no effect yet.
  The output behavior can be changed as above, in this example
  - channel 1 alternately outputs 0/1 during this phase
  - channel 2 alternately outputs 0 during this phase
- **Phase 3, watchdog case**: the watchdog has expired after the parameterized time [▸ 31], in this example 25 ms.
  The outputs now go into the parameterized or safe state.
  The output behavior can be changed as below, in this example
  - channel 1 continuously outputs the last sample during this phase
  - channel 2 outputs 1 during this phase

Note on the EL2262:

Firmware	CycleCounter monitoring behavior	Watchdog behavior
< FW09	Register $030D_{hex}$: incremented + 1 No output influence possible	Outputs: FALSE
>= FW09	Register $030D_{hex}$: incremented + 1 Output behavior as parameterized	Output behavior as parameterized

CycleCounter monitoring demonstration

In this further example the watchdog is set to 25 ms. By default, the last sample block is output repeatedly in phase 2 until phase 3:

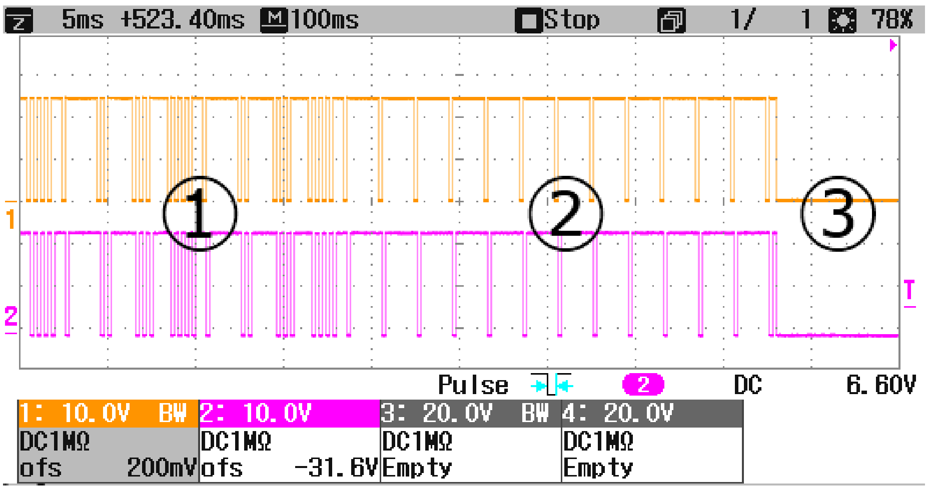


Fig. 198: Default output behavior if CycleCounter monitoring is deactivated

If CycleCounter monitoring is activated, the default is output = FALSE in phase 2.

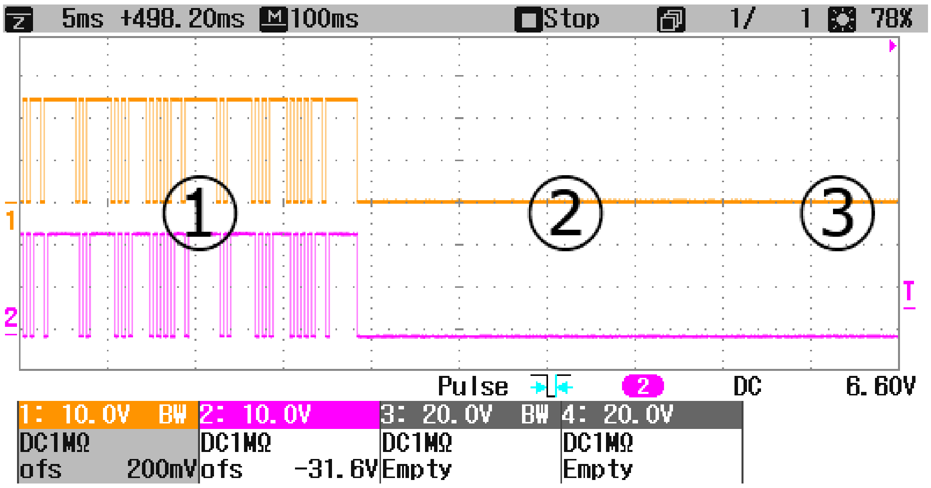


Fig. 199: Default output behavior if CycleCounter monitoring is activated

Now the CycleCounter special behavior "ALT" in phase 2 is parameterized for channel 1 and the Watchdog special behavior "REP" for channel 2: :

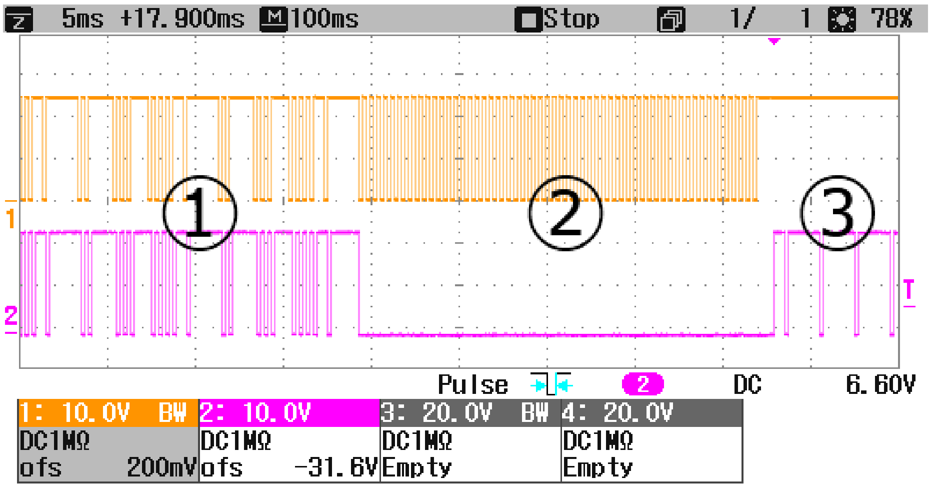


Fig. 200: Parameterized output behavior (system-specific)

6.10 Example programs

Using the sample programs

This document contains sample applications of our products for certain areas of application. The application notes provided here are based on typical features of our products and only serve as examples. The notes contained in this document explicitly do not refer to specific applications. The customer is therefore responsible for assessing and deciding whether the product is suitable for a particular application. We accept no responsibility for the completeness and correctness of the source code contained in this document. We reserve the right to modify the content of this document at any time and accept no responsibility for errors and missing information.

Procedure for starting the program

- After clicking the Download button, save the zip file locally on your hard disk, and unzip the *.TSM (configuration) and the *.PRO (PLC program) files into a temporary working folder.
- The *.pro file can be opened by double click or by the TwinCAT PLC Control application with menu selection “File/ Open”. The *.tsm file is provided for the TwinCAT System Manager (to review or overtake configurations).
- Connect the hardware and connect the Ethernet adapter of your PC to the EtherCAT coupler (further information on this can be found in the corresponding coupler manuals)
- Select the local Ethernet adapter (with real-time driver, if applicable) under System configuration, I/O configuration, I/O devices, Device (EtherCAT); then on the “Adapter” tab choose “Search...”, select the appropriate adapter and confirm (see Fig. *Searching the Ethernet adapter* + *Selection and confirmation of the Ethernet adapter*).

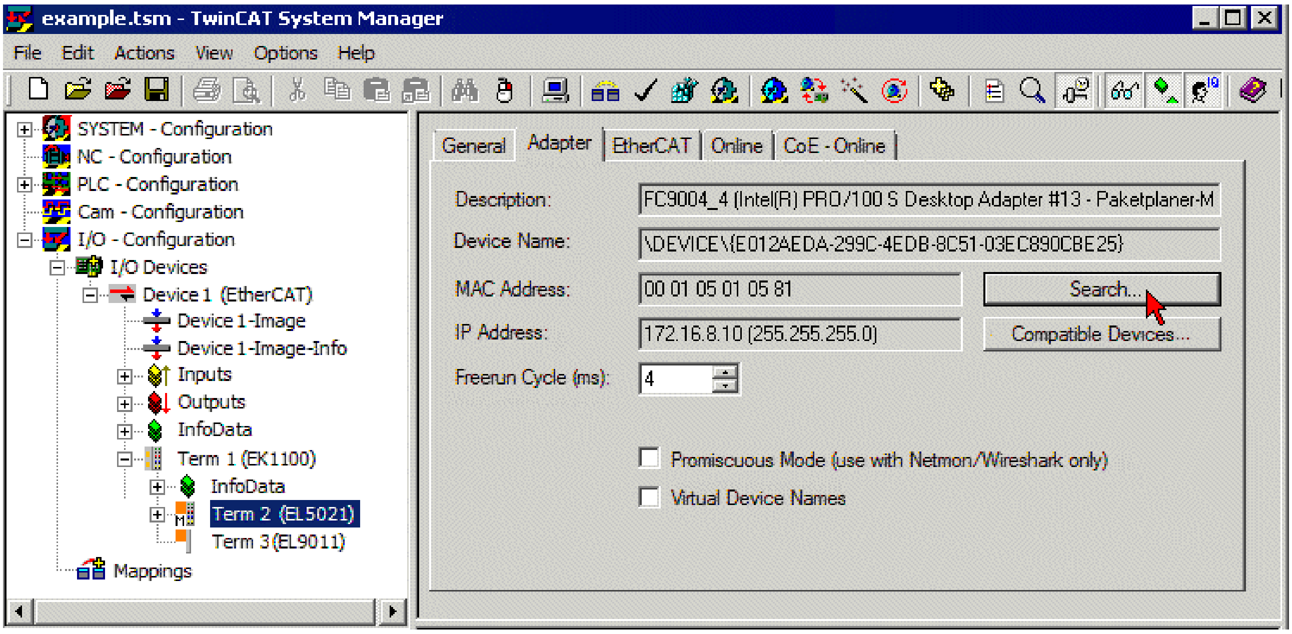


Fig. 201: Searching the Ethernet adapter

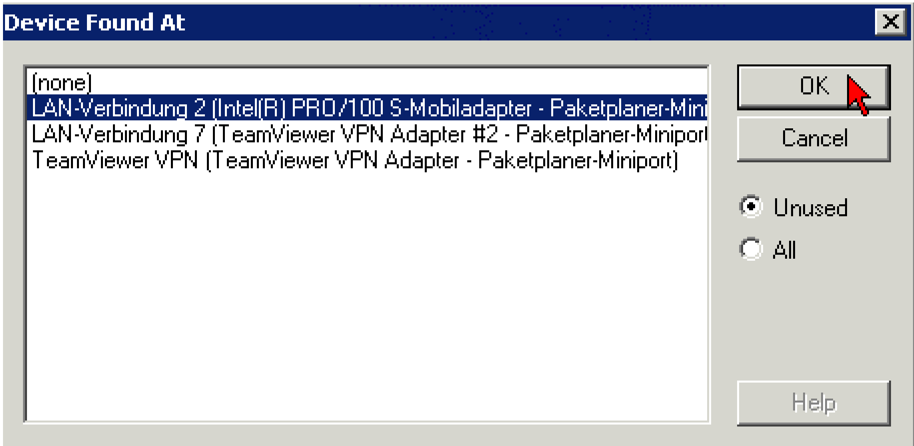


Fig. 202: Selection and confirmation of the Ethernet adapter

- Activate and confirm the configuration (Fig. *Activation of the configuration* + *Confirming the activation of the configuration*)

Fig. 203: Activation of the configuration

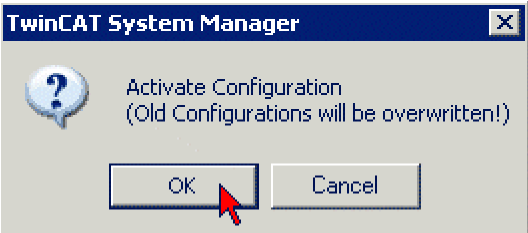


Fig. 204: Confirming the activation of the configuration

- Confirm new variable mapping, restart in RUN mode (Fig. *Generate variable mapping* + *Restarting TwinCAT in RUN mode*)

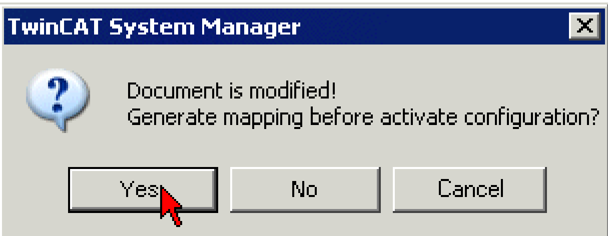


Fig. 205: Generating variable mapping

Fig. 206: Restarting TwinCAT in RUN mode

- In TwinCAT PLC, under the “Project” menu, select “Rebuild all” to compile the project (Fig. *Compile project*)

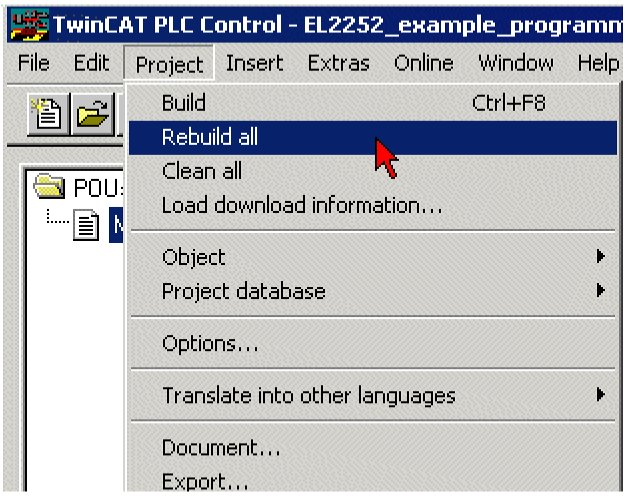


Fig. 207: Compile project

- In TwinCAT PLC: log in with the “F11” button, confirm loading the program (Fig. *Confirming program start*), run the program with the “F5” button

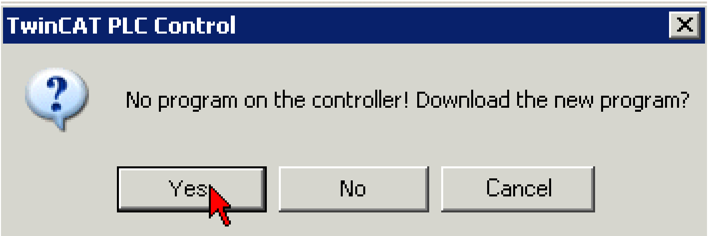


Fig. 208: Confirming program start

6.10.1 Example program 1: Frequency measurement with inductive sensor

Example 1: Frequency measurement with inductive sensor

Download: (https://infosys.beckhoff.com/content/1033/el126x/Resources/1675495563.zip)

Content:

- Version TwinCAT2: two files *.pro and *.tsm
- Version TwinCAT3: one file *.tnzip

Data:

- 1 ms cycle time
- 1000x oversampling on both channels

Connection diagram:

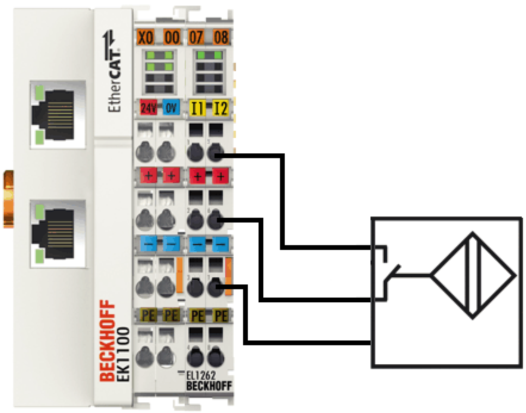


Fig. 209: *Wiring for example program 1*

6.10.2 Example program 2: Application of the SENT protocol with EL1262-0050

SAE J2716 SENT (Single Edge Nibble Transmission)

The SENT protocol is a point-to-point scheme for transmitting signal values from a sensor to a controller. It is intended to allow high resolution data transmission with a lower system cost than available serial data solutions. (1)

Hardware

The SENT protocol is a one-way, synchronous voltage interface which requires three wires: a signal line (low state < 0.5V, high state > 4.1V), a supply voltage line (5 V) and a ground line.

Protocol

Data is transmitted in units of 4 bits (1 Nibble). The interval between two falling edges (single edge) of the modulated signal with a constant amplitude voltage identifies the beginning of a frame and must be evaluated on the receiver side.

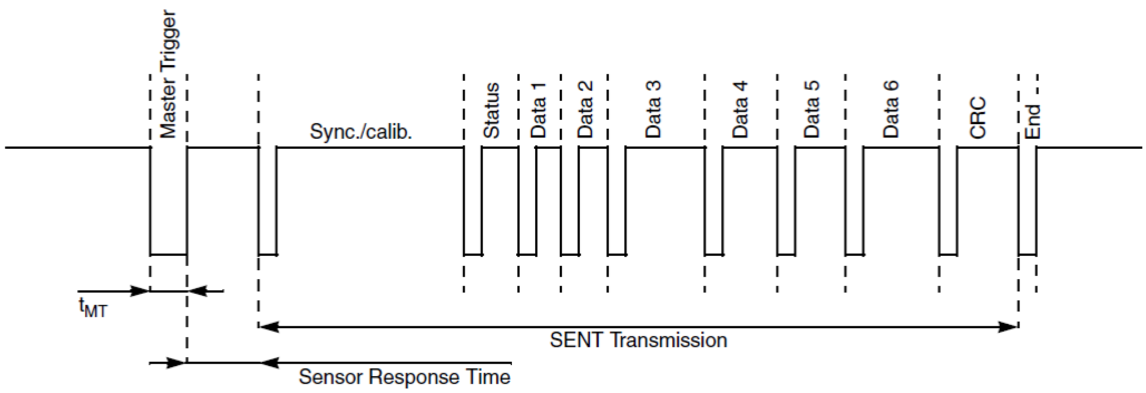


Fig. 210: SENT protocol

A SENT message is 32 bits long (8 Nibbles) and consists of the following components:

- 24 bits of signal data (6 Nibbles) that represents 2 measurement channels of 3 Nibbles each (such as pressure and temperature)
- 4 bits (1 Nibble) for CRC error detection
- 4 bits (1 Nibble) of status/communication information

(1) SAE J2716 standard, sae.org, accessed 2011-09-13

Implementation of the Control

```
⊞-fb_SENT
   wEL1262_State (%IB128) = 16#0008
   bEL1262_WC (%IX130.0) = FALSE
⊞-byInput (%IB0)
⊞-fbFlanke
   iNewData = 16#2F45
   byStatus = 16#0C
   iValue_1 = 16#00CD
   iValue_2 = 16#0034
   LogicCheckOK = 16#00002F45
   LogicChecknOK = 16#00000000

IF wEL1262_State.3 AND NOT bEL1262_WC THEN

    fb_SENT(byInput:=byInput);  (* Call Sent functionblock *)

    fbFlanke(CLK:=fb_SENT.bNewData ); (* if new Data are available *)
    IF fbFlanke.Q THEN
        iNewData:=iNewData+1;   (* Counter: Counts up ++1 if a sent protocol rece
        byStatus:=fb_SENT.arSentData[0];
        iValue_1:=BYTE_TO_WORD(fb_SENT.arSentData[3])+ROL(BYTE_TO_WORD(fb_SENT.ar
        iValue_2:=fb_SENT.arSentData[5]+fb_SENT.arSentData[4]*16#F;
        (* Only for Slave which mirror the Data nibbles *)
        IF (fb_SENT.arSentData[1]+fb_SENT.arSentData[6])=15 THEN
            LogicCheckOK:=LogicCheckOK+1;
        ELSE
            LogicChecknOK:=LogicChecknOK+1;
        END_IF
    END_IF

END_IF
```

Fig. 211: Extract of the ST implementation within TwinCAT

Example 2: Reading the SENT protocol

Download TwinCAT 2: (https://infosys.beckhoff.com/content/1033/el126x/Resources/4241239179.zip)

Download TwinCAT 3: (https://infosys.beckhoff.com/content/1033/el126x/Resources/9757494539.zip)

- Hardware requirements:
    - EL1262-0050
    - EL9505
    - A control that allows at least 1 ms task cycle time (IPC/ CX)

The EL1262-0050 is able to work with an oversampling of 1 µs and also 1000 Bit per 1 ms are available.

A Sent telegram has a resolution of 3 µs per digit. So, if 1 µ resolution will be used, it is enough to interpret the received data signal.

Connection diagram:

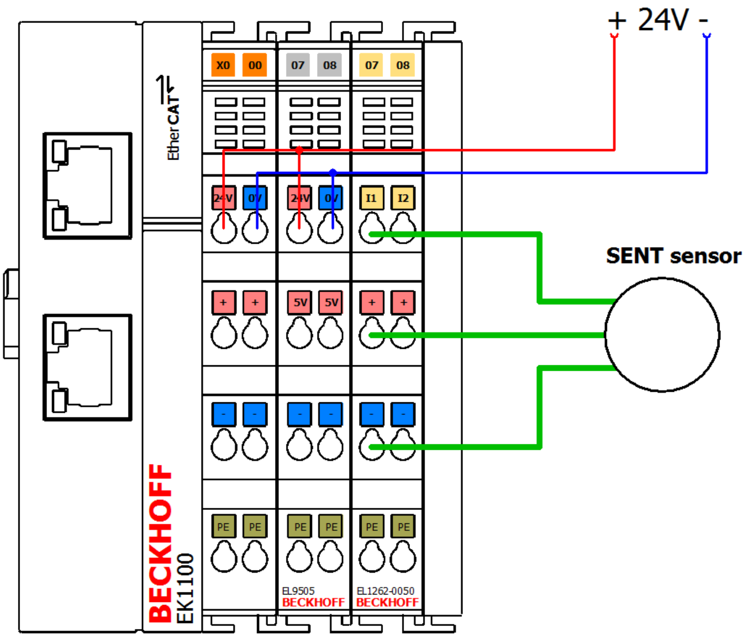


Fig. 212: Wiring for example program 2

Starting the example program

The application examples have been tested with a test configuration and are described accordingly. Certain deviations when setting up actual applications are possible.

The following hardware and software were used for the test configuration:

- TwinCAT master PC with Windows XP Professional SP 3, TwinCAT version 2.10 (Build 1330) and INTEL PRO/100 VE Ethernet adapter
- Beckhoff EtherCAT Coupler EK1100, EL1262 and EL9011 Terminals
- Inductive proximity limit switch, switching to positive potential, with 3-wire connection, max. switching frequency: 3000 Hz

6.10.3 Example program 3: Access to TEDS with EL1262-0050 and EL2262

A simple communication with TEDS modules as components of sensors and actuators can be applied with the terminals EL1262-0050 and EL2262.

See example: reading and writing TEDS data**.**

The URN can be read out with the example to use further functions of it.

Example 3: Reading and writing TEDS data

Preparations for starting the sample programs (tnzip file / TwinCAT 3)

- Click on the download button to save the Zip archive locally on your hard disk, then unzip the *.tnzip archive file in a temporary folder.

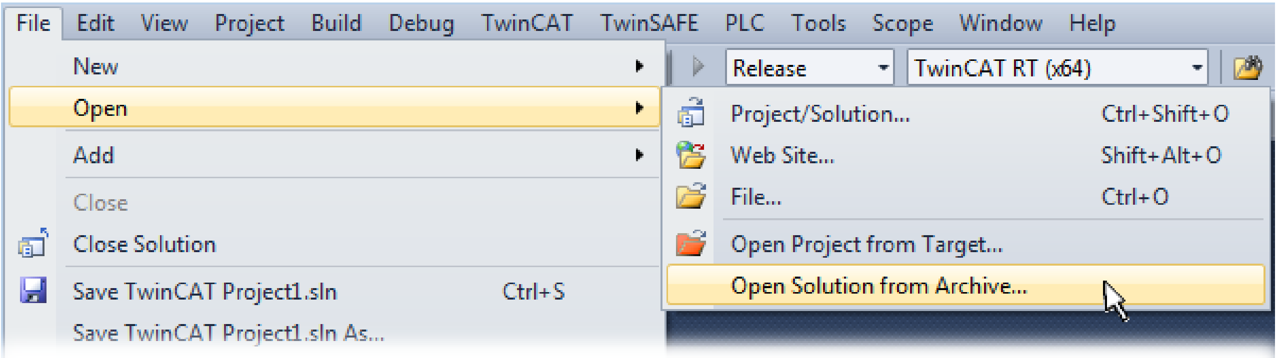


Fig. 213: Opening the *. tnzip archive

- Select the .tnzip file (sample program).
- A further selection window opens. Select the destination directory for storing the project.
- For a description of the general PLC commissioning procedure and starting the program please refer to the terminal documentation or the EtherCAT system documentation.
- The EtherCAT device of the example should usually be declared your present system. After selection of the EtherCAT device in the “Solutionexplorer” select the “Adapter” tab and click on “Search...”:

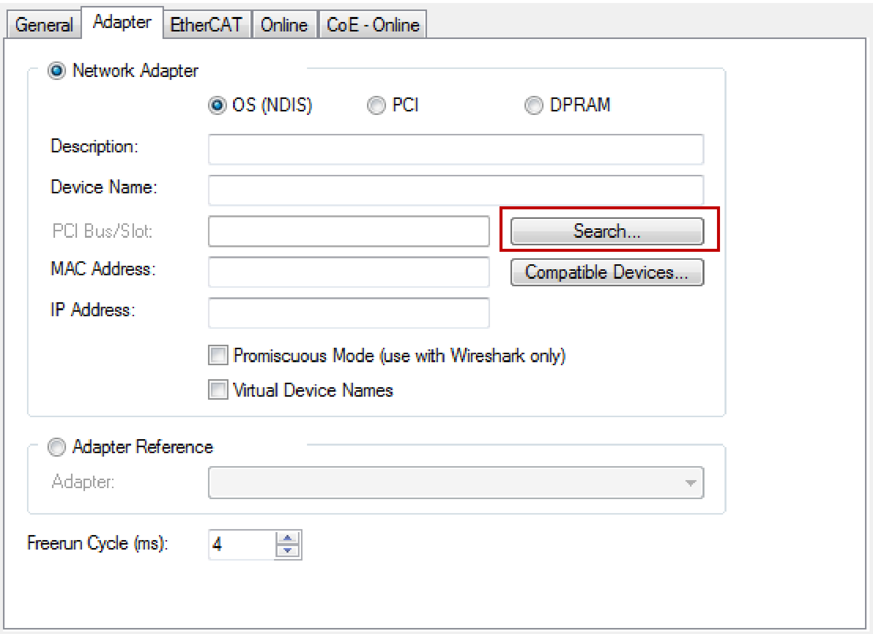
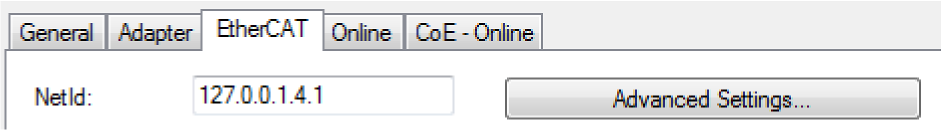


Fig. 214: Search of the existing HW configuration for the EtherCAT configuration of the example

- Checking NetId: the “EtherCAT” tab of the EtherCAT device shows the configured NetId:
  

  The first four numbers must be identical with the project NetId of the target system. The project NetId can be viewed within the TwinCAT environment above, where a pull down menu can be opened to choose a target system (by clicking right in the text field). The number blocks are placed in brackets there next to each computer name of a target system.
- Modify the NetId: By right clicking on “EtherCAT device” within the solution explorer a context menu opens where “Change NetId...” have to be selected. The first four numbers of the NetId of the target computer must be entered; both last values are 4.1 usually.
  Example:
  - NetId of project: myComputer (123.45.67.89.1.1)
  - Entry via „Change NetId...“: 123.45.67.89.4.1

Program description / function

This sample program illustrates how to read/write the data of a separate TEDS module (TEDS = Transducer Electronic Data Sheet). Such TEDS modules are available on the market for retrofitting sensors or actuators, in order to identify the device after installation or to read out specific data (calibration, manufacturer etc.). The device used in this example was an HBM TEDS 1−TEDS−BOARD−L, version 2018.

This sample program is expressly intended as a feasibility demonstration. Specifically, there is no claim to interoperability with any other TEDS modules. It is the responsibility of users to transfer the methods formulated here to their own implementations.

This demonstration does not cover TEDS modules that are integrated in the sensor and communicate on the sensor lines. This is common for IEPE (vibration) or strain gauges/measuring bridges. It is possible to connect an IEPE sensor equipped with TEDS to Beckhoff ELM3602/ELM3604 terminals.

The following configuration is required:

[EK1100] + [EL2262] + [EL9505] + [EL1262-0050] + EL9011

The configuration can control 2 TEDS modules. Only single-channel operation is shown in the example.

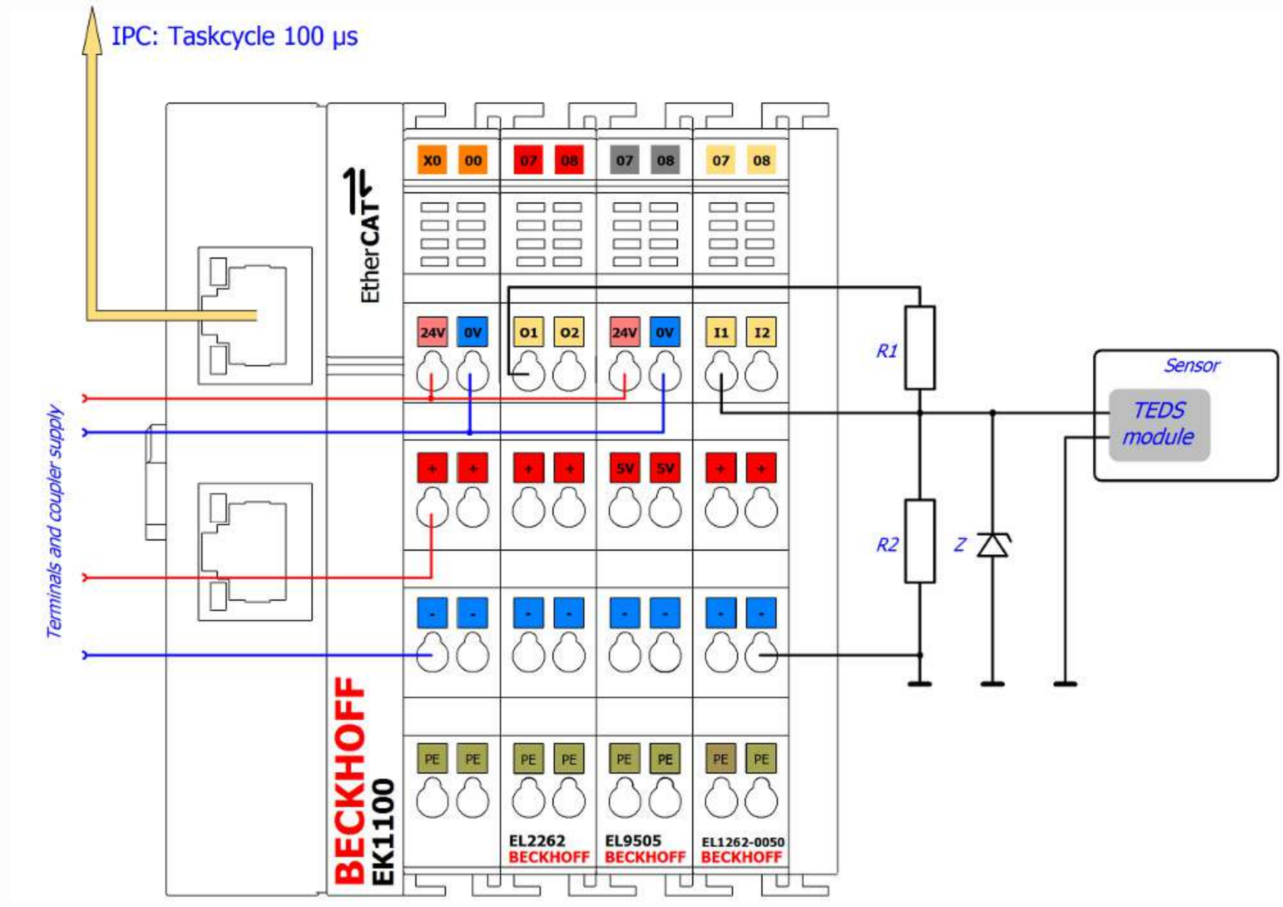


Fig. 215: Wiring for sample program 3

The voltage divider can be dimensioned with R1 = 2180 Ω (e.g. 680 Ω + 1500 Ω), R2 = 680 Ω and Z = 5.1 V for example.

Notes on the program (visualization)

First the URN has to be read (A). Only then are further functions available.

The program determines the URN for each bit by reinitializing the module, since the terminal for the input causes a time offset that is too large (see "Bit repeat count" at the top right).

Data can be written either by entering hexadecimal values (B) or a text string (ASCII) (C); hexadecimal values must be separated by spaces in the text field. Which of the two inputs is to be used for writing can be specified with the checkbox “*Write ASCII data*” (E):

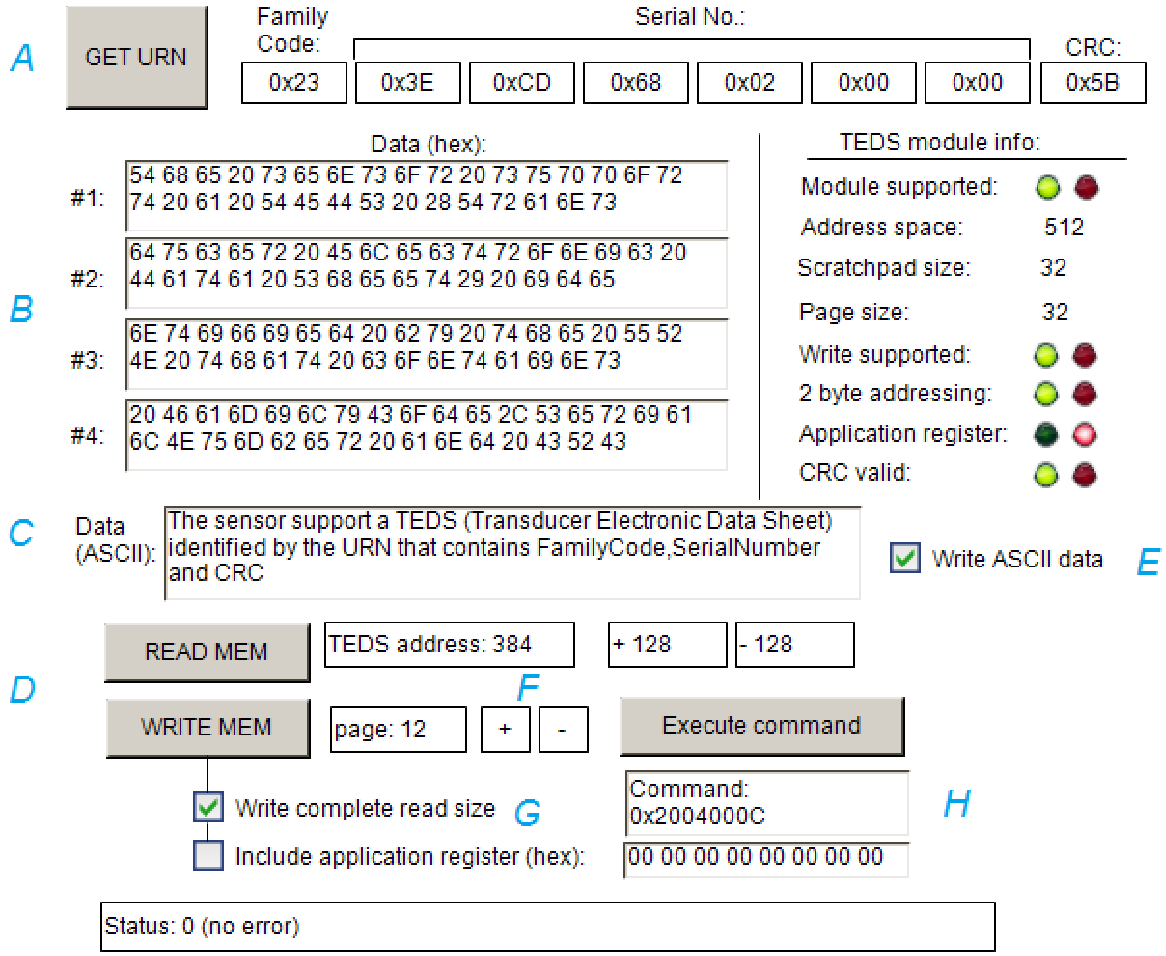


Fig. 216: Visualization of the sample program for TEDS with EL1262-0050 and EL2262

The basic function after the identified URN is (D) reading (READ MEM) and writing (WRITE MEM) TEDS data. By issuing such a command, the associated command statement is generated in the text field (H) and can also be changed and then executed with "Execute command". Via +/- the TEDS address or page can be changed (F). Both the start address and "page" can be entered directly for read / write accesses.

The hexadecimal data (B) of *text field* #1 to #4 each represent 32 bytes of the total read/write buffer size of 128 bytes, as configured in the sample program. If the checkbox “*Complete read size*” (G) is unchecked, only *text field* #1 will be used for writing usually (except the module supports page sizes > 32 byte). Accordingly, only the first characters of the ASCII data text will be written. In any case, the number of bytes as a page of the TEDS module is configured will be used. Note, that the module usually supports write access to addresses of a multiple value of the page size only. For example, assuming a page size of 32 bytes and the address 234 is input, an error 0x35 ‘writing fail’ will occur by a WRITE MEM command; but if address 352 is used, this is valid and there is no error).

Selection of “Include application register” provides whether the application register shall be written or read additionally (G).

Download:
https://infosys.beckhoff.com/content/1033/el126x/Resources/5750275595.zip

6.10.4 Example program 4: Parameterization of the CycleCounter and watchdog behavior

Download (https://infosys.beckhoff.com/content/1033/el126x/Resources/1909494283.zip)

This example uses a TwinCAT 3 PLC program to demonstrate the behavior of an EL2262 with regard to the CycleCounter and Watchdog cases.

The example is also applicable to the EL1262-0010.

Data:

- 2 ms cycle time
- 10x oversampling on both channels
- defined output pattern, which is repeated every 4 cycles

In the Settings dialog of the terminal the SyncManager watchdog time is set to 25 ms.

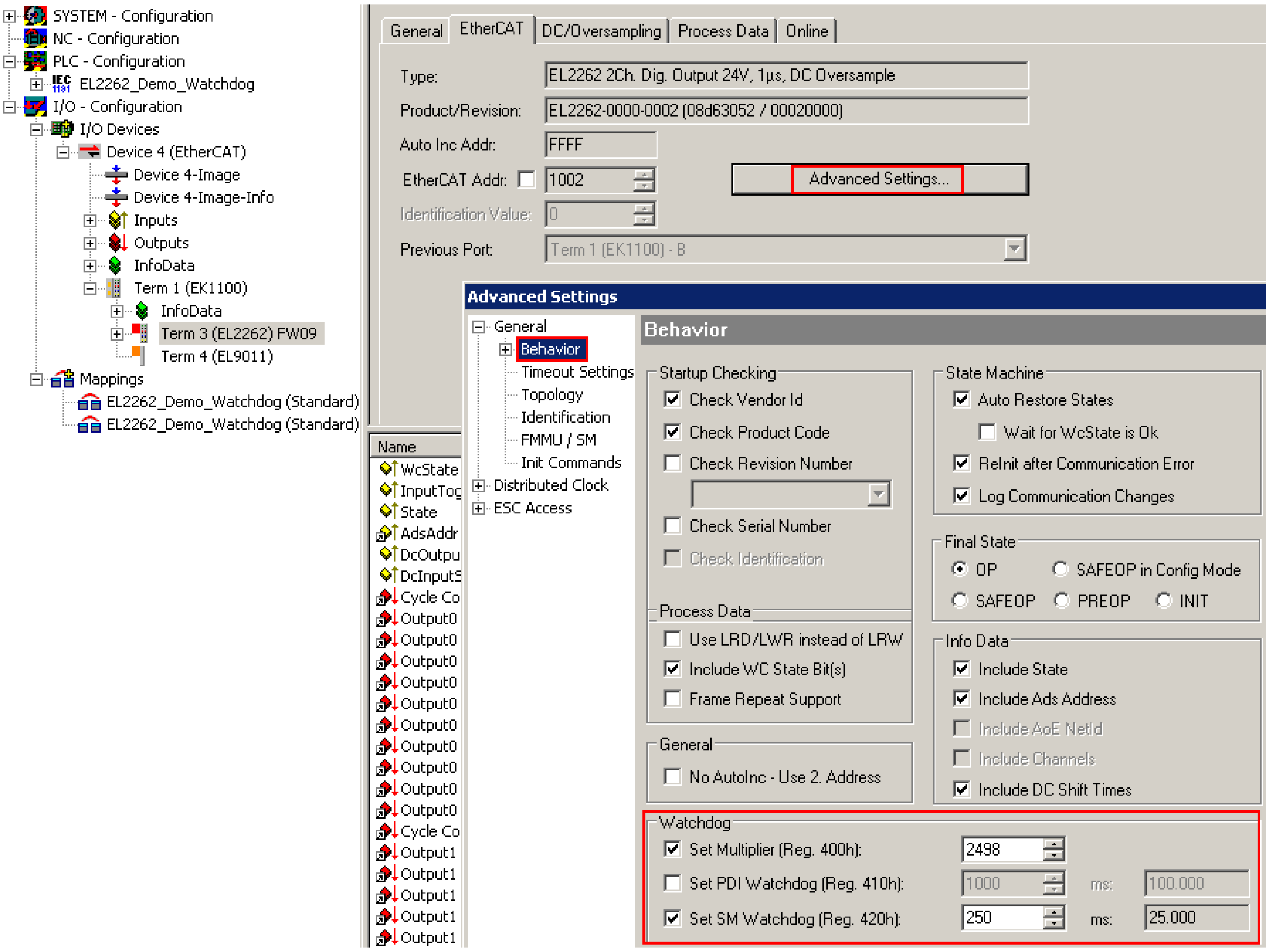
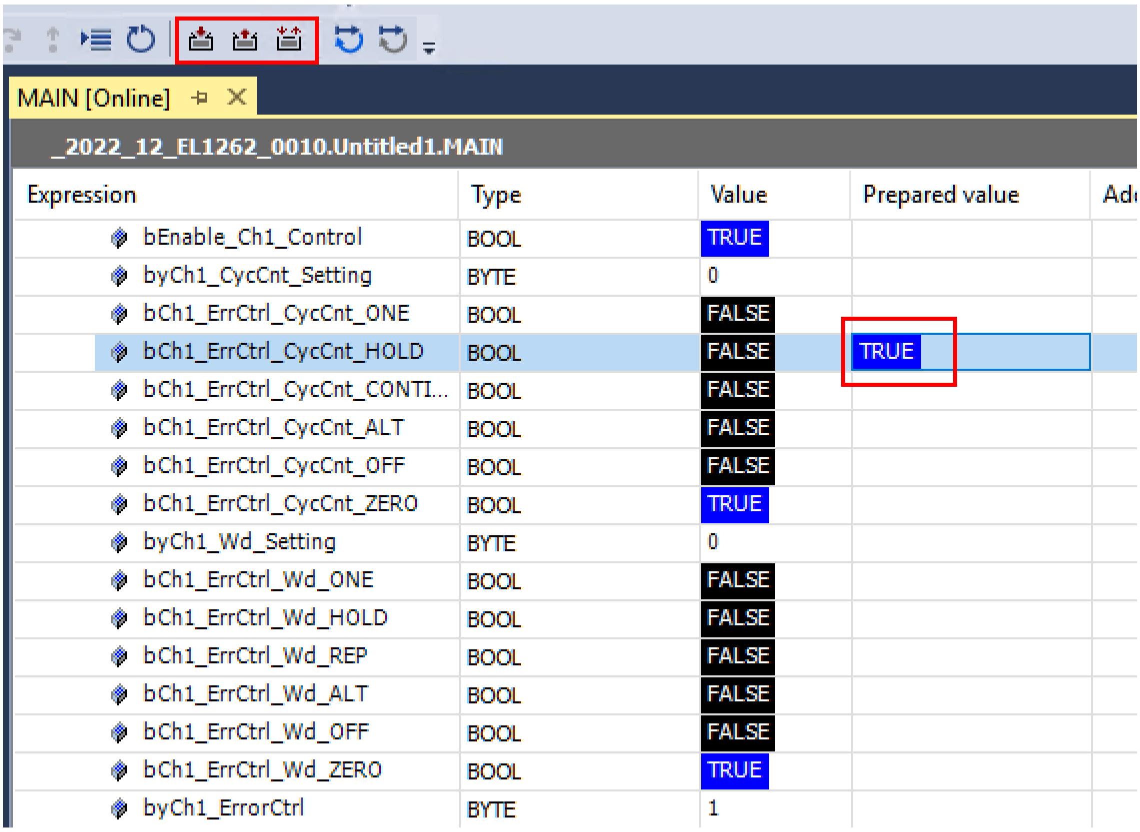


Fig. 217: System Manager Watchdog Time

After starting the PLC, the CycleCounter and the Watchdog behavior for each channel can be set online:



It is loaded every second via ADS/acyclic data traffic into the registers 0x0F00/0x0F01 and is then effective.

Any PDI errors counted in register 0x030D are read.

To readjust the communication interruption the EtherCAT cable can be pulled to the EL2262 or via *DisableEcFrameSending* the cyclic data sending can be stopped by software.

Using the ESC registers

If settings are loaded into ESC registers (in this case 0x0F00, for example), they are retained until they are overwritten or until the system is de-energized. If the system was de-energized, the required values have to be re-loaded into the registers.

7 Appendix

7.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

7.2 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.

NOTICE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 196].
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!

EL1262			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 09	01	EL1262-0000-0002	2008/02
	02		2011/01
	03		2011/02
10 *)	04 *)	EL1262-0000-0003	02024/01

EL1262-0010			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
02 - 03	01	EL1262-0010-0000	2023/10
04*)	02 *)	EL1262-0010-0001	2024/03

EL1262-0050			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 - 04 *)	03	EL1262-0050-0002	2012/06
	04 *)	EL1262-0050-0003	2024/01

EL1264			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
00 *)	05 *)	EL1262-0000-0016	03/2025

*) 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.

7.3 Firmware Update EL/ES/EM/ELM/EP/EPP/ERPxxxx

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

NOTICE

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.

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:

- Each EtherCAT slave has a device description, consisting of identity (name, product code), timing specifications, communication settings, etc.
  This device description (ESI; EtherCAT Slave Information) can be downloaded from the Beckhoff website in the download area as a zip file and used in EtherCAT masters for offline configuration, e.g. in TwinCAT.
  Above all, each EtherCAT slave carries its device description (ESI) electronically readable in a local memory chip, the so-called **ESI EEPROM**. When the slave is switched on, this description is loaded locally in the slave and informs it of its communication configuration; on the other hand, the EtherCAT master can identify the slave in this way and, among other things, set up the EtherCAT communication accordingly.

NOTICE

Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.
- Meaning for the ESI file: Modification on the application side (i.e. by the user) is not permitted.
- Meaning for the ESI EEPROM: Even if a writeability is technically given, the ESI parts in the EEPROM and possibly still existing free memory areas must not be changed beyond the normal update process. Especially for cyclic memory processes (operating hours counter etc.), dedicated memory products such as EL6080 or IPC's own NOVRAM must be used.

- 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.

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

NOTICE

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.

7.3.1 Device description ESI file/XML

NOTICE

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:

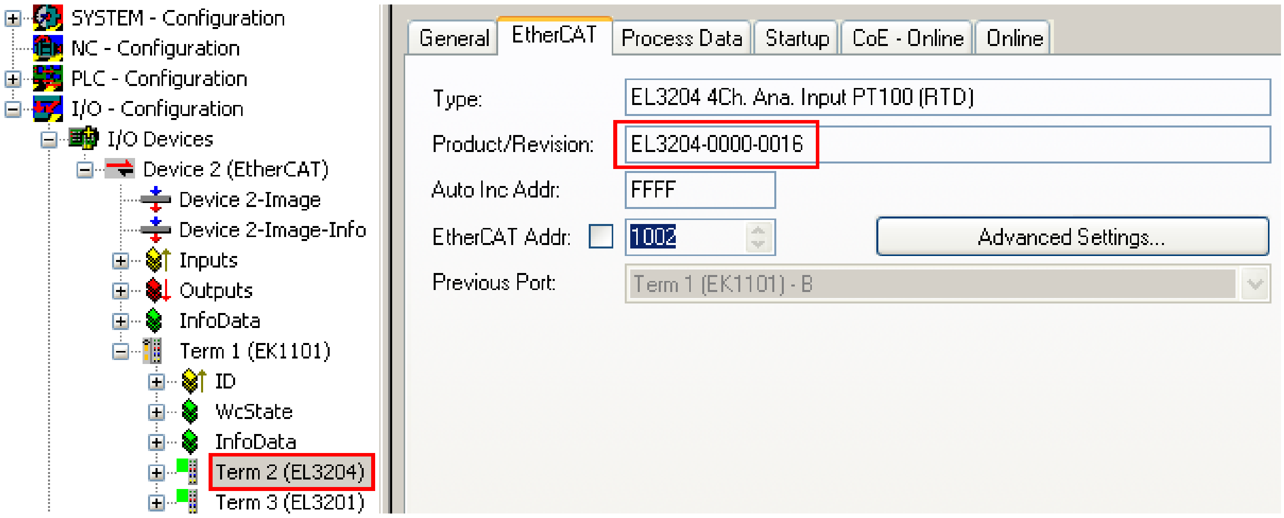


Fig. 218: Device identifier consisting of name EL3204-0000 and revision -0016

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:

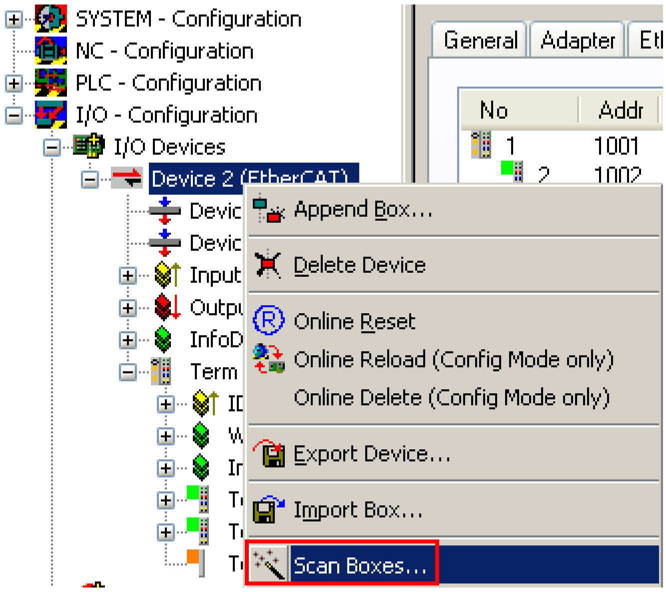


Fig. 219: Scan the subordinate field by right-clicking on the EtherCAT device

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

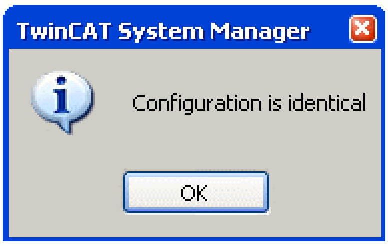


Fig. 220: Configuration is identical

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

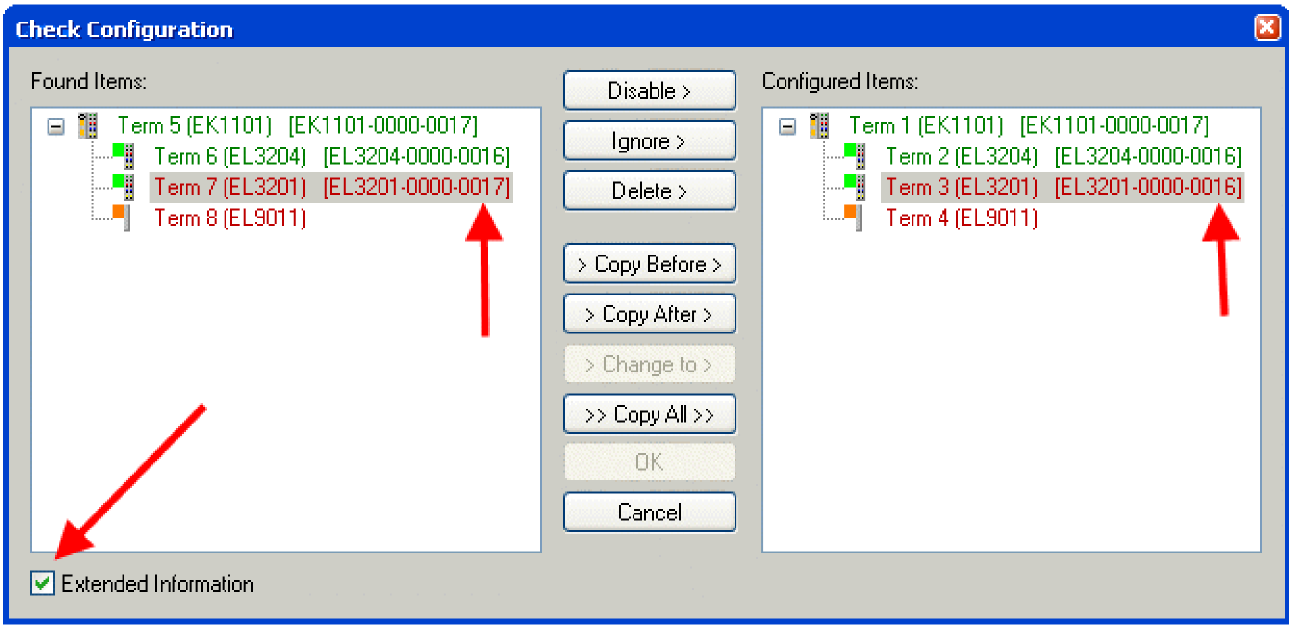


Fig. 221: Change dialog

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*

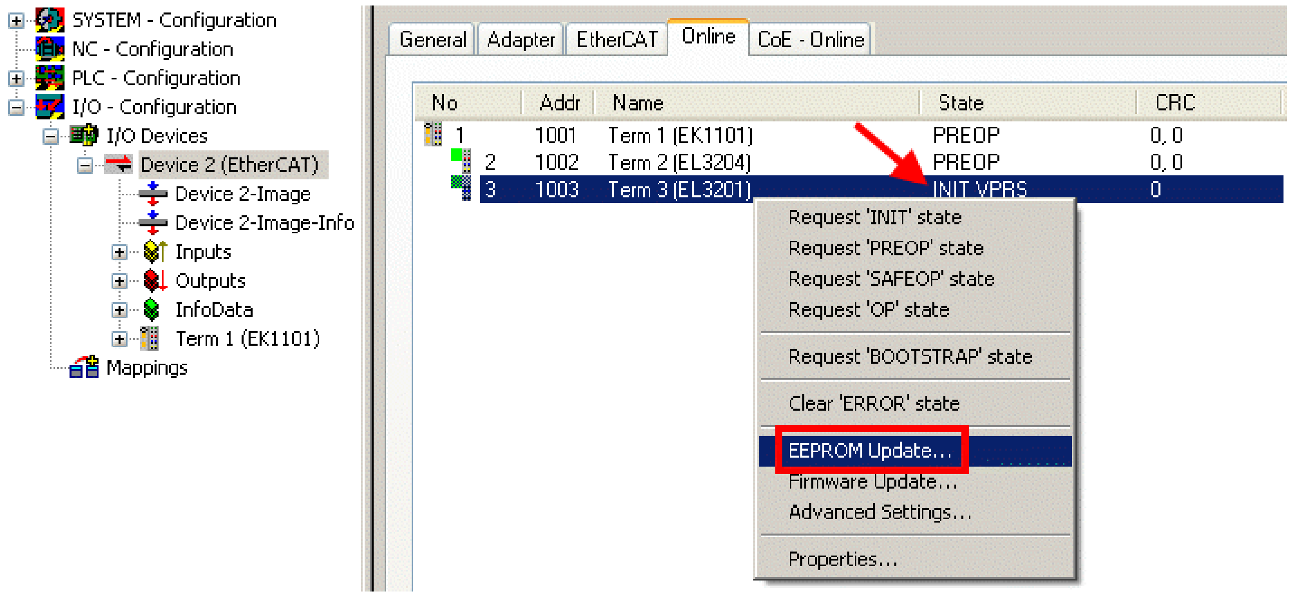


Fig. 222: EEPROM Update

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.

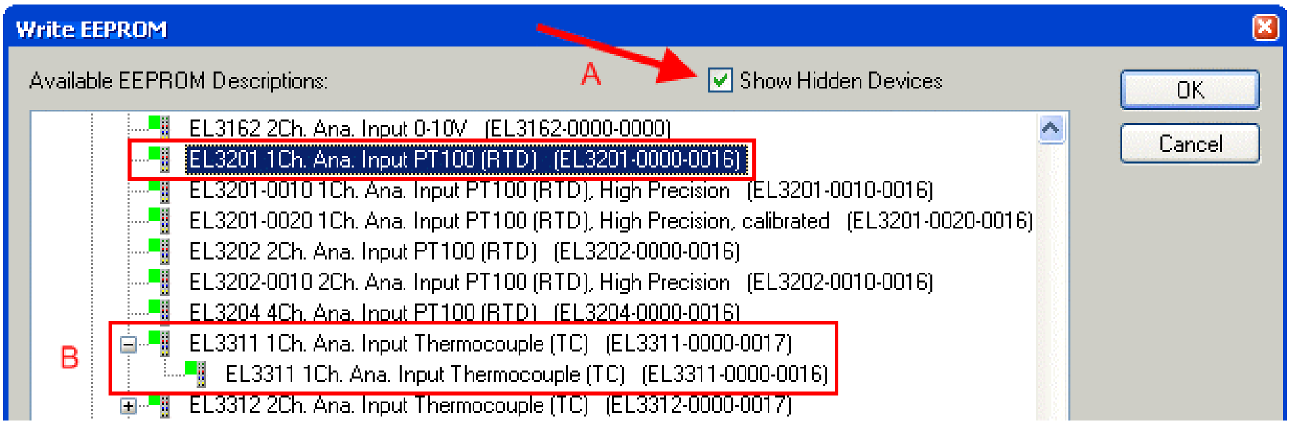


Fig. 223: Selecting the new ESI

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.

7.3.2 Firmware explanation

Determining the firmware version

Determining the version via the TwinCAT 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.

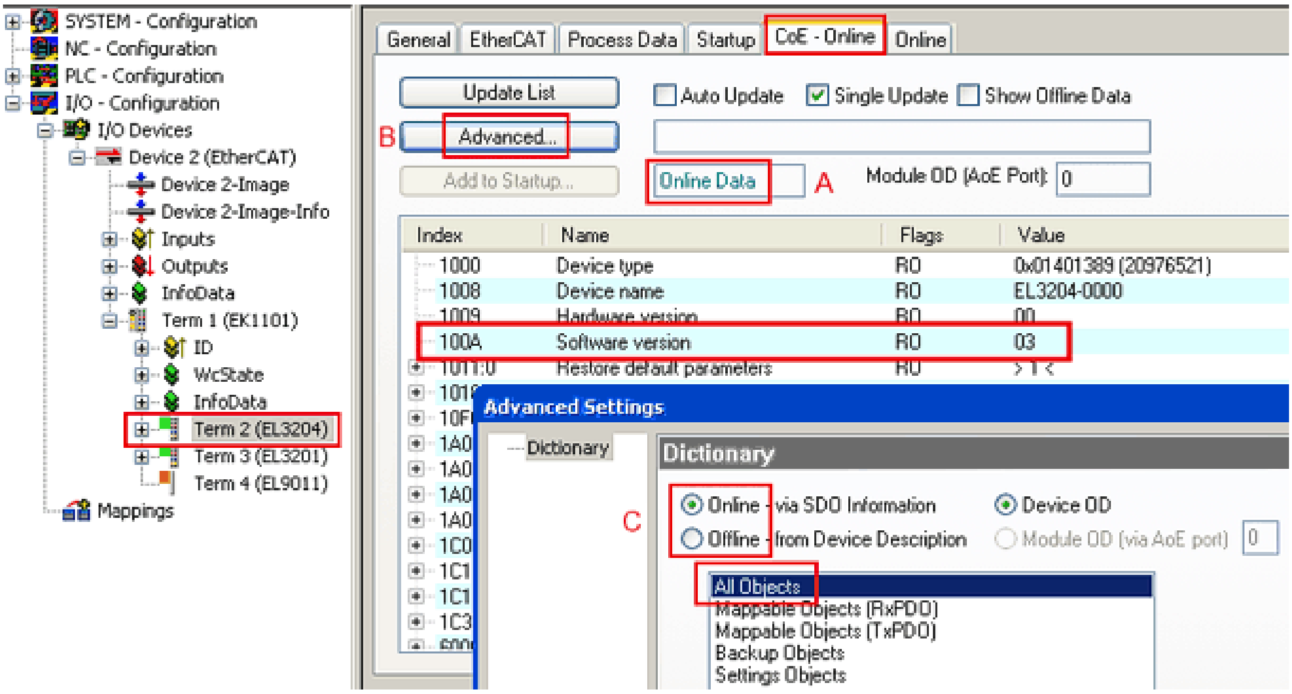


Fig. 224: Display of EL3204 firmware version

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*.

7.3.3 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.*

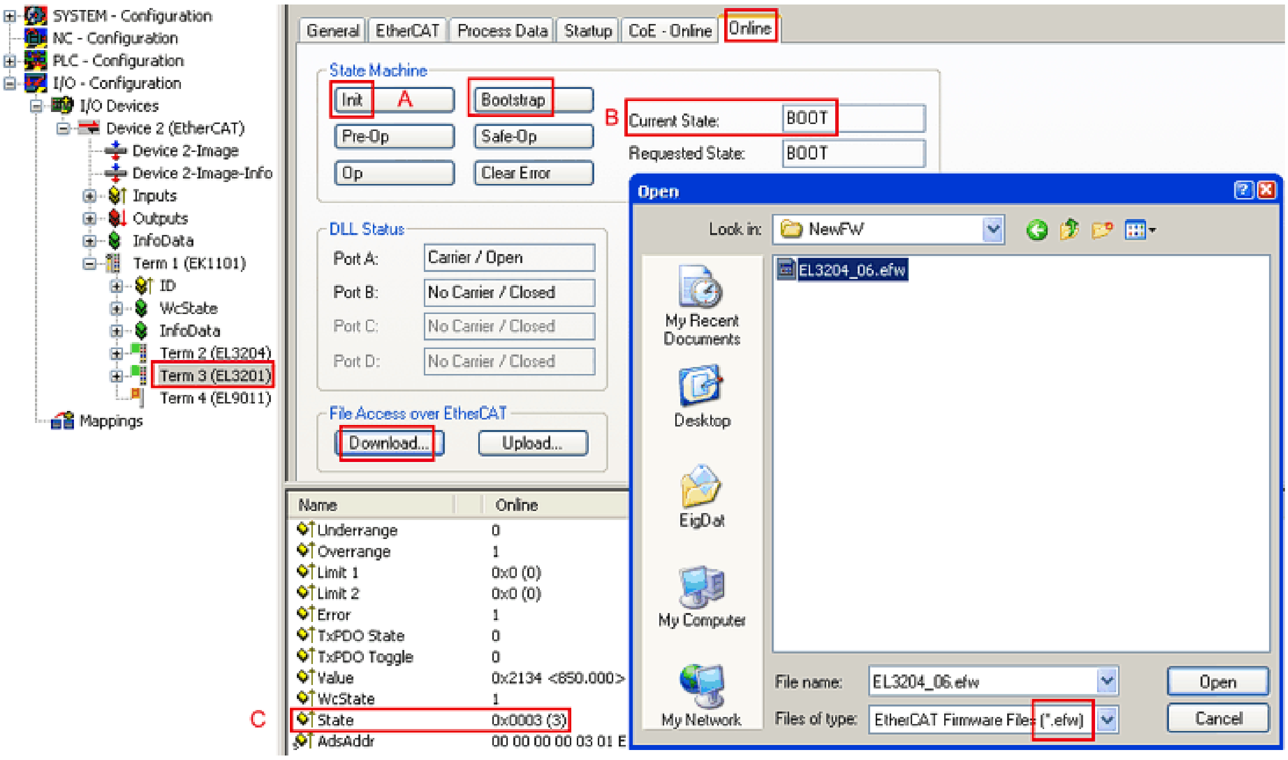
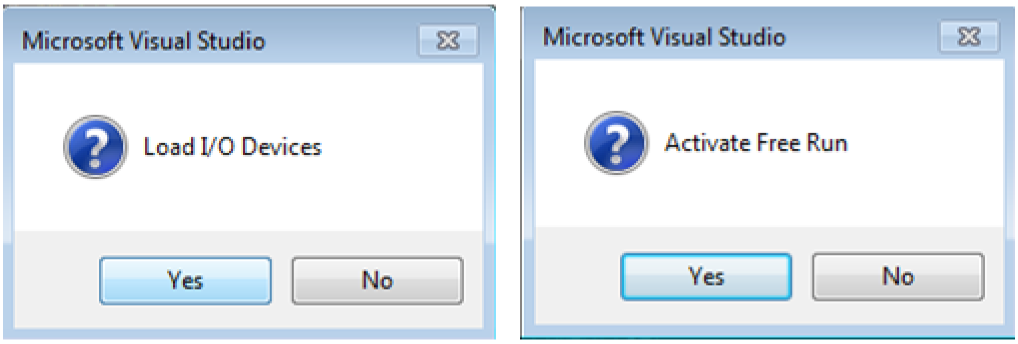


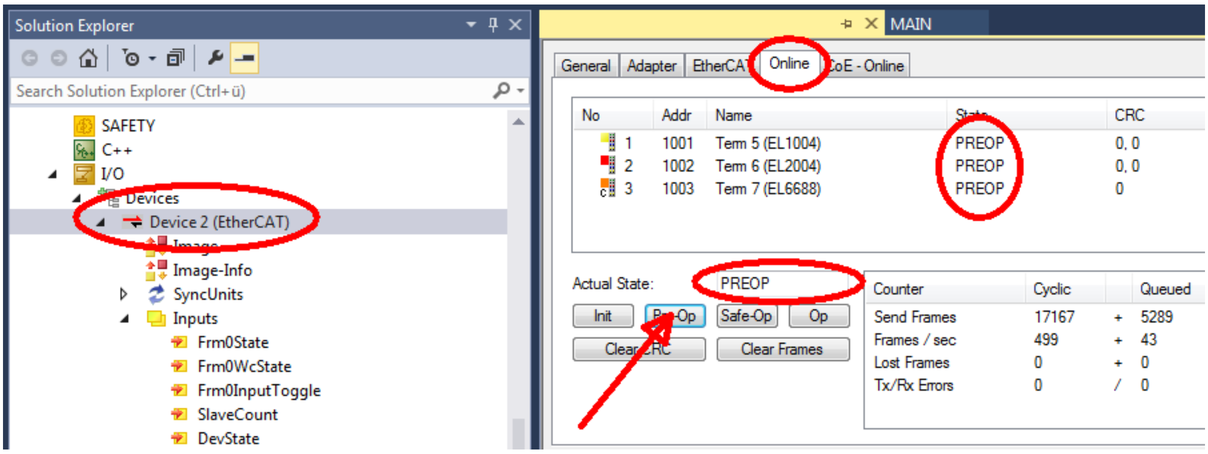
Fig. 225: Firmware Update

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.

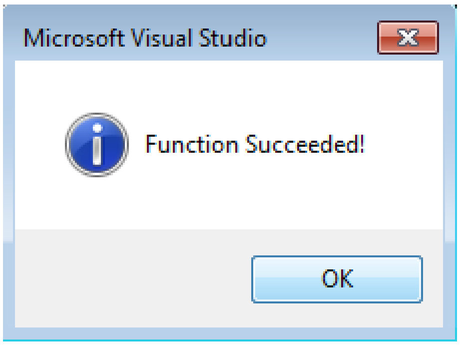


- Switch EtherCAT Master to PreOP



- 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.

7.3.4 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 TwinCAT 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.

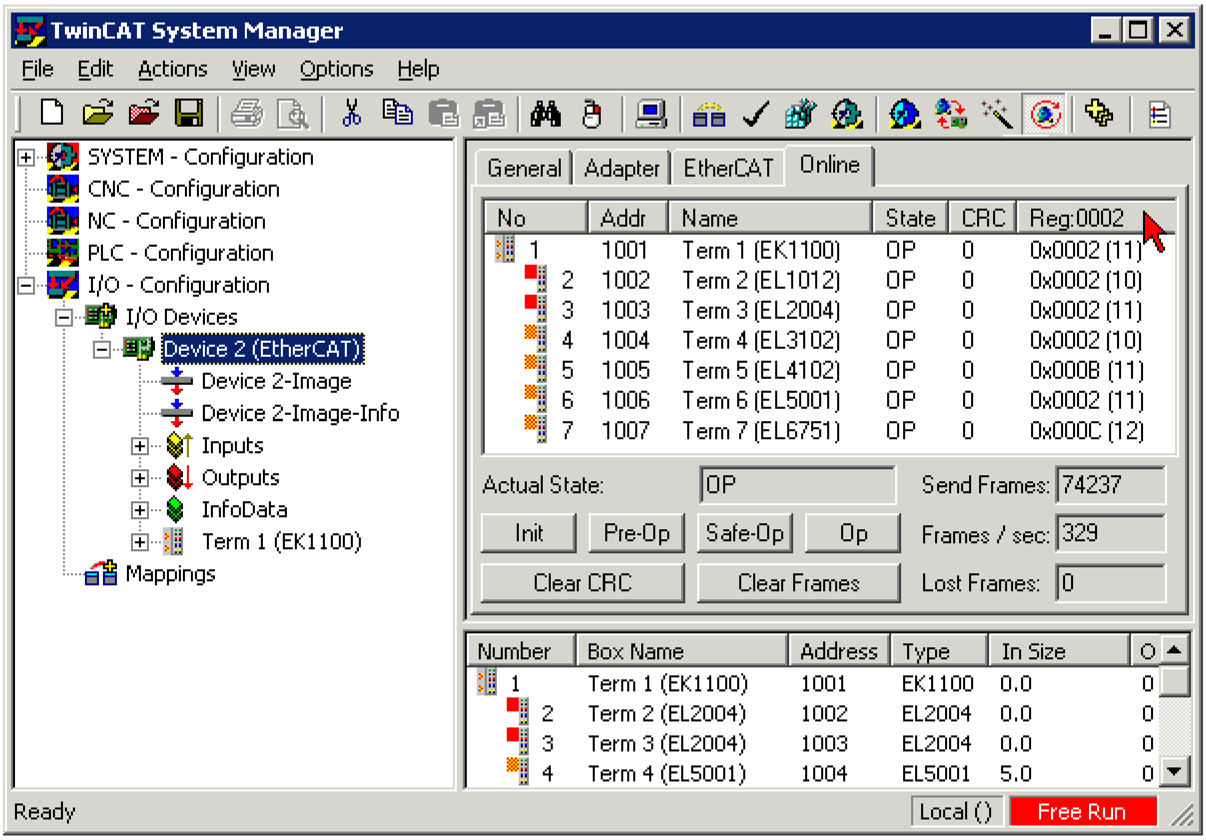


Fig. 226: FPGA firmware version definition

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

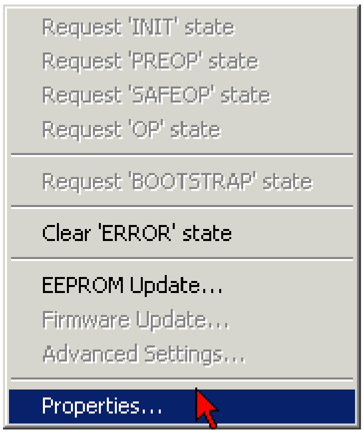


Fig. 227: Context menu *Properties*

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.

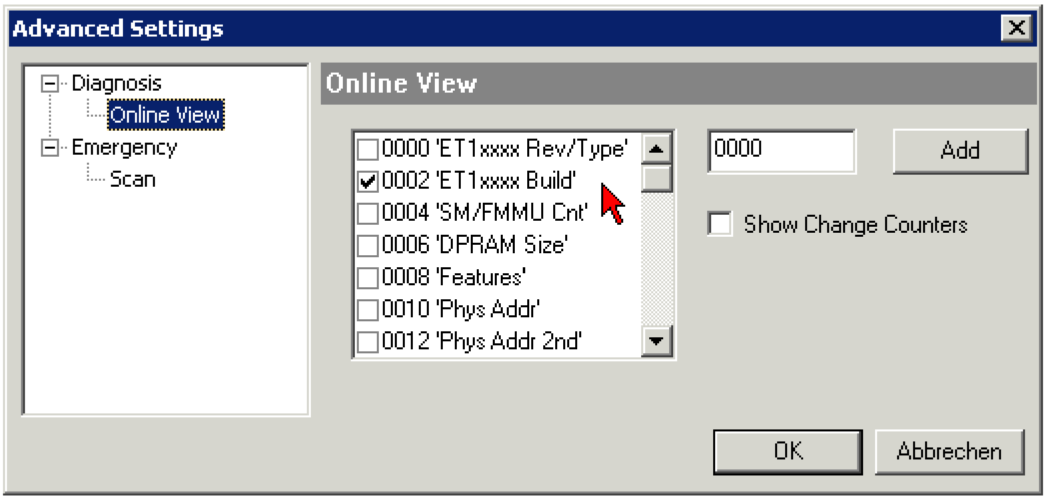


Fig. 228: Dialog *Advanced Settings*

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.

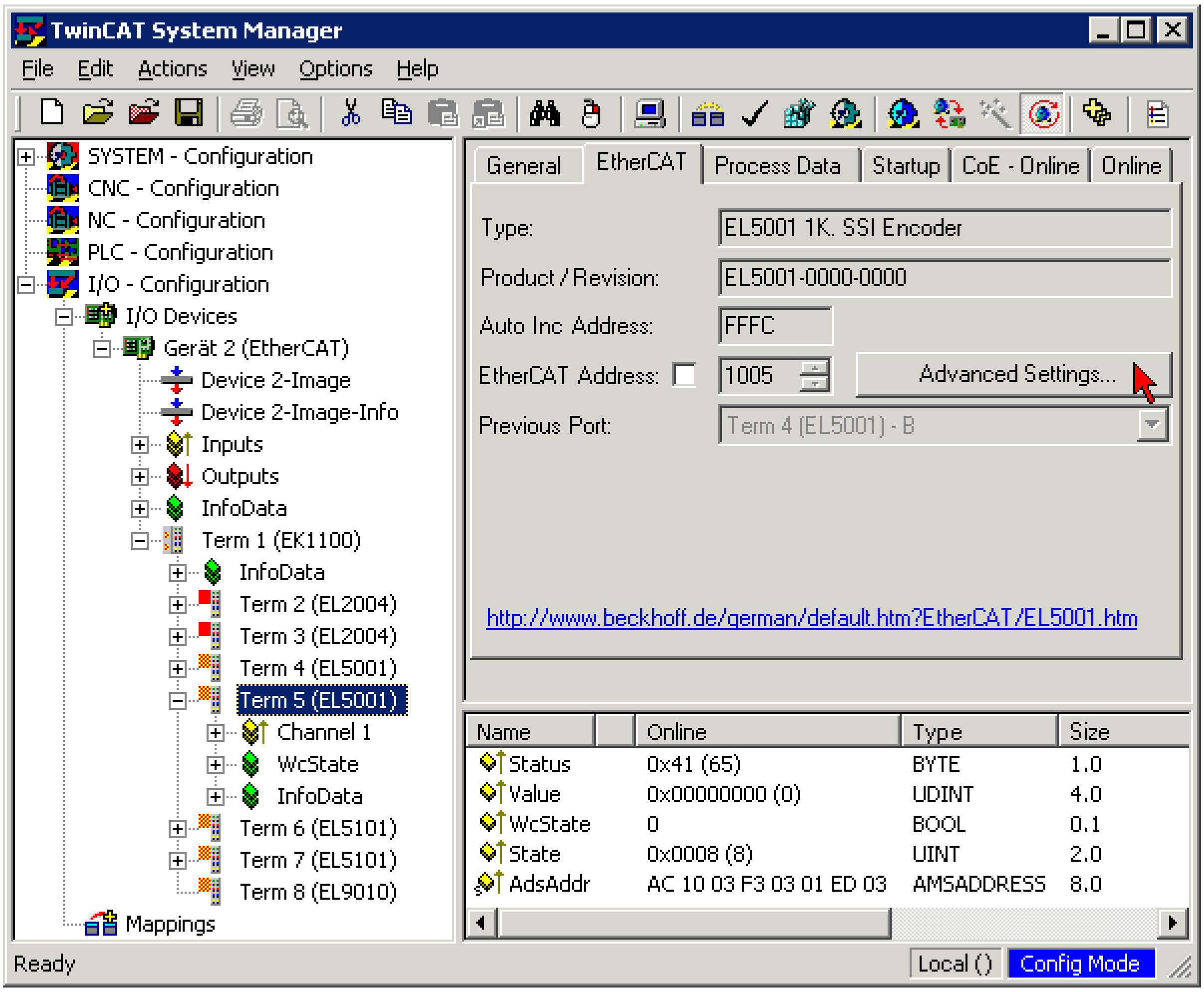
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

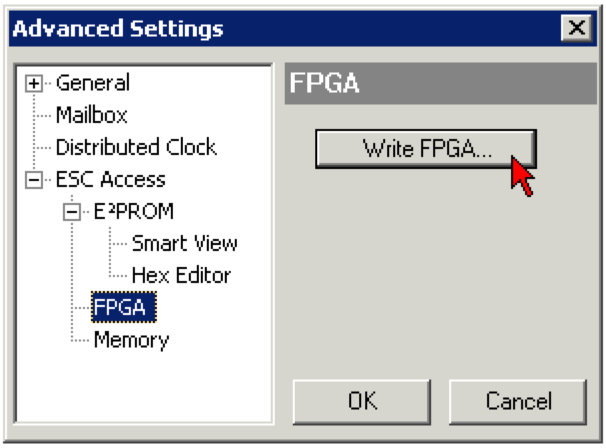
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- 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.

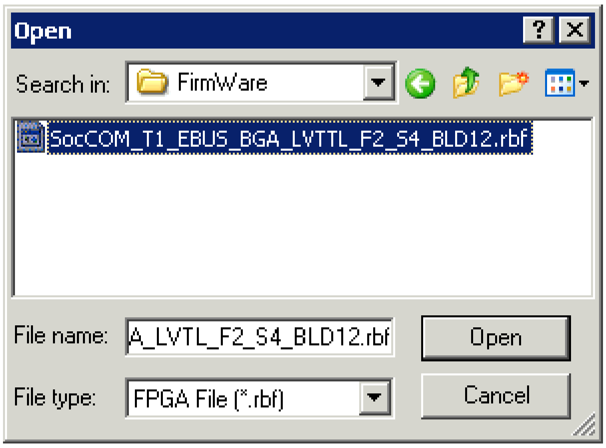
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
click the *Advanced Settings* button in the *EtherCAT* tab:



- 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

NOTICE

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!

7.3.5 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.

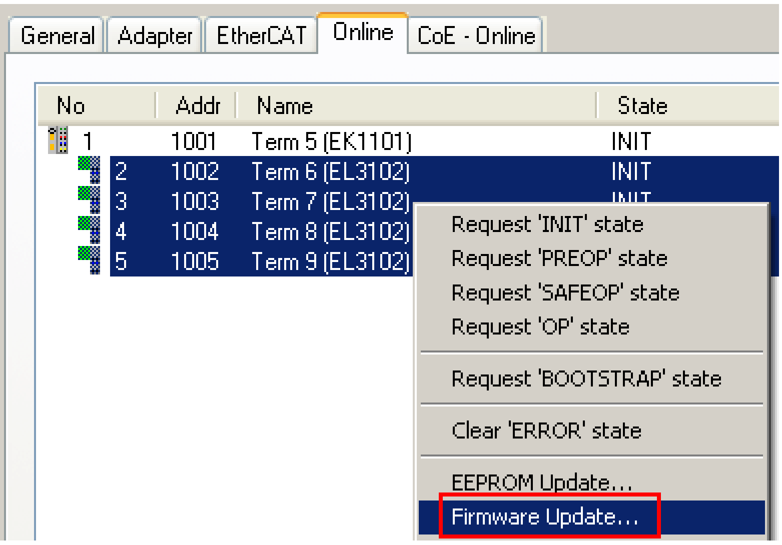


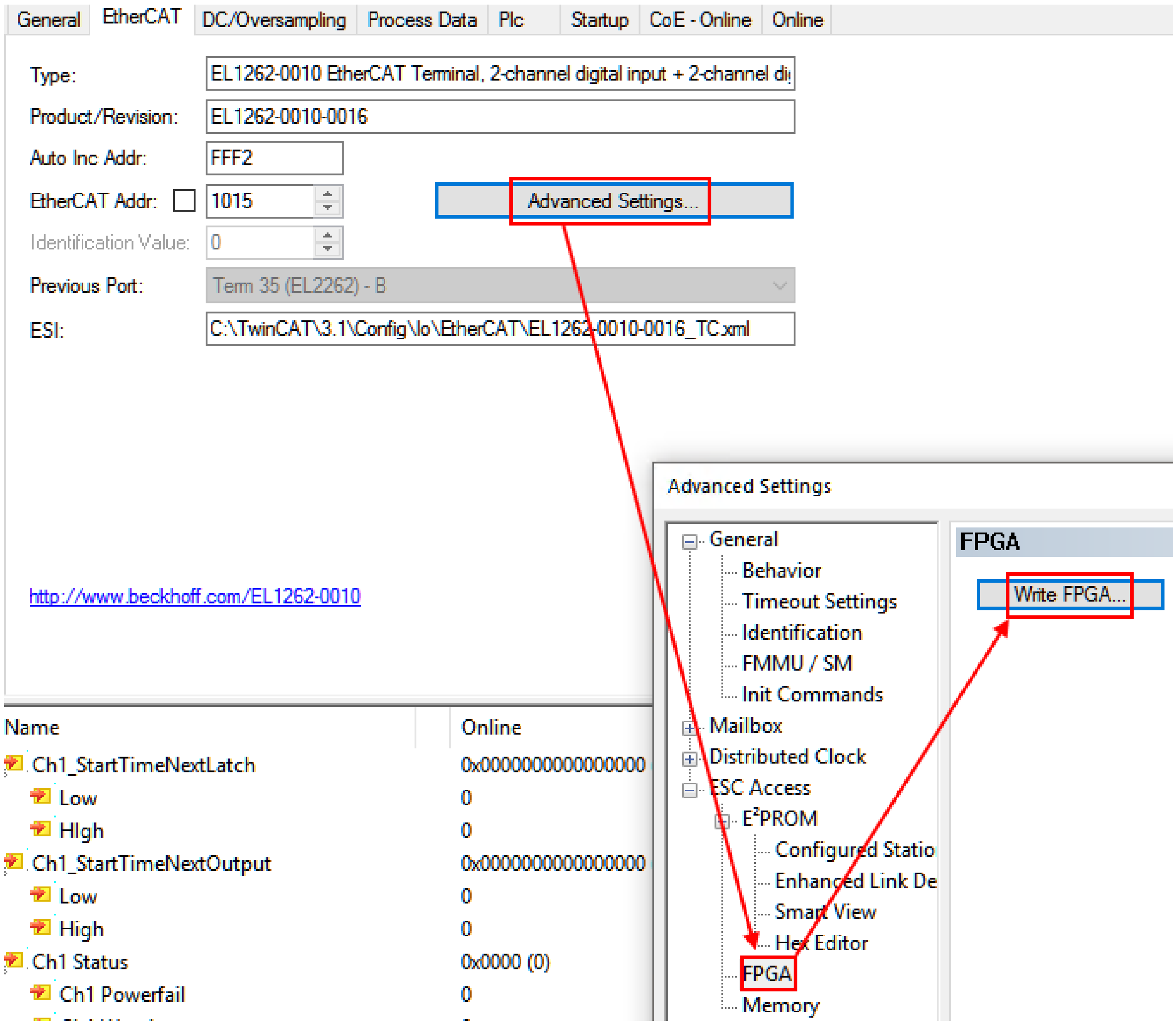
Fig. 229: Multiple selection and firmware update

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

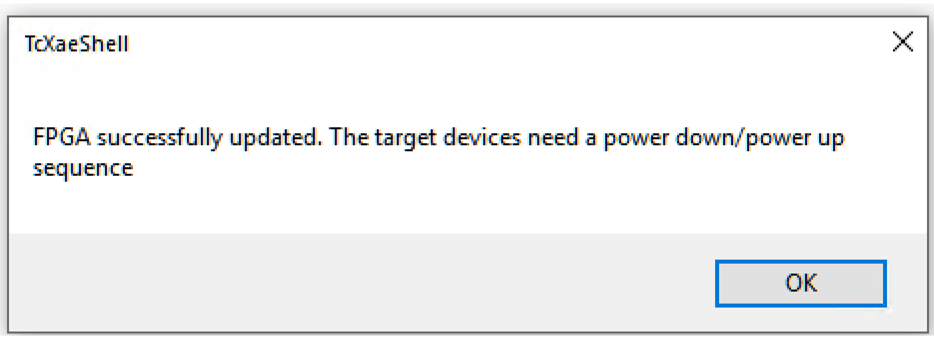
7.4 Firmware update EL1262-0010

FPGA update

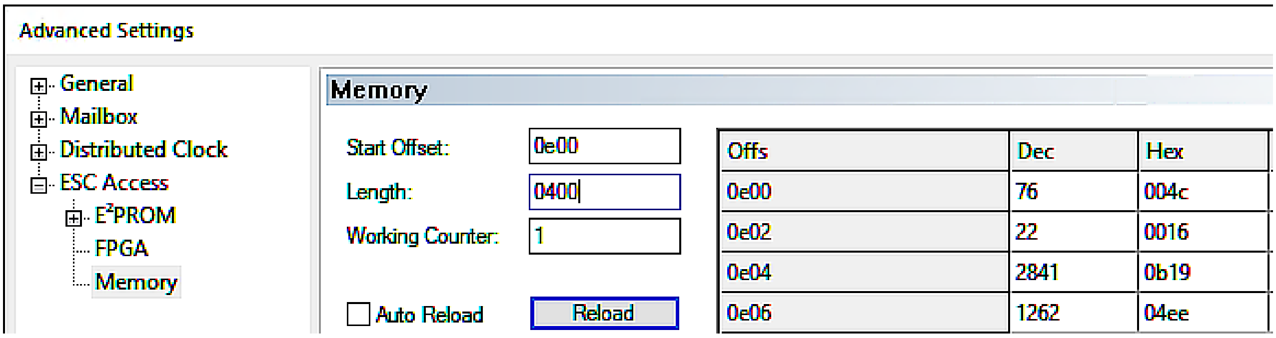
As FPGA zip file:



*select .zip file, takes several minutes, note progress bar  , runs through twice

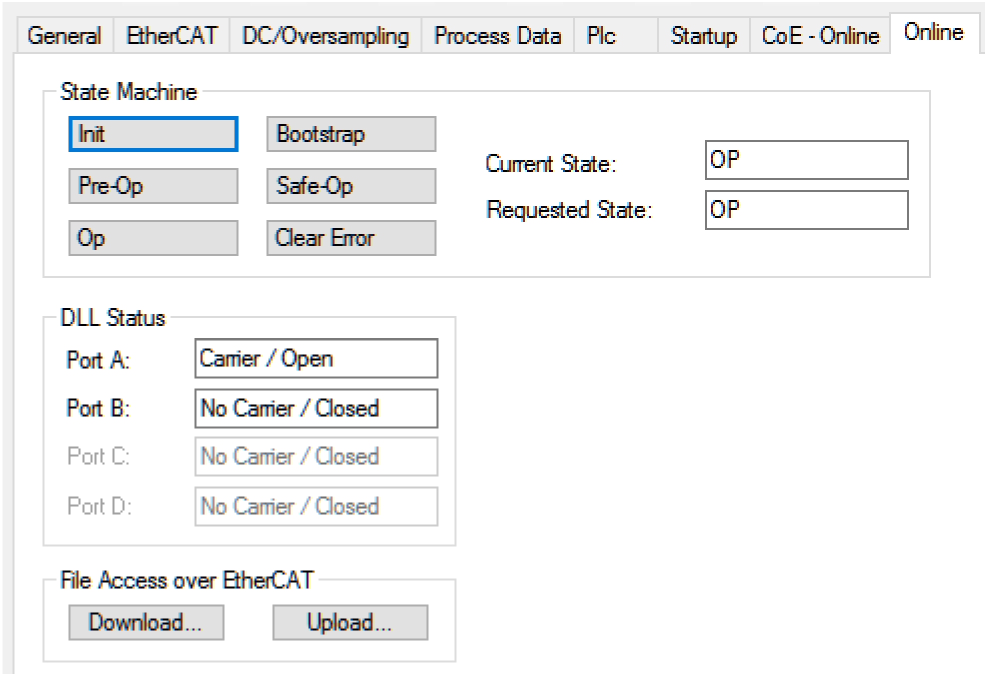
Ok: 

The FPGA state can be viewed in register x0E00



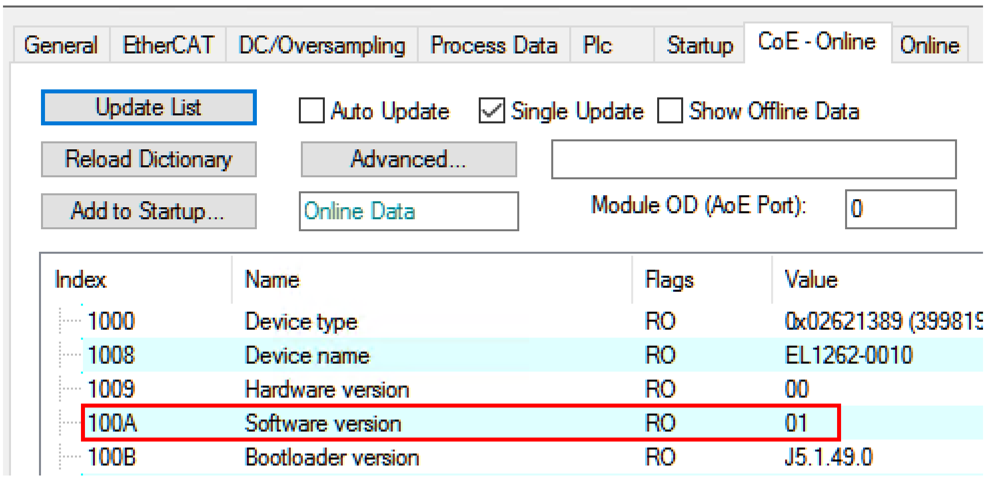
µC update

The µC firmware update has to be done as usual, e.g. via the dialog



Procedure: Set device to bootstrap, load *.efw file into device.

The current FW status can be viewed in the CoE directory



7.5 Restoring the delivery state

To restore the delivery state (factory settings) of CoE objects for EtherCAT devices ("slaves"), the CoE object *Restore default parameters*, SubIndex 001 can be used via EtherCAT master (e.g. TwinCAT) (see Fig. *Selecting the Restore default parameters PDO*).

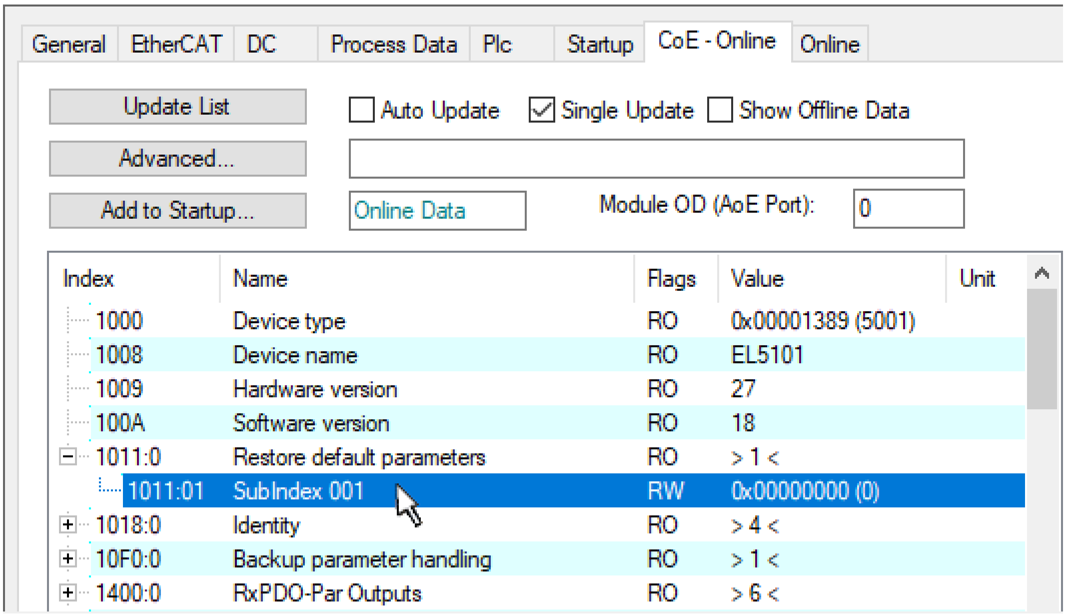


Fig. 230: Selecting the *Restore default parameters* PDO

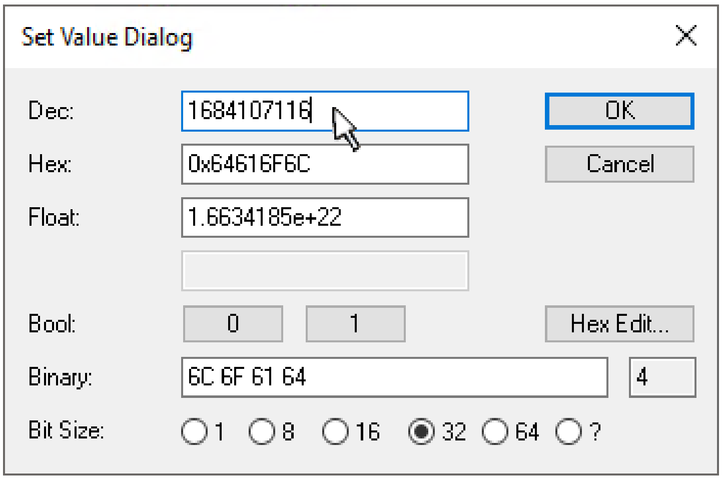


Fig. 231: Entering a restore value in the Set Value dialog

Double-click on *SubIndex 001* to enter the Set Value dialog. Enter the reset value **1684107116** in field *Dec* or the value **0x64616F6C** in field *Hex* (ASCII: "load") and confirm with *OK* (Fig. *Entering a restore value in the Set Value dialog*).

- All changeable entries in the slave are reset to the default values.
- The values can only be successfully restored if the reset is directly applied to the online CoE, i.e. to the slave. No values can be changed in the offline CoE.
- TwinCAT must be in the RUN or CONFIG/Freerun state for this; that means EtherCAT data exchange takes place. Ensure error-free EtherCAT transmission.
- No separate confirmation takes place due to the reset. A changeable object can be manipulated beforehand for the purposes of checking.
- This reset procedure can also be adopted as the first entry in the startup list of the slave, e.g. in the state transition PREOP->SAFEOP or, as in Fig. *CoE reset as a startup entry*, in SAFEOP->OP.

All backup objects are reset to the delivery state.

Alternative restore value

In some older terminals (FW creation approx. before 2007) the backup objects can be switched with an alternative restore value: Decimal value: 1819238756, Hexadecimal value: 0x6C6F6164.

An incorrect entry for the restore value has no effect.

7.6 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.

Support

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

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

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

Service

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

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

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

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

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

Trademark statements

Beckhoff®, 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.

Third-party trademark statements

DeviceNet and EtherNet/IP are trademarks of ODVA, Inc.

DSP System Toolbox, Embedded Coder, MATLAB, MATLAB Coder, MATLAB Compiler, MathWorks, Predictive Maintenance Toolbox, Simscape, Simscape™ Multibody™, Simulink, Simulink Coder, Stateflow and ThingSpeak are registered trademarks of The MathWorks, Inc.

Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

More Information:
www.beckhoff.com/el1xxx

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