

BECKHOFF New Automation Technology

Documentation | EN

EL2595

1-channel LED constant current terminal



Table of contents

1 Foreword	5
1.1 Notes on the documentation.....	5
1.2 Safety instructions	6
1.3 Documentation issue status	7
1.4 Version identification of EtherCAT devices	7
1.4.1 Beckhoff Identification Code (BIC).....	10
2 Product overview	12
2.1 Introduction.....	12
2.2 Technical data	13
2.3 Start	13
3 Basics communication	14
3.1 EtherCAT basics.....	14
3.2 EtherCAT cabling – wire-bound.....	14
3.3 General notes for setting the watchdog.....	15
3.4 EtherCAT State Machine	17
3.5 CoE Interface.....	19
3.6 Distributed Clock	24
4 Mounting and wiring	25
4.1 Installation on mounting rails	25
4.2 Connection system	27
4.3 Positioning of passive Terminals	30
4.4 Prescribed installation position	31
4.5 UL notice	33
4.6 LEDs and connection	34
5 Commissioning	36
5.1 TwinCAT Quick Start	36
5.1.1 TwinCAT 2	39
5.1.2 TwinCAT 3	49
5.2 TwinCAT Development Environment	62
5.2.1 Installation of the TwinCAT real-time driver.....	63
5.2.2 Notes regarding ESI device description.....	68
5.2.3 TwinCAT ESI Updater	72
5.2.4 Distinction between Online and Offline	72
5.2.5 OFFLINE configuration creation	73
5.2.6 ONLINE configuration creation	78
5.2.7 EtherCAT subscriber configuration.....	86
5.2.8 Import/Export of EtherCAT devices with SCI and XTI	95
5.3 Quick start.....	101
5.4 General Notes - EtherCAT Slave Application	103
5.5 Basic Function Principles	112
5.5.1 General procedure.....	112
5.5.2 Normal digital output.....	113
5.5.3 External trigger input.....	115

5.5.4	Distributed Clocks controlled output	116
5.5.5	Emergency Flash	116
5.5.6	Operating hour and switching cycle counter	116
5.6	Process data	116
5.6.1	Preselection of process data	117
5.6.2	Explanation of the process data	117
5.6.3	Process data overview	121
5.7	Distributed Clocks settings	125
5.7.1	Basic principles	125
5.7.2	Determination of the current Distributed Clock time	126
5.7.3	Process data	129
5.7.4	Example of a sequence with activated Distributed Clocks	129
5.8	Object description and parameterization	131
5.8.1	Profile-specific objects	131
5.8.2	Standard objects	136
5.9	CoE data	142
6	Diagnostics	147
6.1	Diagnostics – basic principles of diag messages	147
6.2	Diagnostics - specific diag messages - EL2595	156
7	Appendix	158
7.1	Firmware compatibility	158
7.2	Firmware Update EL/ES/EM/ELM/EPxxxx	158
7.2.1	Device description ESI file/XML	160
7.2.2	Firmware explanation	163
7.2.3	Updating controller firmware *.efw	164
7.2.4	FPGA firmware *.rbf	165
7.2.5	Simultaneous updating of several EtherCAT devices	169
7.3	Restoring the delivery state	170
7.4	Support and Service	171

1 Foreword

1.1 Notes on the documentation

Intended audience

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

It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

The logo for EtherCAT, featuring the word "EtherCAT" in a bold, black, sans-serif font. A red arrow points from the top of the "A" towards the right, ending above the "T". A registered trademark symbol (®) is located to the right of the "T".

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.

Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

DANGER

Serious risk of injury!

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

WARNING

Risk of injury!

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

CAUTION

Personal injuries!

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

NOTE

Damage to environment/equipment or data loss

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



Tip or pointer

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
1.2	- Update chapter "Technical data" - Update structure
1.1	- Update chapter "Technical data" - Update structure - Update revisions status
1.0	- 1 st public issue - Complements and corrections - Update structure
0.5	- Migration
0.4	- Complements and corrections
0.3	- Complements and corrections
0.2	- Complements and corrections
0.1	- Provisional documentation for EL2595

1.4 Version identification of EtherCAT devices

Designation

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

- family key
- type
- version
- revision

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

Notes

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

- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

Identification number

Beckhoff EtherCAT devices from the different lines have different kinds of identification numbers:

Production lot/batch number/serial number/date code/D number

The serial number 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**

KK - week of production (CW, calendar week)

YY - year of production

FF - firmware version

HH - hardware version

Example with

Ser. no.: 12063A02: 12 - production week 06 - production year 2006 3A - firmware version 3A 02 - hardware version 02

Exceptions can occur in the **IP67 area**, where the following syntax can be used (see respective device documentation):

Syntax: D ww yy x y z u

D - prefix designation

ww - calendar week

yy - year

x - firmware version of the bus PCB

y - hardware version of the bus PCB

z - firmware version of the I/O PCB

u - hardware version of the I/O PCB

Example: D.22081501 calendar week 22 of the year 2008 firmware version of bus PCB: 1 hardware version of bus PCB: 5 firmware version of I/O PCB: 0 (no firmware necessary for this PCB) hardware version of I/O PCB: 1

Examples of markings



Fig. 1: EL5021 EL terminal, standard IP20 IO device with serial/ batch number and revision ID (since 2014/01)



Fig. 2: EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number

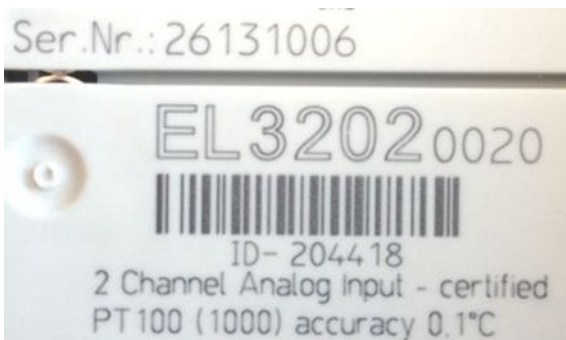


Fig. 3: EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418

1.4.1 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.

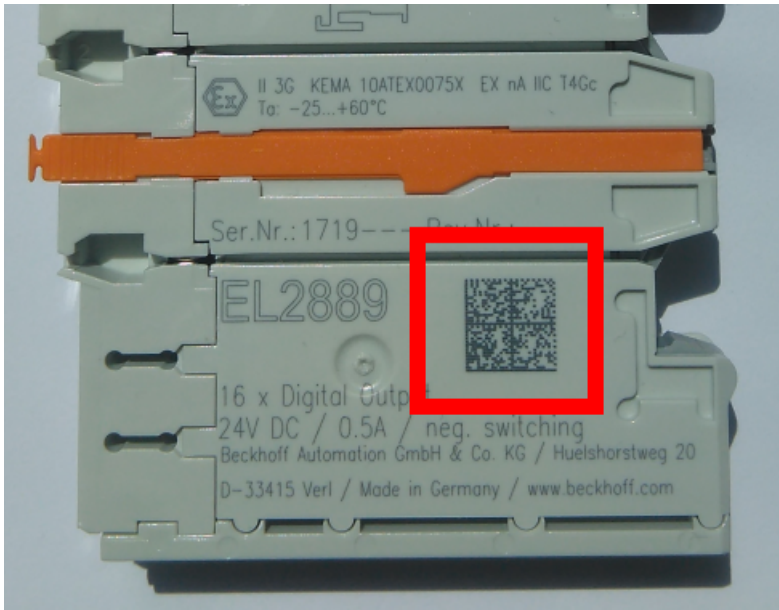


Fig. 4: BIC as data matrix code (DMC, code scheme ECC200)

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. The data under positions 1 to 4 are always available.

The following information is contained:

Item no.	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	Beckhoff order number	1P	8	1P 072222
2	Beckhoff Traceability Number (BTN)	Unique serial number, see note below	S	12	S BTNk4p562d7
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 or calibrated terminals	51S	12	51S 678294104
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	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 item 1 to 4 and 6. The data identifiers are marked in red for better display:

BTN

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

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

2 Product overview

2.1 Introduction

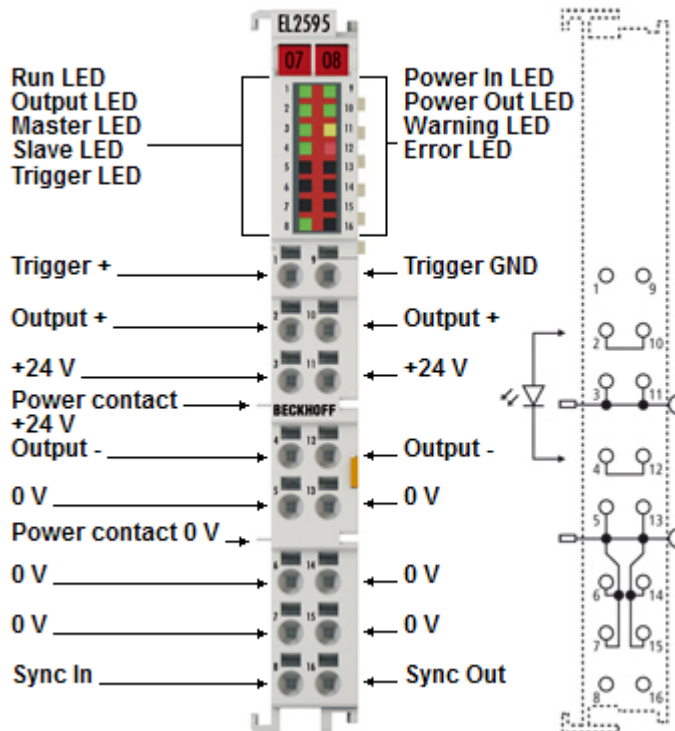


Fig. 5: EL2595

1-channel LED constant current source

The EL2595 EtherCAT Terminal contains a constant current source for LEDs and is designed to control one or more LEDs connected in series.

The user specifies the required set current, the integrated power supply unit then provides the required forward voltage, depending on the connected LEDs. For this purpose, the EL2595 has a step-up/step-down power supply unit, which generates the output voltage from the 24 V_{DC} input voltage. The output voltage is also adjusted during operation in order to maintain the setpoint current.

The current can be switched quickly for short-term lighting, hence even extremely short flashes of light are possible. The pulse duration is adjustable from 200 μs to endless. As with the EL2252, the flash time itself can be set by a distributed clock timestamp; however, an external trigger input is also available. Extensive real-time diagnosis functions for input current/voltage and output current/voltage enable detailed control of the LED light intensity. If a definable load corridor is left, e.g. due to a load error, the EL2595 switches off after a warning to protect the load. This can be reset.

Quick links

- [EtherCAT basics](#)
- [Basic Function Principles \[► 112\]](#)
- [Quick start \[► 101\]](#)
- [Most important CoE entries \[► 142\]](#)
- [Process data \[► 116\]](#)

2.2 Technical data

Technical data	EL2595
Application recommendation	Continuous light >300 mA and simple lighting applications
Connection technology	2-wire
Number of outputs	1
Input voltage (power contacts)	24 V _{DC} (-15 %/+20 %)
Rated load voltage	2 - 48 V _{DC} , (controlled automatically, observe the dead band in control mode)
Load type	resistive
Distributed clocks	yes
Distr. Clocks accuracy	<< 1 μs
Min. output current (continuous)	300 mA
Max. output current (continuous)	700 mA (short-circuit proof)
Switching times	200 μs to endless
Trigger input	max. 24 V, typ. 8 mA max., switching level approx. 3 V, reverse polarity up to max. 5 V maximum load
Current consumption via E-bus	typ. 110 mA
Electrical isolation	500 V (E-bus/field voltage)
Current consumption power contacts	typ. 20 mA + load
Special features	optional automatic operation in case of communication interruption, extensive real-time diagnostics, external trigger input
Power supply for the electronics	via the E-bus
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)
Mounting [▶ 25]	on 35 mm mounting rail according 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 class	IP20
Installation position	prescribed installation position - see note [▶ 31]!
Approval	CE

2.3 Start

For commissioning:

- Install the EL2595 as described in chapter [Installation \[\[▶ 25\]\(#\)\]](#)
- Configure the EL2595 in TwinCAT as described in chapter [Commissioning \[\[▶ 101\]\(#\)\]](#).

3 Basics communication

3.1 EtherCAT basics

Please refer to the [EtherCAT System Documentation](#) for the EtherCAT fieldbus basics.

3.2 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

- i** 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.

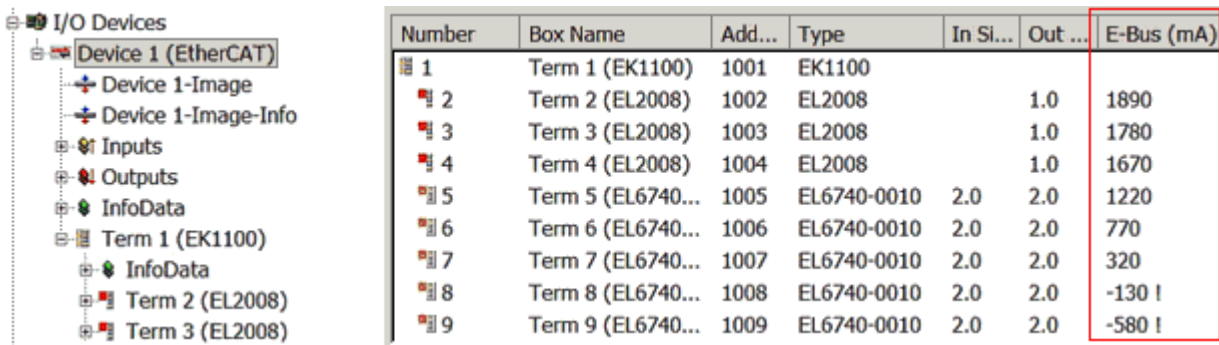


Fig. 6: System manager current calculation

NOTE

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

3.3 General notes for setting the watchdog

ELxxxx terminals are equipped with a safety feature (watchdog) that switches off the outputs after a specifiable time e.g. in the event of an interruption of the process data traffic, depending on the device and settings, e.g. in OFF state.

The EtherCAT slave controller (ESC) in the EL2xxx terminals features two watchdogs:

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

SM watchdog (SyncManager Watchdog)

The SyncManager watchdog is reset after each successful EtherCAT process data communication with the terminal. If no EtherCAT process data communication takes place with the terminal for longer than the set and activated SM watchdog time, e.g. in the event of a line interruption, the watchdog is triggered and the outputs are set to FALSE. The OP state of the terminal is unaffected. The watchdog is only reset after a successful EtherCAT process data access. Set the monitoring time as described below.

The SyncManager watchdog monitors correct and timely process data communication with the ESC from the EtherCAT side.

PDI watchdog (Process Data Watchdog)

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

PDI (Process Data Interface) is the internal interface between the ESC and local processors in the EtherCAT slave, for example. The PDI watchdog can be used to monitor this communication for failure.

The PDI watchdog monitors correct and timely process data communication with the ESC from the application side.

The settings of the SM- and PDI-watchdog must be done for each slave separately in the TwinCAT System Manager.

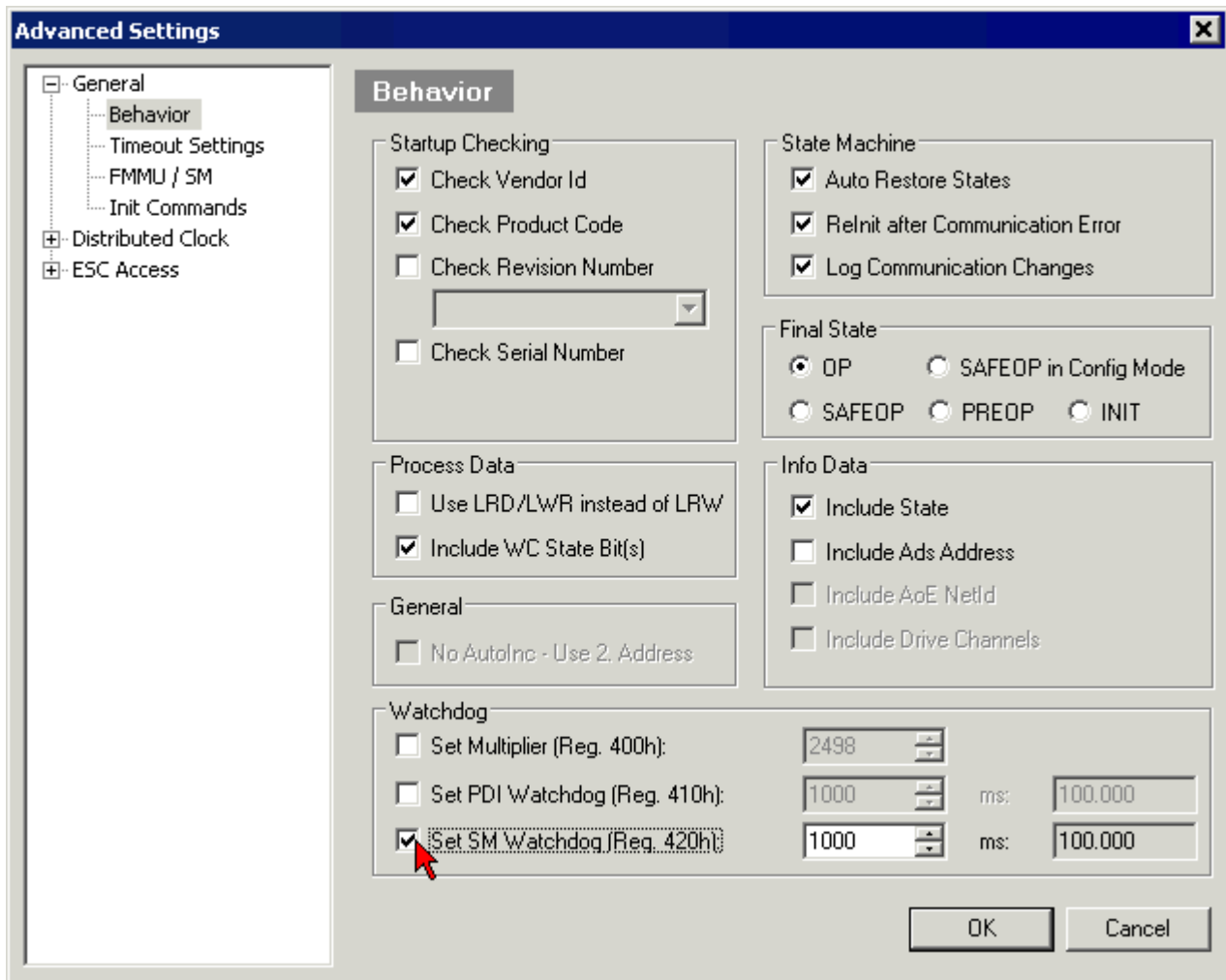


Fig. 7: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the multiplier is valid for both watchdogs.
- each watchdog has its own timer setting, the outcome of this in summary with the multiplier is a resulting time.
- Important: the multiplier/timer setting is only loaded into the slave at the start up, if the checkbox is activated.
If the checkbox is not activated, nothing is downloaded and the ESC settings remain unchanged.

Multiplier

Multiplier

Both watchdogs receive their pulses from the local terminal cycle, divided by the watchdog multiplier:

$$1/25 \text{ MHz} * (\text{watchdog multiplier} + 2) = 100 \text{ } \mu\text{s} \text{ (for default setting of 2498 for the multiplier)}$$

The standard setting of 1000 for the SM watchdog corresponds to a release time of 100 ms.

The value in multiplier + 2 corresponds to the number of basic 40 ns ticks representing a watchdog tick. The multiplier can be modified in order to adjust the watchdog time over a larger range.

Example "Set SM watchdog"

This checkbox enables manual setting of the watchdog times. If the outputs are set and the EtherCAT communication is interrupted, the SM watchdog is triggered after the set time and the outputs are erased. This setting can be used for adapting a terminal to a slower EtherCAT master or long cycle times. The default SM watchdog setting is 100 ms. The setting range is 0...65535. Together with a multiplier with a range of 1...65535 this covers a watchdog period between 0...~170 seconds.

Calculation

Multiplier = 2498 → watchdog base time = $1 / 25 \text{ MHz} * (2498 + 2) = 0.0001 \text{ seconds} = 100 \mu\text{s}$
SM watchdog = 10000 → $10000 * 100 \mu\text{s} = 1 \text{ second watchdog monitoring time}$

⚠ CAUTION

Undefined state possible!

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

⚠ CAUTION

Damage of devices and undefined state possible!

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

3.4 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 and
- Operational
- Boot

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

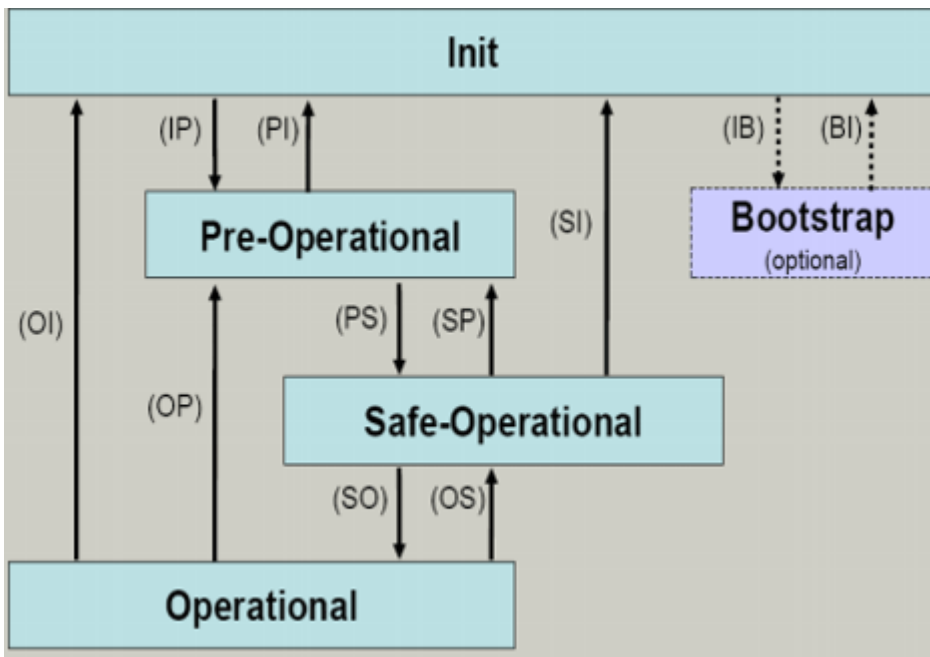


Fig. 8: States of the EtherCAT State Machine

Init

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

Pre-Operational (Pre-Op)

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

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

Safe-Operational (Safe-Op)

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

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

● **Outputs in SAFEOP state**

i The default set `watchdog [▶ 15]` monitoring sets the outputs of the module in a safe state - depending on the settings in `SAFEOP` and `OP` - e.g. in `OFF` state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the `SAFEOP` state.

Operational (Op)

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

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

Boot

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

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

3.5 CoE Interface

General description

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

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

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

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

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

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

● Availability

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

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

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

Fig. 9: "CoE Online" tab

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

Data management and function "NoCoeStorage"

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

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

i Data management

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

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

Please refer to the technical data in this documentation as to whether this applies to the respective device.

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

i 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.

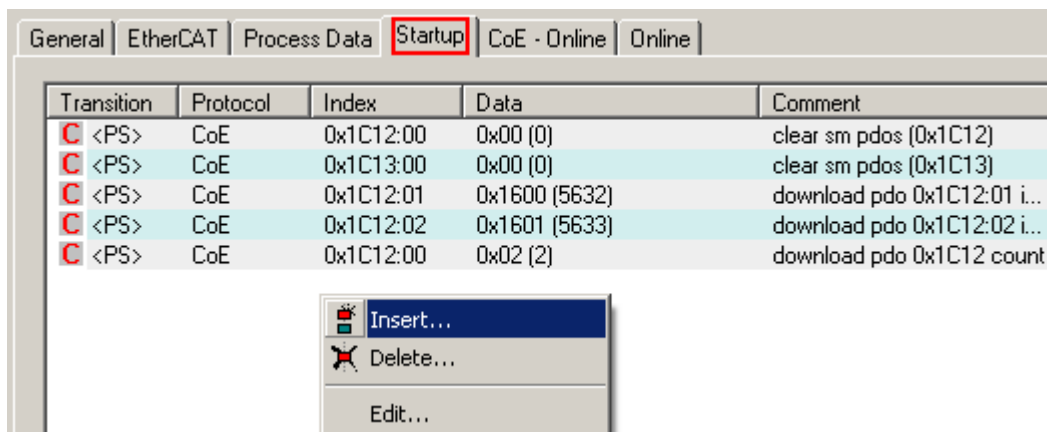


Fig. 10: Startup list in the TwinCAT System Manager

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

Online/offline list

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

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

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

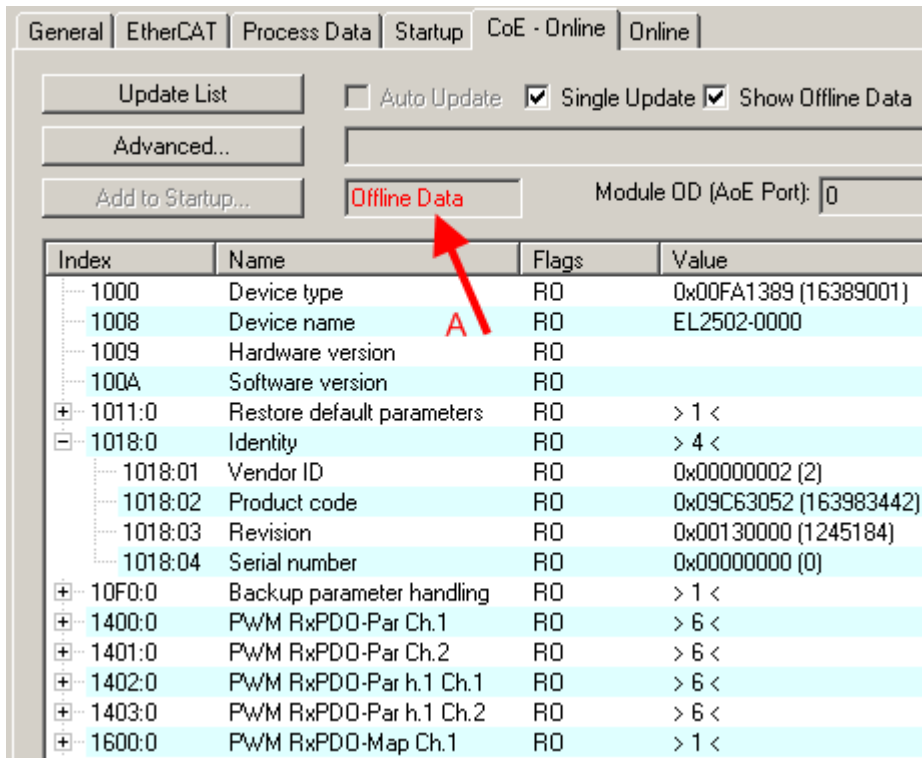


Fig. 11: Offline list

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

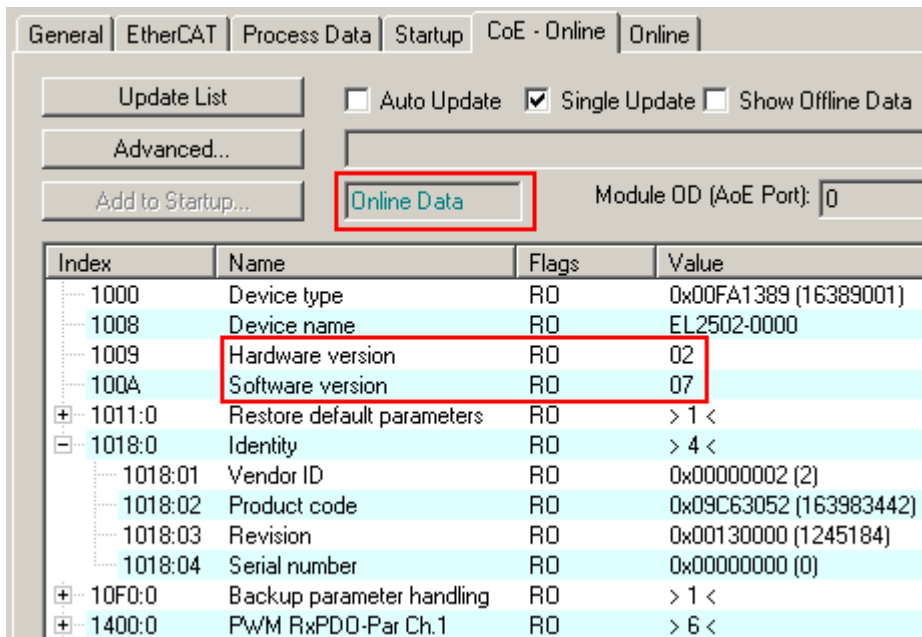


Fig. 12: Online list

Channel-based order

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

In the CoE system 16 indices, each with 255 subindices, are generally sufficient for representing all channel parameters. The channel-based order is therefore arranged in $16_{\text{dec}}/10_{\text{hex}}$ 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.

3.6 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](#).

4 Mounting and wiring

4.1 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!

Assembly

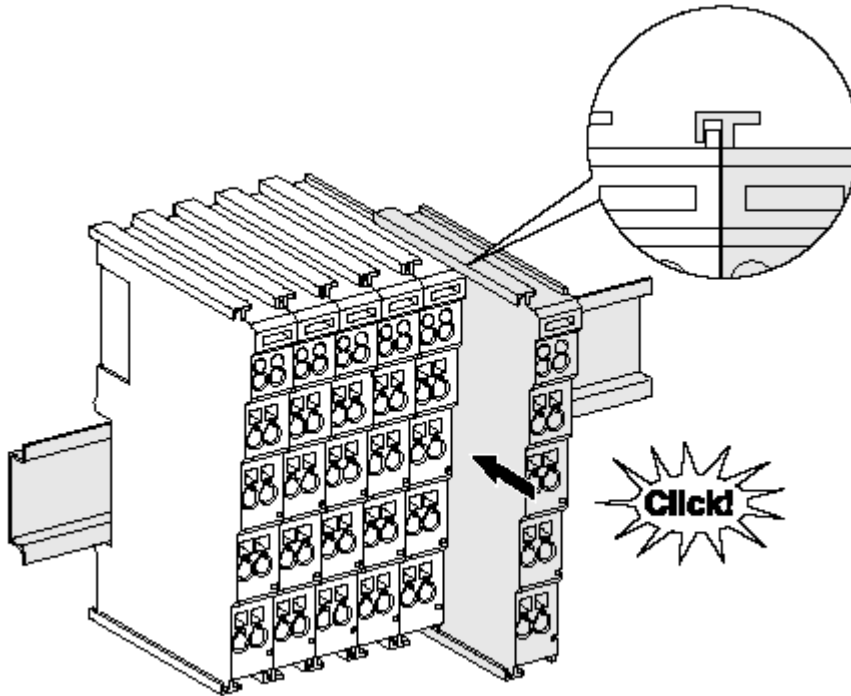


Fig. 13: Attaching on mounting rail

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.

i 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

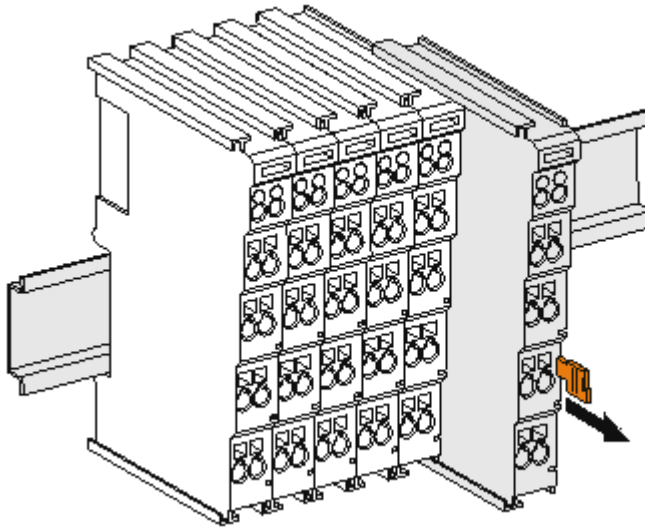


Fig. 14: Disassembling of terminal

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 on the Bus Coupler (up to 24 V) or for higher voltages via power feed terminals.

i 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.

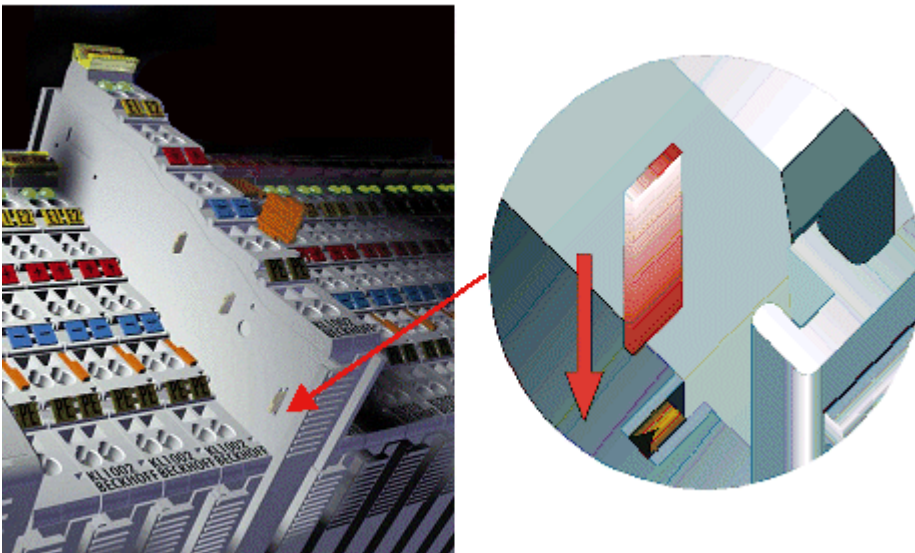


Fig. 15: Power contact on left side

NOTE

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!

4.2 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 KLxxxx and ELxxxx series with standard wiring include electronics and connection level in a single enclosure.
- The terminals of KSxxxx and ESxxxx 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



Fig. 16: Standard wiring

The terminals of KLxxxx and ELxxxx series have been tried and tested for years. They feature integrated screwless spring force technology for fast and simple assembly.

Pluggable wiring

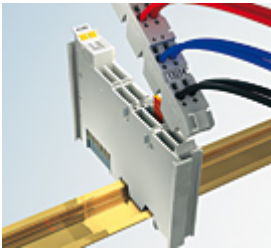


Fig. 17: Pluggable wiring

The terminals of KSxxxx and ESxxxx series feature a pluggable connection level. The assembly and wiring procedure for the KS series is the same as for the KLxxxx and ELxxxx series. The KS/ES series terminals enable 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 mm² and 2.5 mm² can continue to be used with the proven spring force technology.

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

High Density Terminals (HD Terminals)



Fig. 18: High Density Terminals

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

i **Wiring HD Terminals**

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

Ultrasonically "bonded" (ultrasonically welded) conductors

i **Ultrasonically "bonded" conductors**

It is also possible to connect the Standard and High Density terminals with ultrasonically "bonded" (ultrasonically welded) conductors. In this case, please note the tables concerning the wire-size width [► 29] below!

Wiring

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

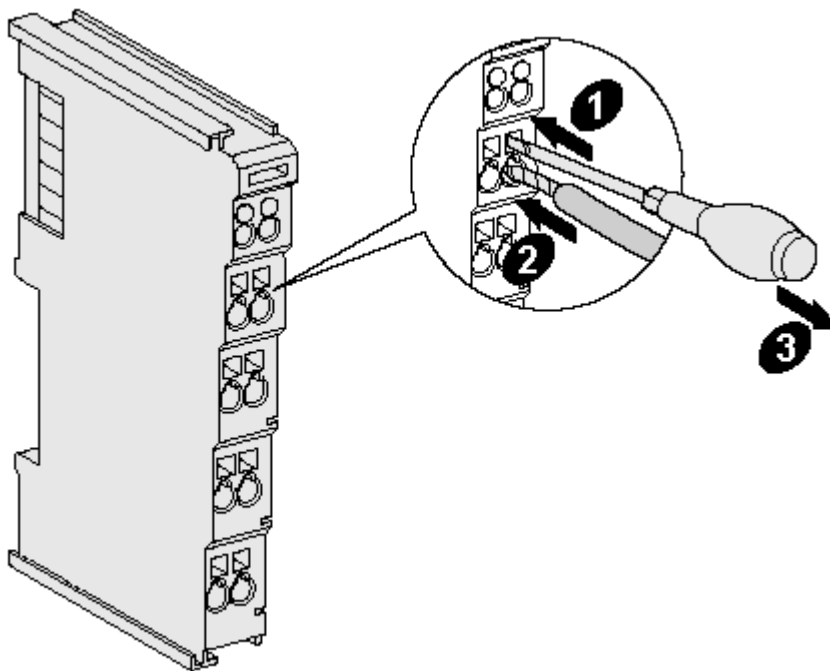


Fig. 19: Mounting a cable on a terminal connection

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

1. Open a spring-loaded terminal by slightly pushing with a screwdriver or a rod into the square opening above the terminal.
2. The wire can now be inserted into the round terminal opening without any force.
3. The terminal closes automatically when the pressure is released, holding the wire securely and permanently.

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width	0.08 ... 2,5 mm ²	0.08 ... 2.5 mm ²
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals ELx8xx, KLx8xx (HD)

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 contact 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 (conductors with a wire end sleeve)	0.14 ... 0.75 mm ²
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 (ultrasonically "bonded" conductors)	only 1.5 mm ² (see notice [▶_29]!)
Wire stripping length	8 ... 9 mm

Shielding

i **Shielding**

Analog sensors and actors should always be connected with shielded, twisted paired wires.

4.3 Positioning of passive Terminals

i **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)

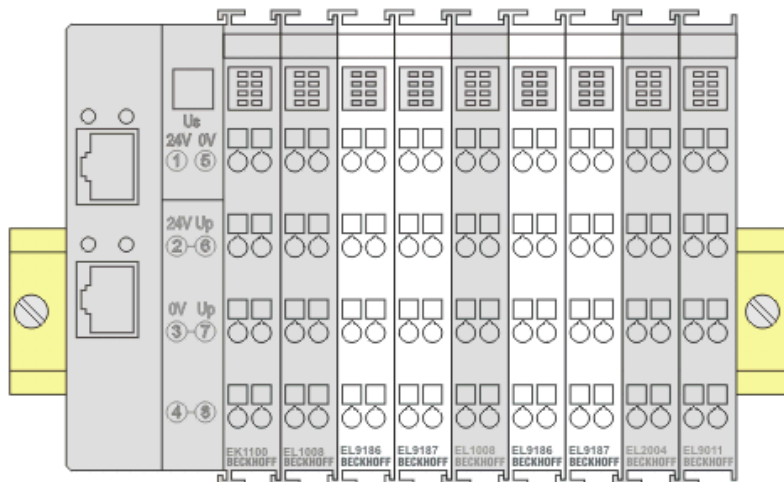


Fig. 20: Correct positioning

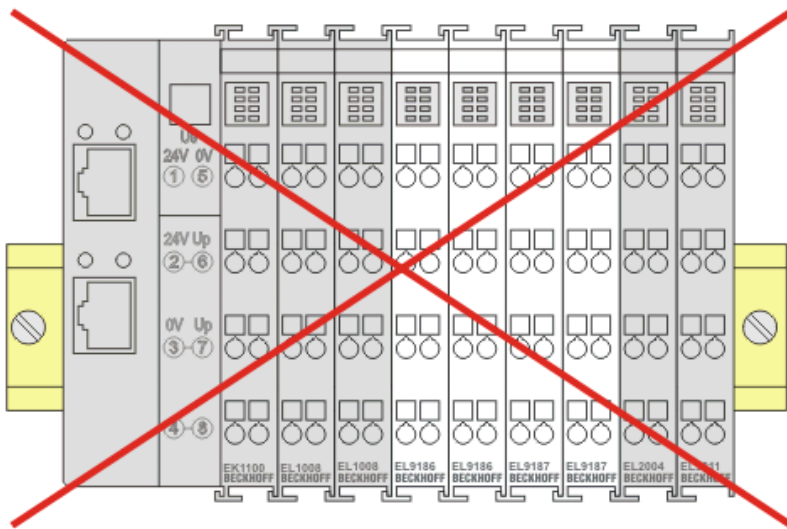


Fig. 21: Incorrect positioning

4.4 Prescribed installation position

NOTE

Constraints regarding installation position and operating temperature range

When installing the terminals ensure that an adequate spacing is maintained between other components above and below the terminal in order to guarantee adequate ventilation!

Prescribed installation position

The prescribed 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.

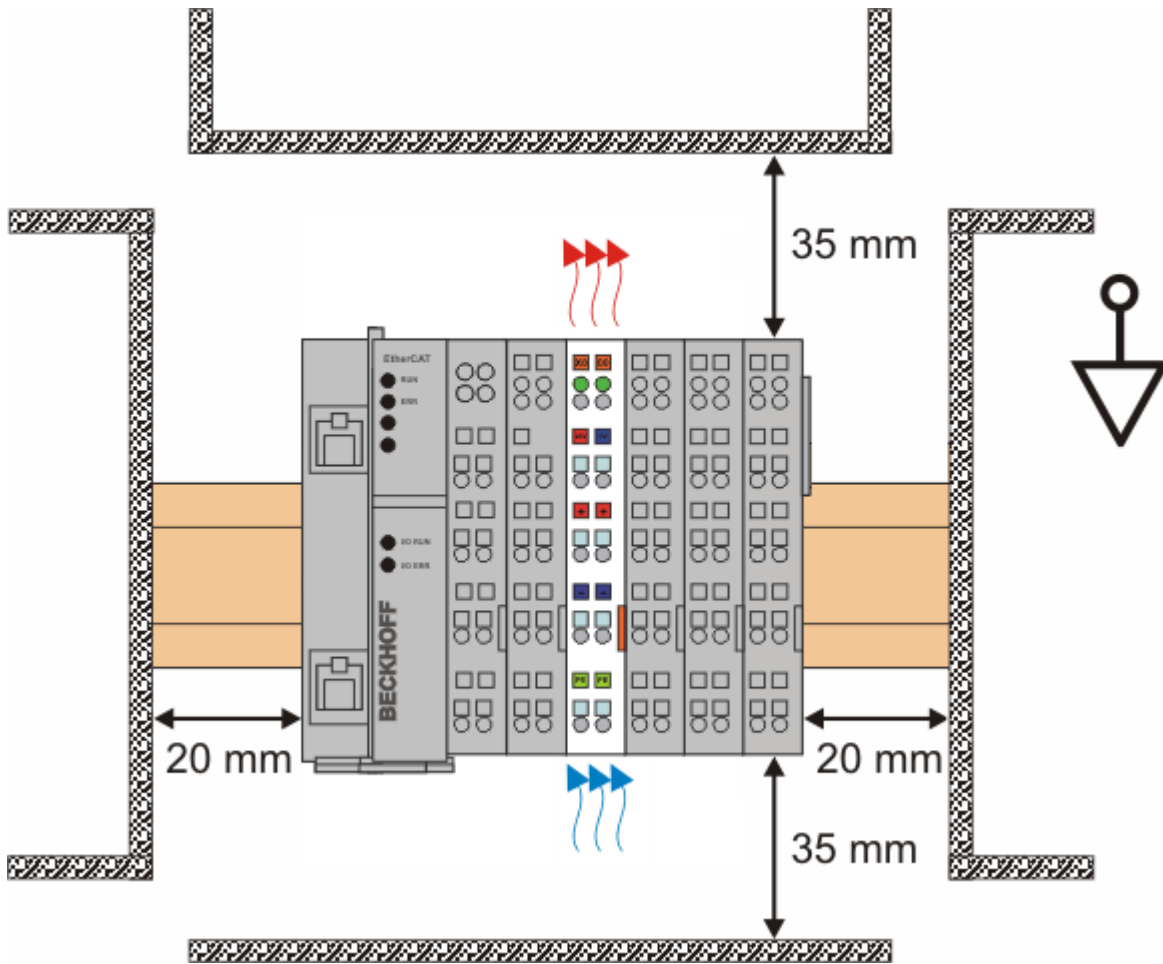





Fig. 22: Recommended minimum distances for standard installation position

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

4.5 UL notice

	<p>Application Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.</p>
	<p>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).</p>
	<p>For devices with Ethernet connectors Not for connection to telecommunication circuits.</p>

Basic principles

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



4.6 LEDs and connection

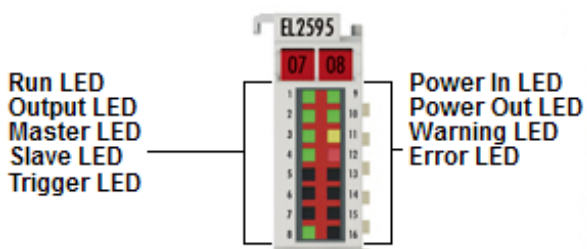


Fig. 23: EL2595 LEDs

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: INIT = initialization of the terminal
		flashing	State of the EtherCAT State Machine: PREOP = function for mailbox communication and different default settings set
		Single flash	State of the EtherCAT State Machine: SAFEOP = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: OP = normal operating state; mailbox and process data communication is possible
		flickering	State of the EtherCAT State Machine: BOOTSTRAP = function for terminal firmware updates
OUTPUT	green	off	Output is disabled or EL2595 is not ready to operate
		on	Output is enabled and EL2595 is ready to operate
MASTER	green	Output stage is configured as master mode (slave mode is not supported in the current firmware)	
SLAVE	green	reserved	
TRIGGER	green	Trigger input is triggered externally	
POWER IN	green	off	Input voltage is outside the configured range
		on	Input voltage is inside the configured range
POWER OUT	green	off	Output voltage is outside the configured range
		on	Output voltage is inside the configured range
WARNING	yellow	off	no defect
		on	- Supply voltage is outside the configured range - Output voltage is outside the configured range - 80 °C temperature exceeded ...
ERROR	red	off	no defect
		on	Configuration error, e.g.: - no supply voltage connected - 100 °C temperature exceeded - short-circuit ...

EL2595 pin assignment

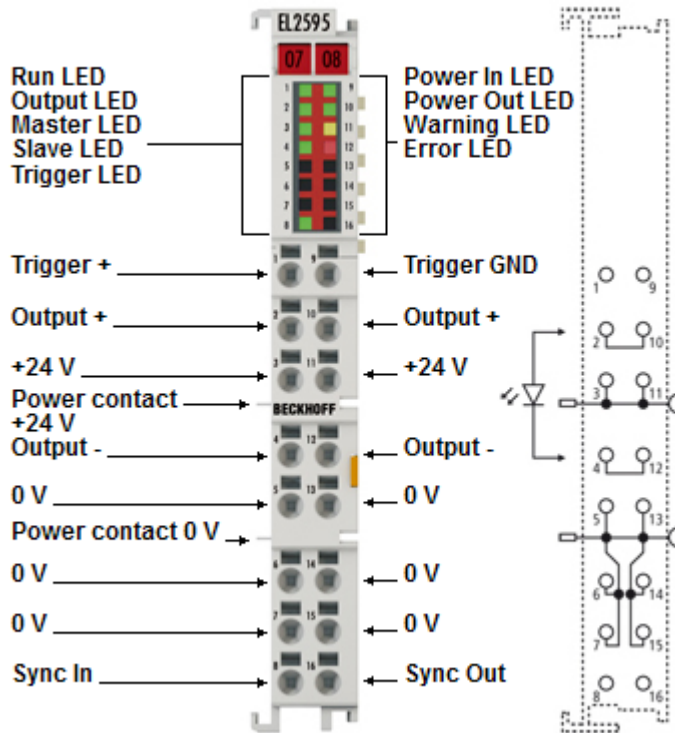


Fig. 24: EL2595 pin assignment

Terminal point	No.	Comment
Trigger +	1	Trigger input + ()
Output +	2	Output + for load
+24 V	3	24 V control voltage (internally connected to positive power contact)
Output -	4	Output - for load
0 V	5	0 V control voltage (internally connected to negative power contact)
0 V	6	0 V control voltage (internally connected to negative power contact)
0 V	7	0 V control voltage (internally connected to negative power contact)
Sync In	8	reserved, connection may not be used
Trigger GND	9	Trigger input - (must be connected to GND of the same potential of Trigger +)
Output +	10	Output + for load
+24 V	11	24 V control voltage (internally connected to positive power contact)
Output -	12	Output - for load
0 V	13	0 V control voltage (internally connected to negative power contact)
0 V	14	0 V control voltage (internally connected to negative power contact)
0 V	15	0 V control voltage (internally connected to negative power contact)
Sync Out	16	reserved, connection may not be used

5 Commissioning

5.1 TwinCAT Quick Start

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

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

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

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

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

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

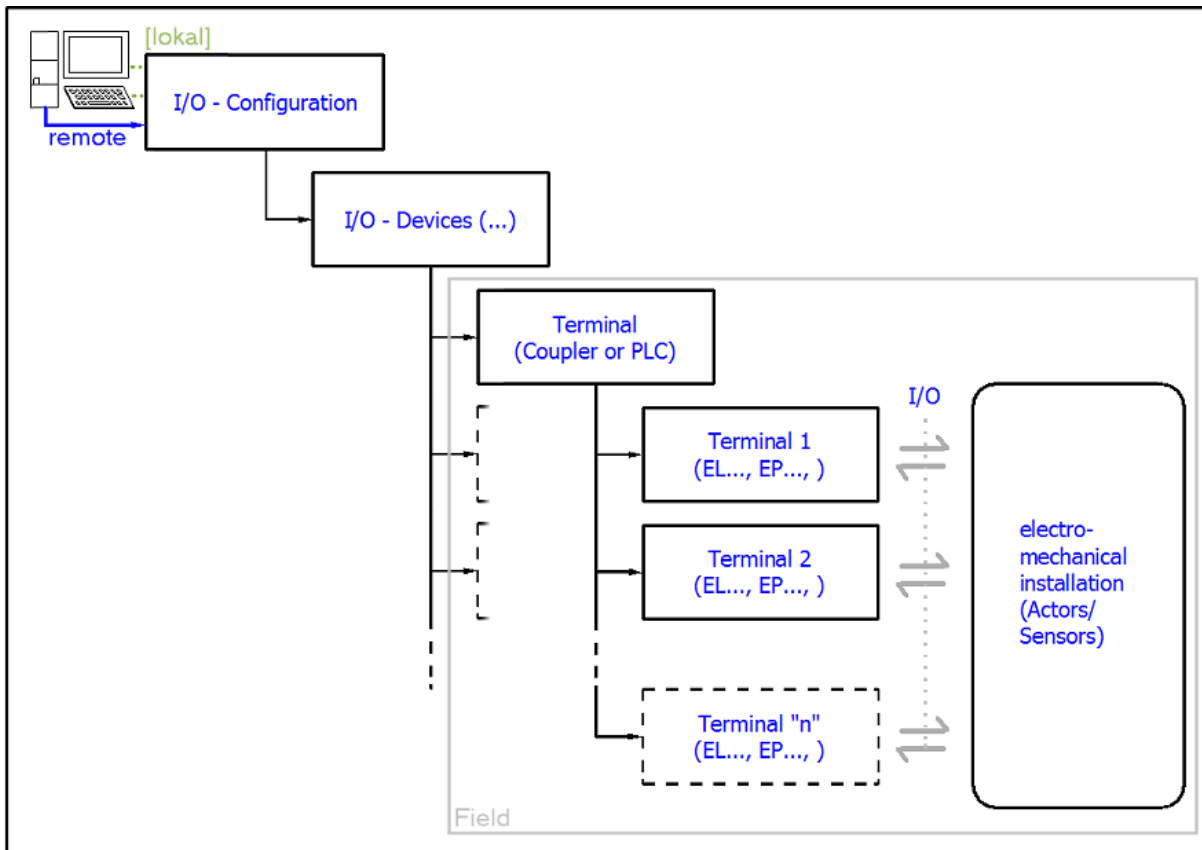


Fig. 25: Relationship between user side (commissioning) and installation

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

Sample configuration (actual configuration)

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

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

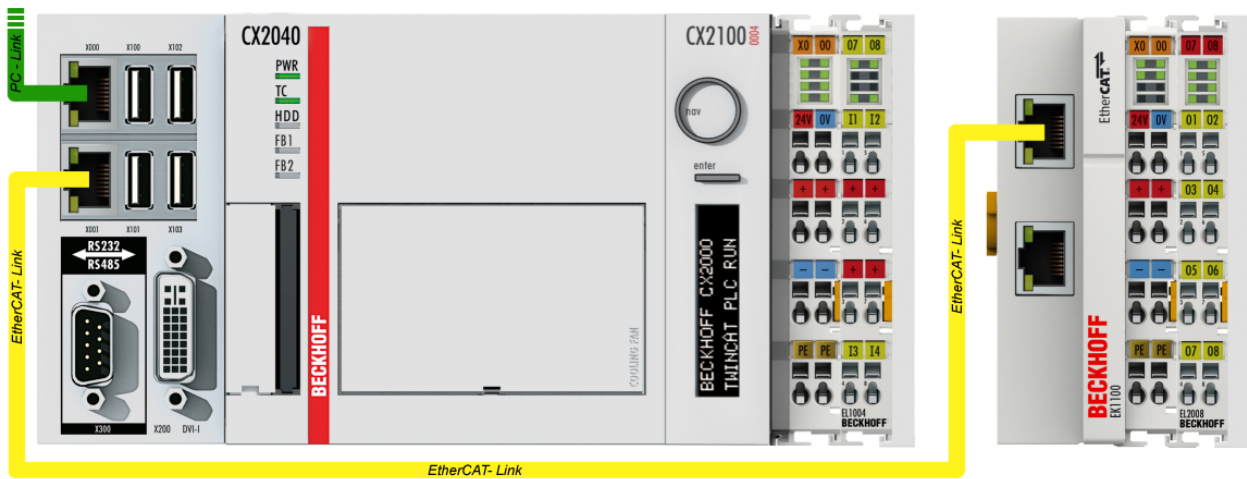


Fig. 26: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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

5.1.1 TwinCAT 2

Startup

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

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

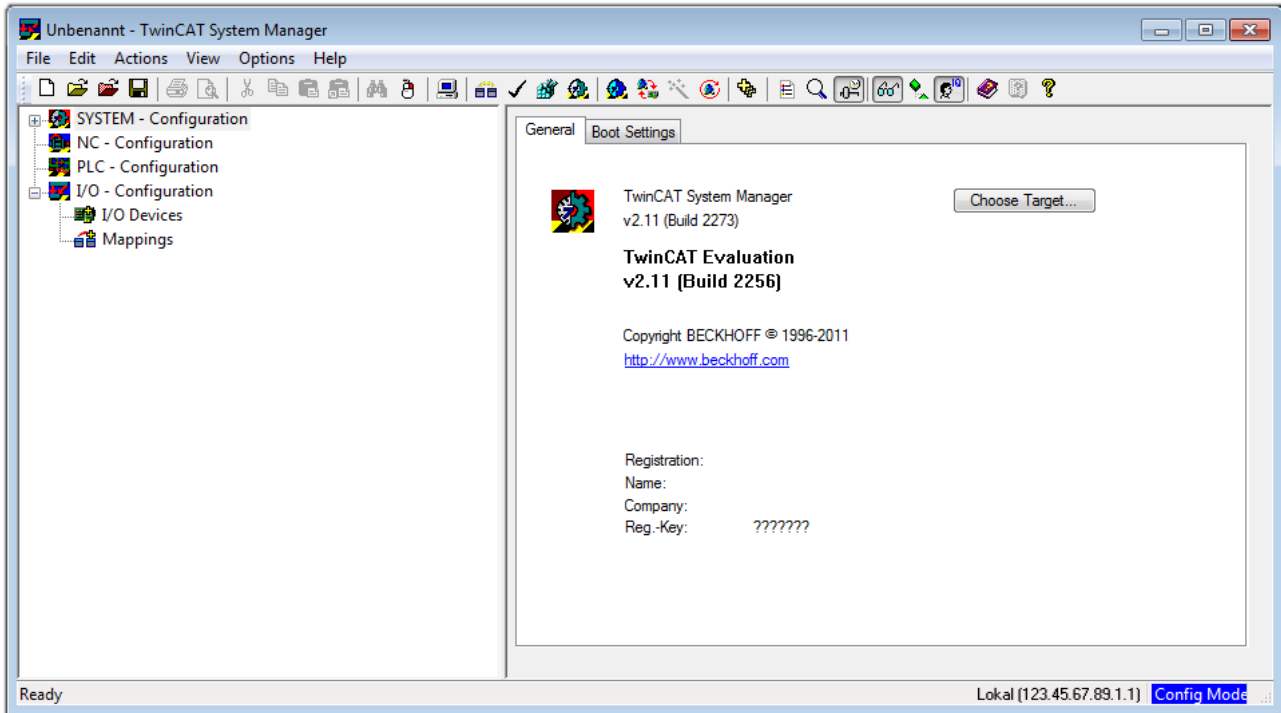


Fig. 27: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is “[Insert Device](#) [▶ 41]”.

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

“Actions” → “Choose Target System...”, via the symbol “” or the “F8” key, open the following window:

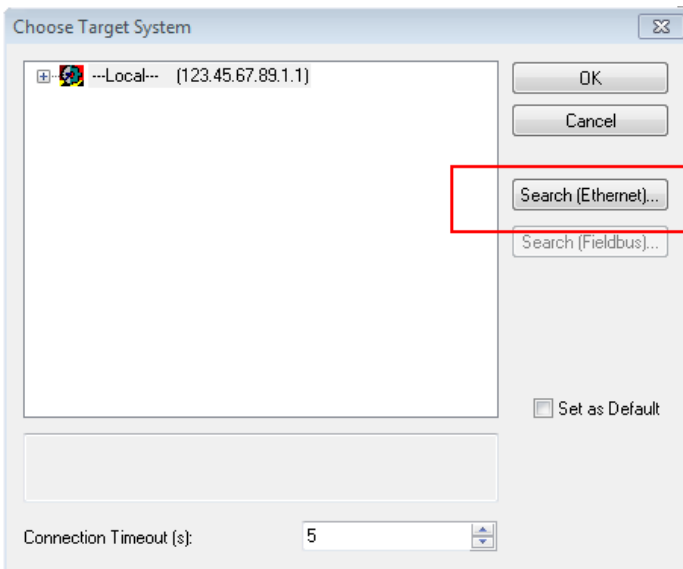


Fig. 28: Selection of the target system

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

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

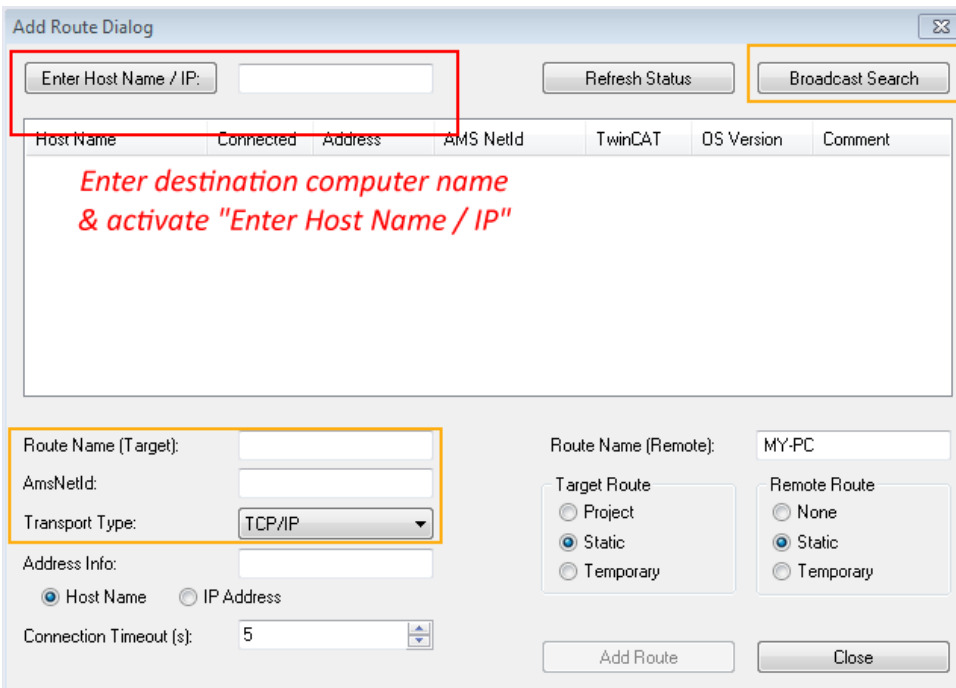
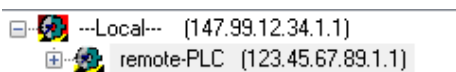


Fig. 29: Specify the PLC for access by the TwinCAT System Manager: selection of the target system



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



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

Adding devices

In the configuration tree of the TwinCAT 2 System Manager user interface on the left, select “I/O Devices” and then right-click to open a context menu and select “Scan Devices...”, or start the action in the menu bar

via . The TwinCAT System Manager may first have to be set to “Config mode” via  or via menu “Actions” → “Set/Reset TwinCAT to Config Mode...” (Shift + F4).

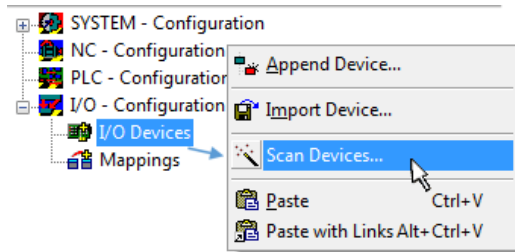


Fig. 30: Select “Scan Devices...”

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

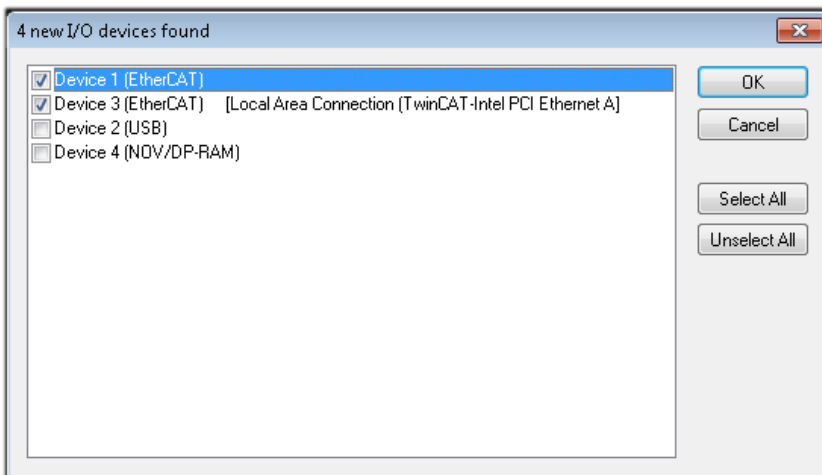


Fig. 31: Automatic detection of I/O devices: selection the devices to be integrated

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

Based on the [sample configuration](#) [▶ 37] described at the beginning of this section, the result is as follows:

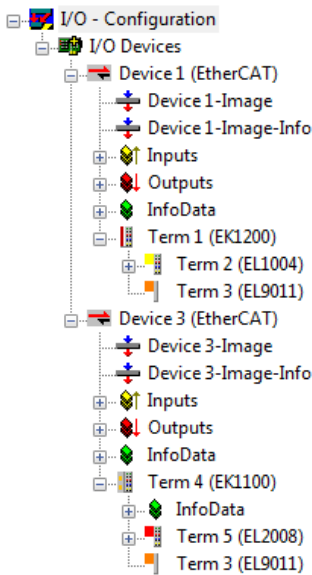


Fig. 32: Mapping of the configuration in the TwinCAT 2 System Manager

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

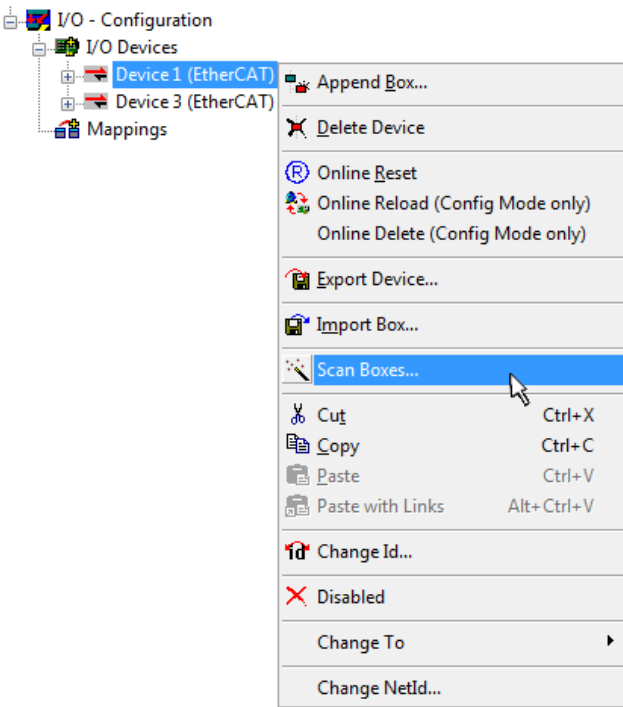


Fig. 33: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

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

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

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

The following section refers to Structured Text (ST).

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

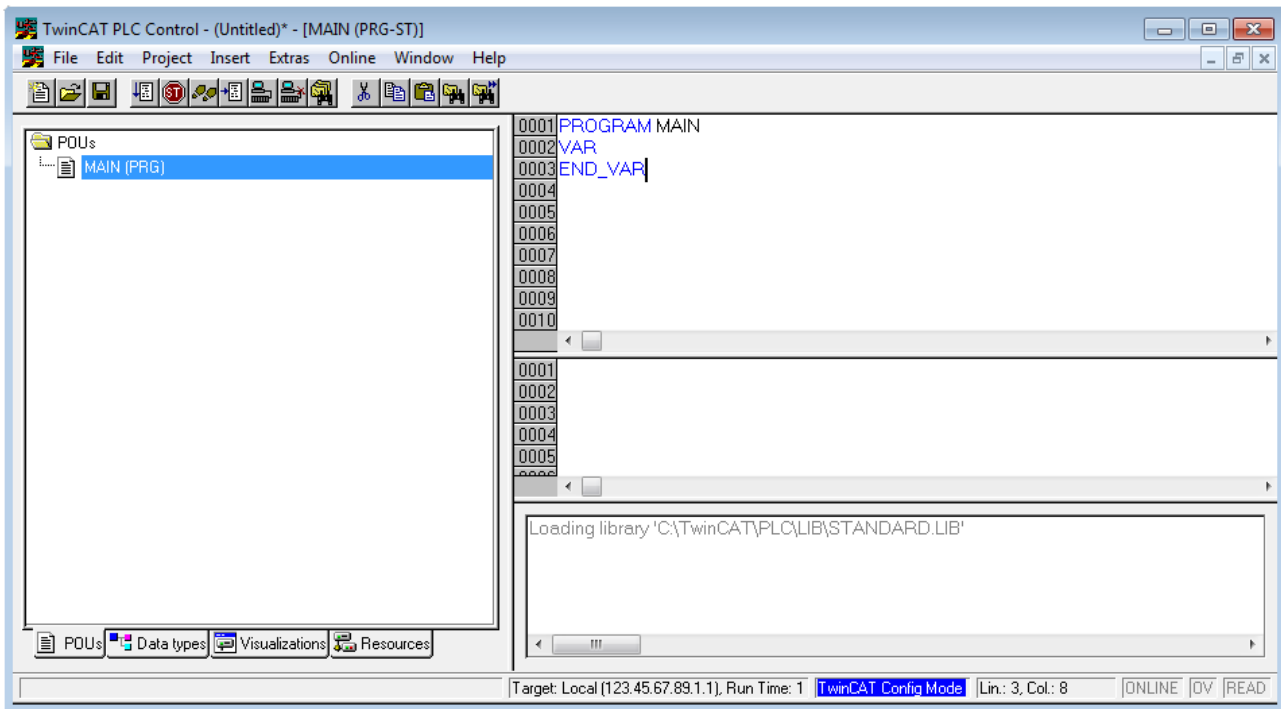


Fig. 34: TwinCAT PLC Control after startup

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

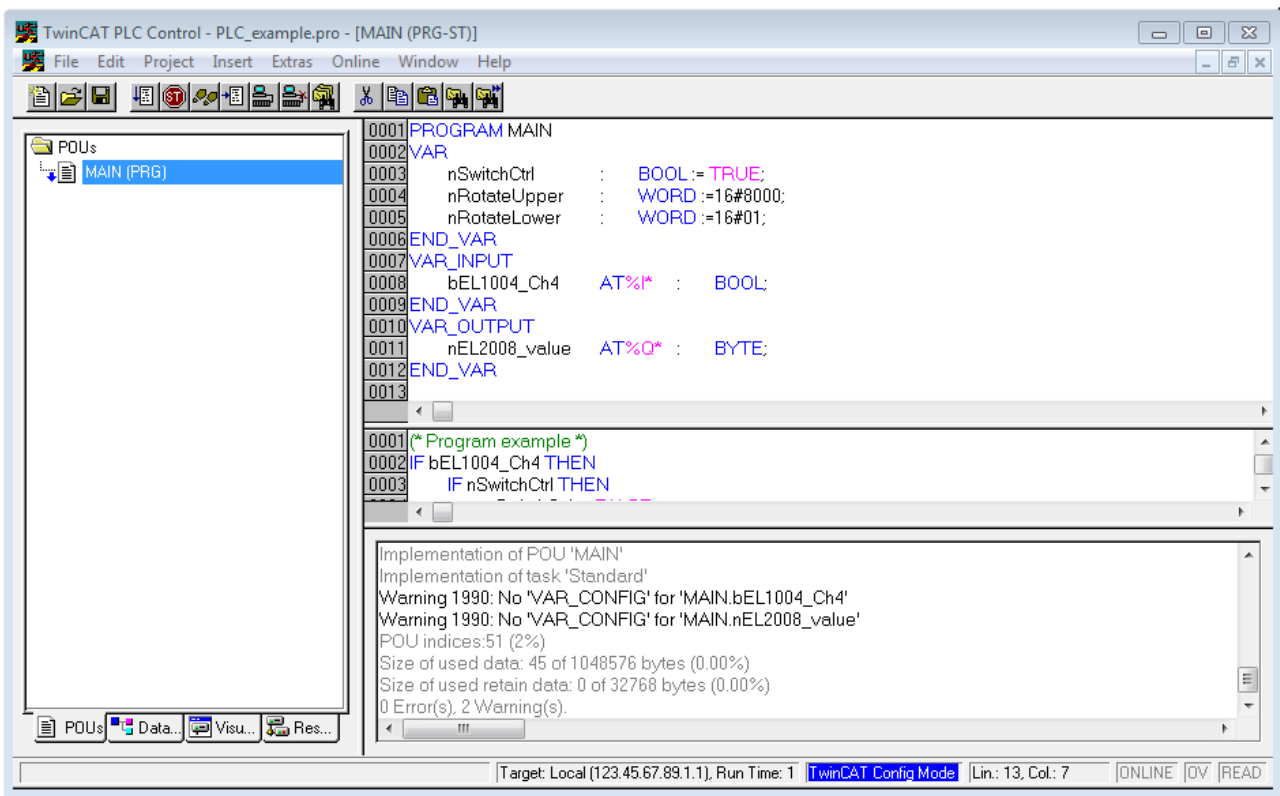


Fig. 35: Sample program with variables after a compile process (without variable integration)

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

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

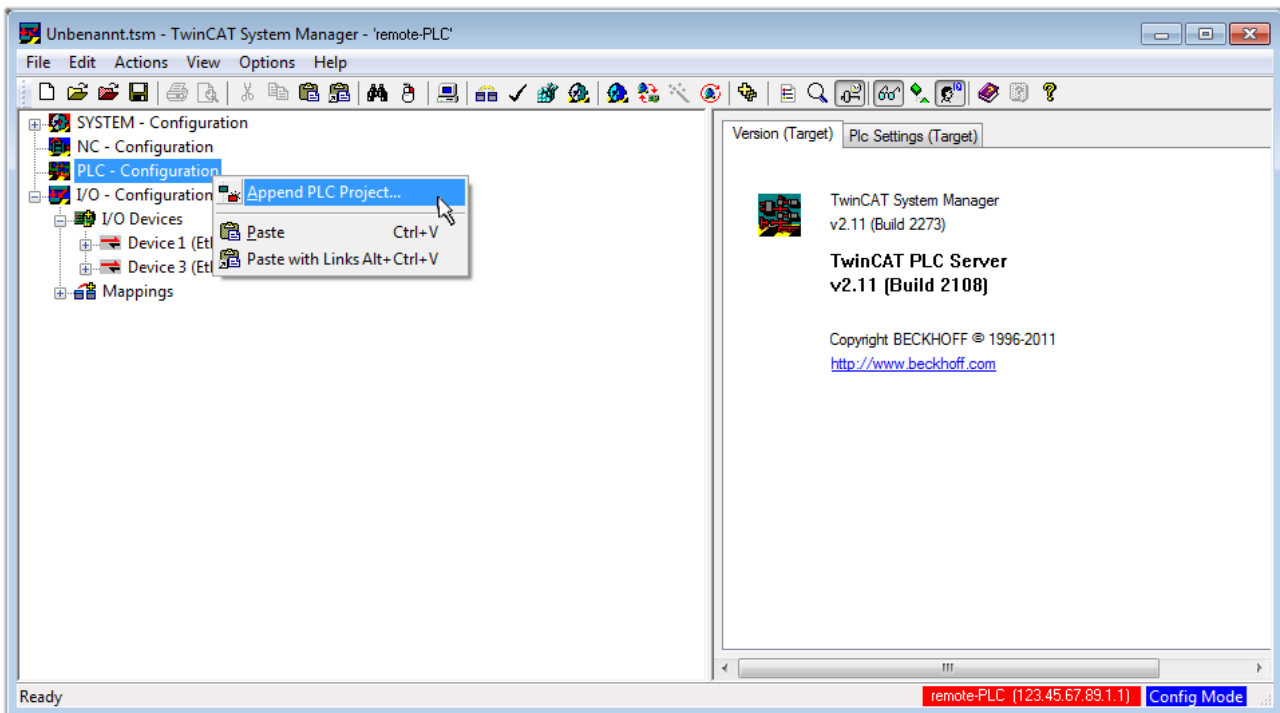


Fig. 36: Appending the TwinCAT PLC Control project

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:

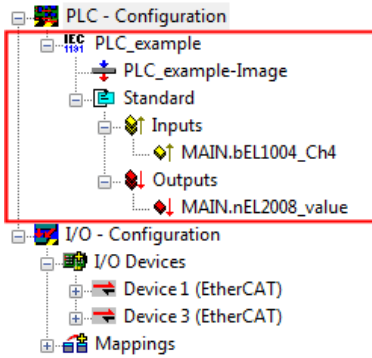


Fig. 37: PLC project integrated in the PLC configuration of the System Manager

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”:

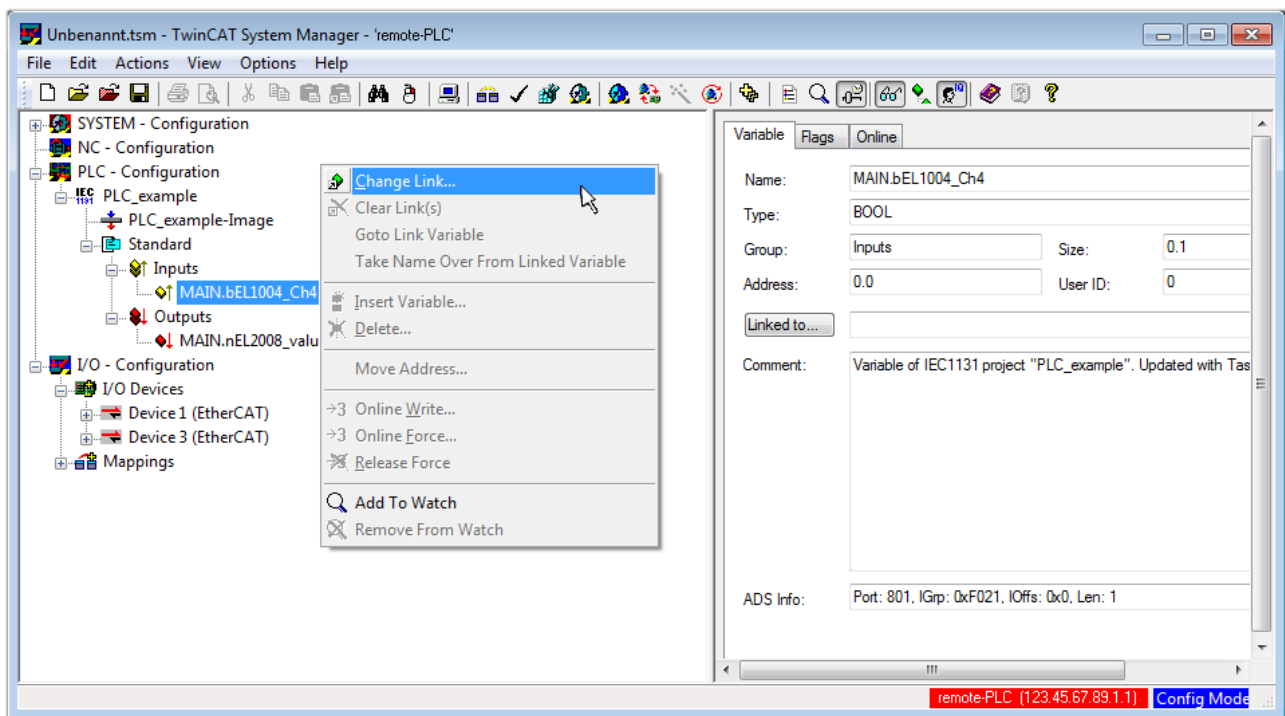


Fig. 38: Creating the links between PLC variables and process objects

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

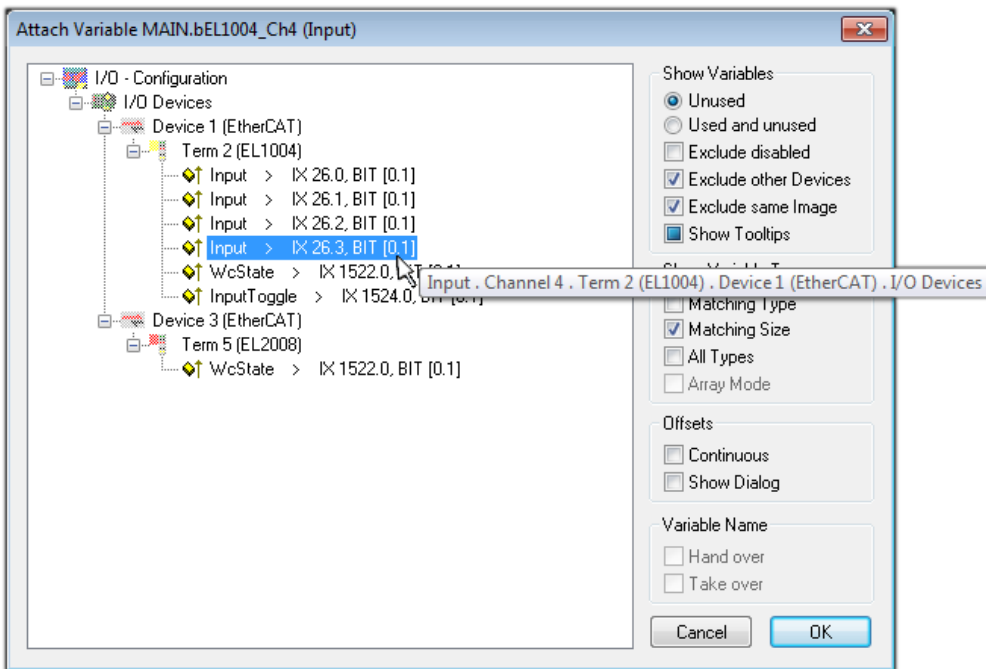


Fig. 39: Selecting PDO of type BOOL

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

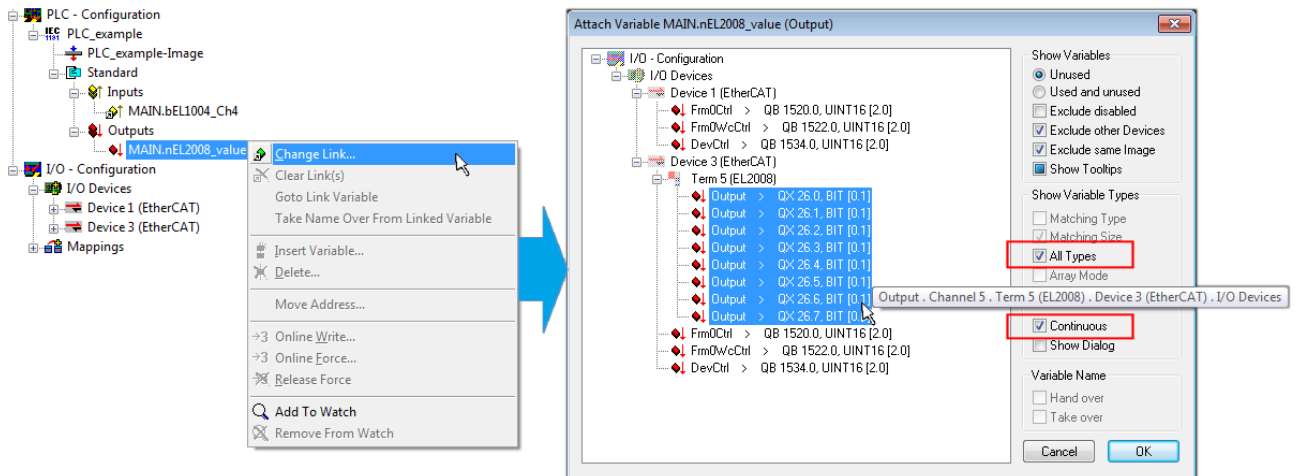



Fig. 40: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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

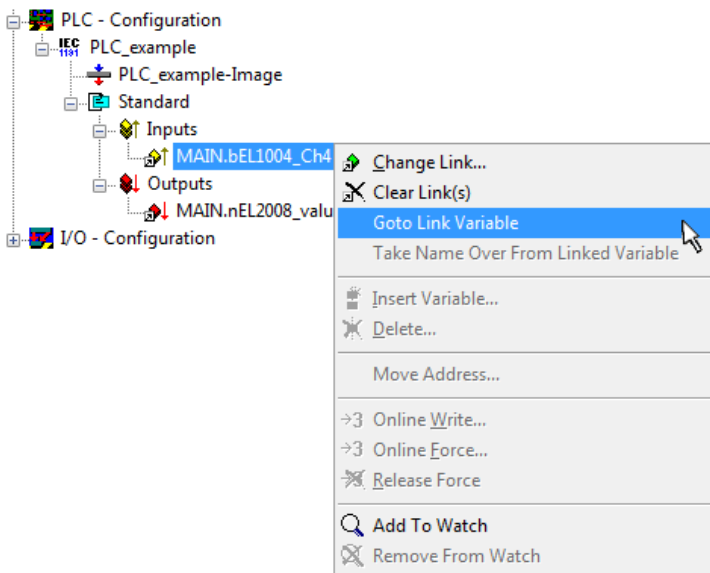

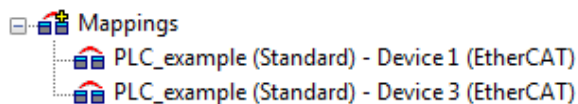


Fig. 41: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

The process of assigning variables to the PDO is completed via the menu selection “Actions” → “Generate

Mappings”, key Ctrl+M or by clicking on the symbol  in the menu.


This can be visualized in the configuration:




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

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated. First, the configuration can be verified

via  (or via “Actions” → “Check Configuration”). If no error is present, the configuration can be

activated via  (or via “Actions” → “Activate Configuration...”) to transfer the System Manager settings to the runtime system. Confirm the messages “Old configurations are overwritten!” and “Restart TwinCAT system in Run mode” with “OK”.

A few seconds later the real-time status **RTime 0%** is displayed at the bottom right in the System Manager. The PLC system can then be started as described below.

Starting the controller

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

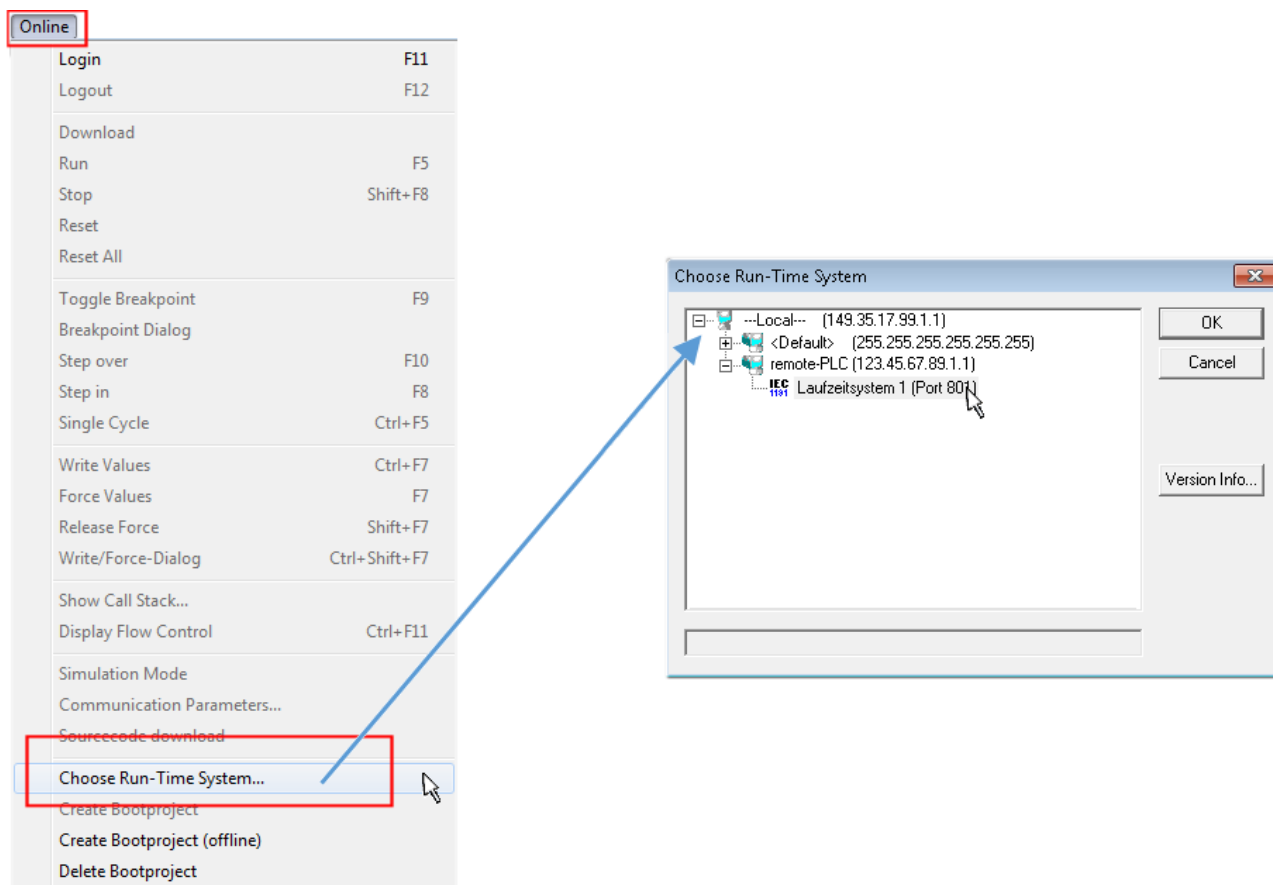



Fig. 42: Choose target system (remote)

In this sample “Runtime system 1 (port 801)” is selected and confirmed. Link the PLC with the real-time

system via menu option “Online” → “Login”, the F11 key or by clicking on the symbol . The control program can then be loaded for execution. This results in the message “No program on the controller! Should the new program be loaded?”, which should be acknowledged with “Yes”. The runtime environment is ready for the program start:

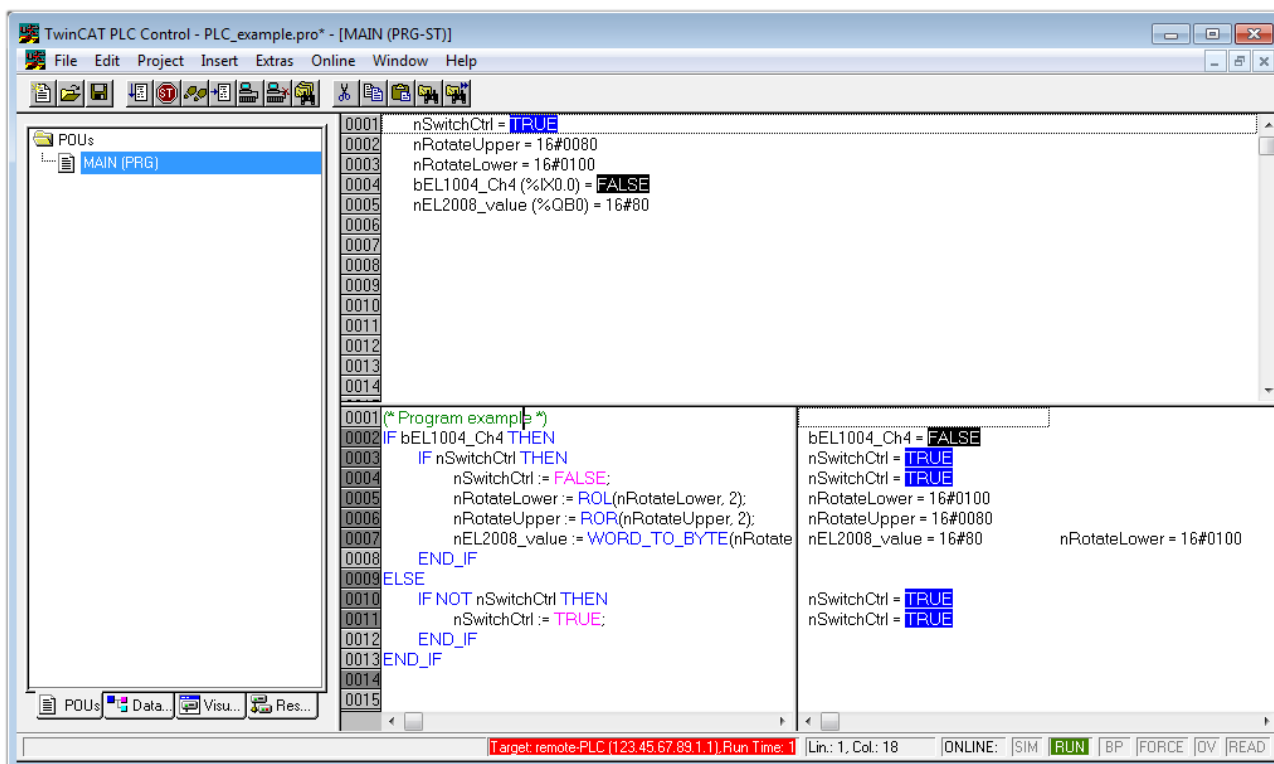


Fig. 43: PLC Control logged in, ready for program startup

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

5.1.2 TwinCAT 3


Startup

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

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



Fig. 44: Initial TwinCAT 3 user interface

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

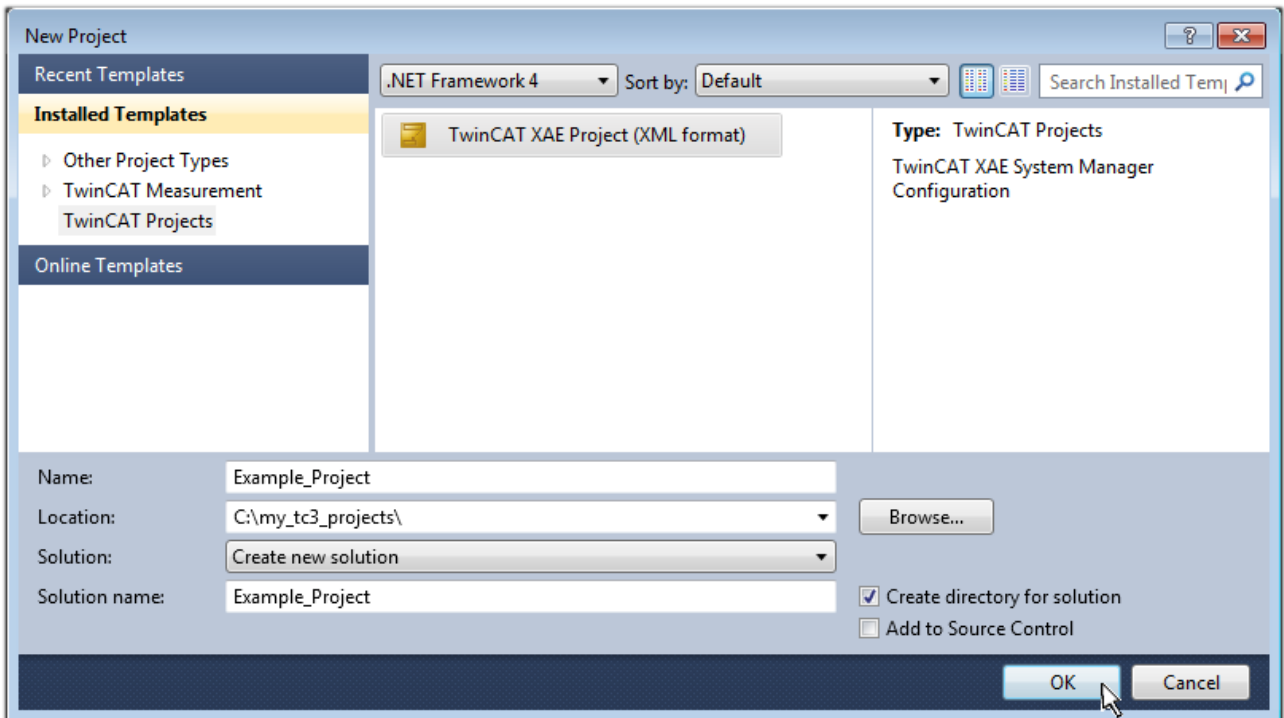


Fig. 45: Create new TwinCAT project

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

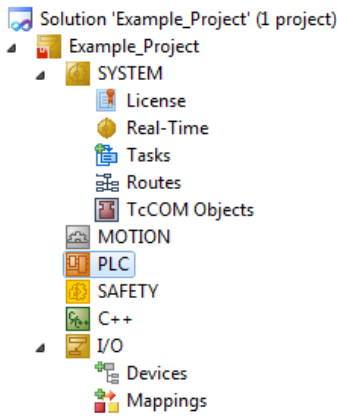
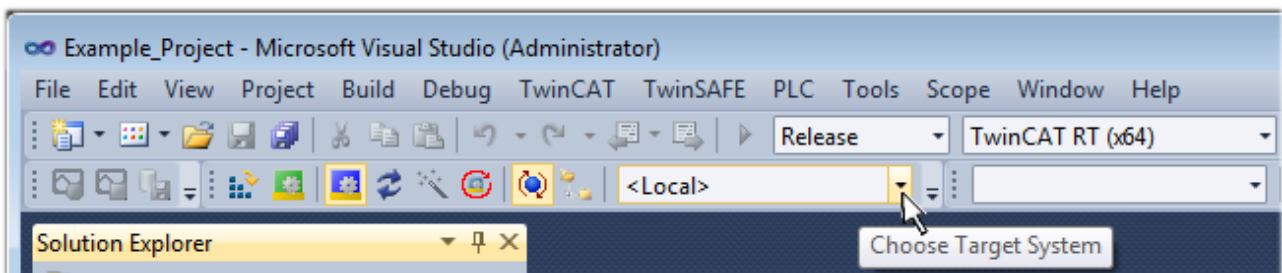


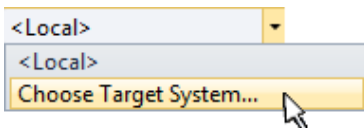
Fig. 46: New TwinCAT3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is “Insert Device [▶ 52]”.

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



expand the pull-down menu:



and open the following window:

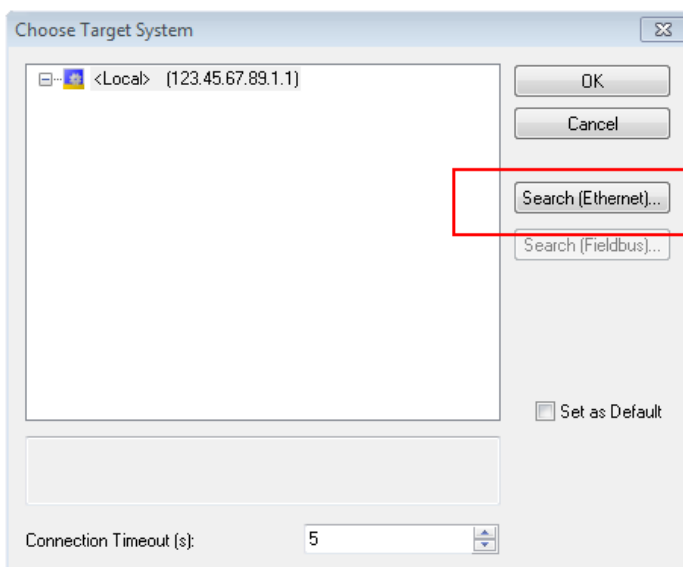


Fig. 47: Selection dialog: Choose the target system

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

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

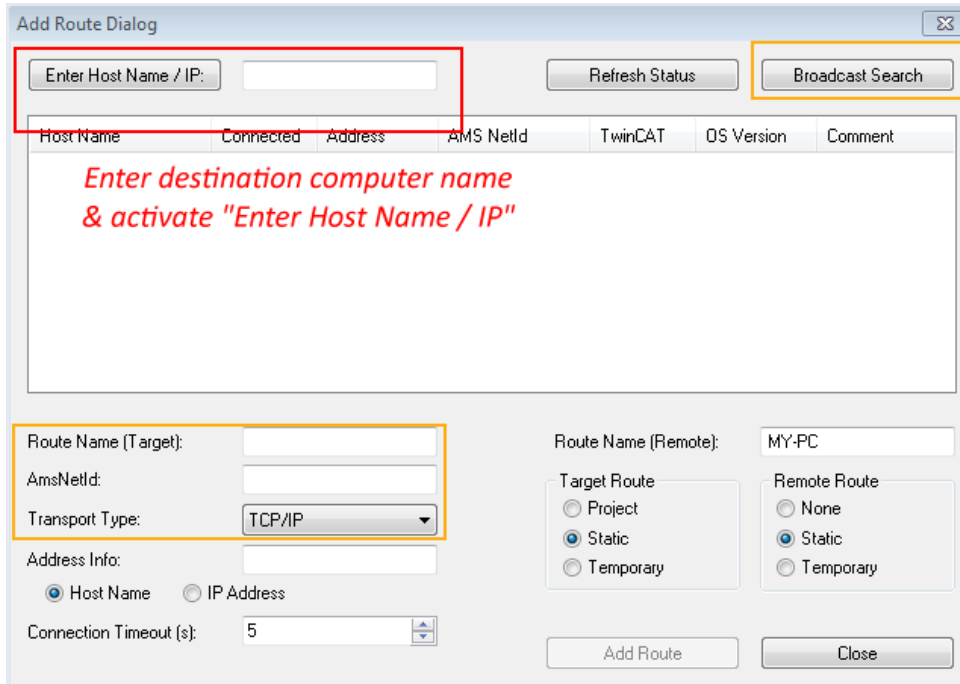
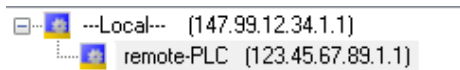


Fig. 48: Specify the PLC for access by the TwinCAT System Manager: selection of the target system


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




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

Adding devices

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

element “I/O”, then right-click to open a context menu and select “Scan” or start the action via  in the

menu bar. The TwinCAT System Manager may first have to be set to “Config mode” via  or via the menu “TwinCAT” → “Restart TwinCAT (Config mode)”.

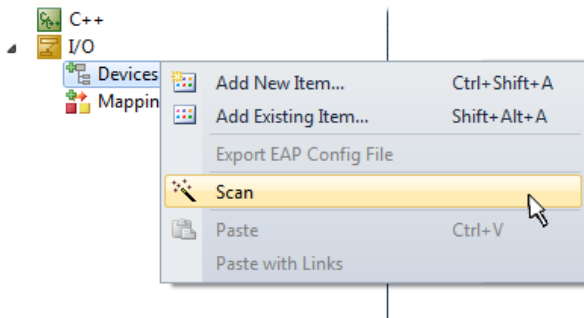


Fig. 49: Select “Scan”

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

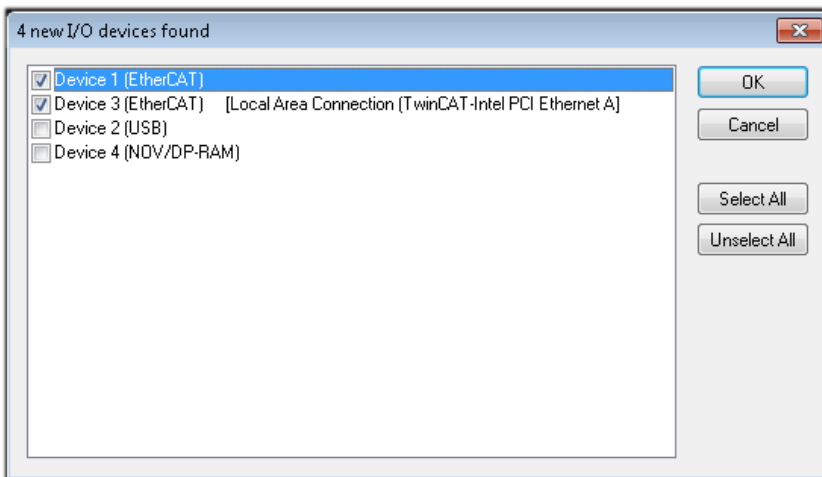


Fig. 50: Automatic detection of I/O devices: selection the devices to be integrated

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

Based on the [sample configuration \[▶ 37\]](#) described at the beginning of this section, the result is as follows:

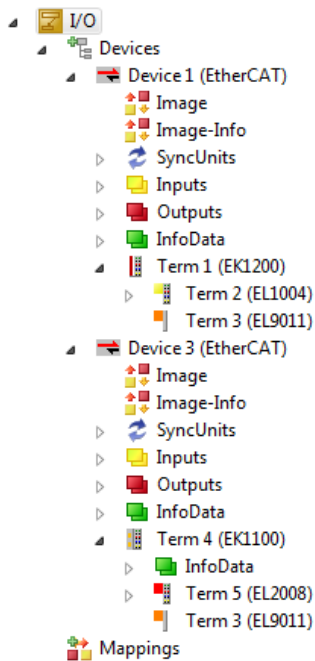


Fig. 51: Mapping of the configuration in VS shell of the TwinCAT3 environment

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

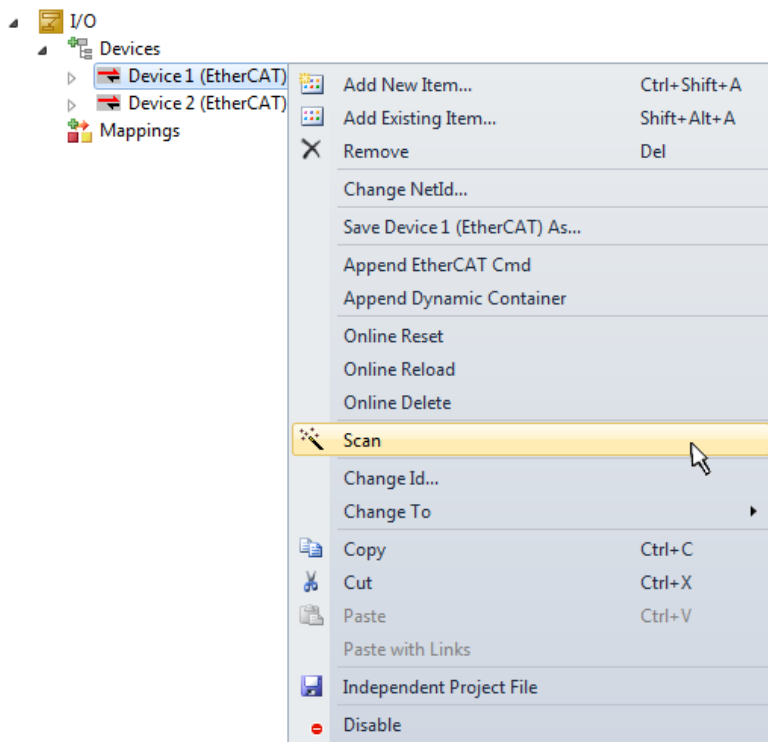


Fig. 52: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

The following section refers to Structured Text (ST).

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

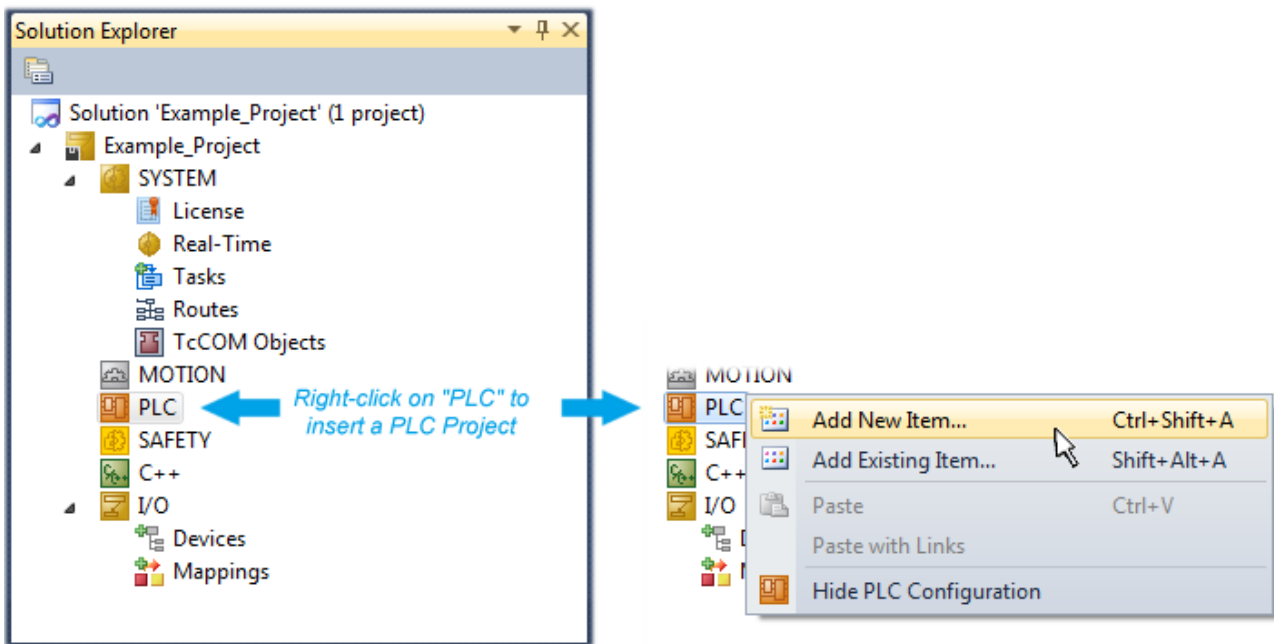


Fig. 53: Adding the programming environment in “PLC”

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

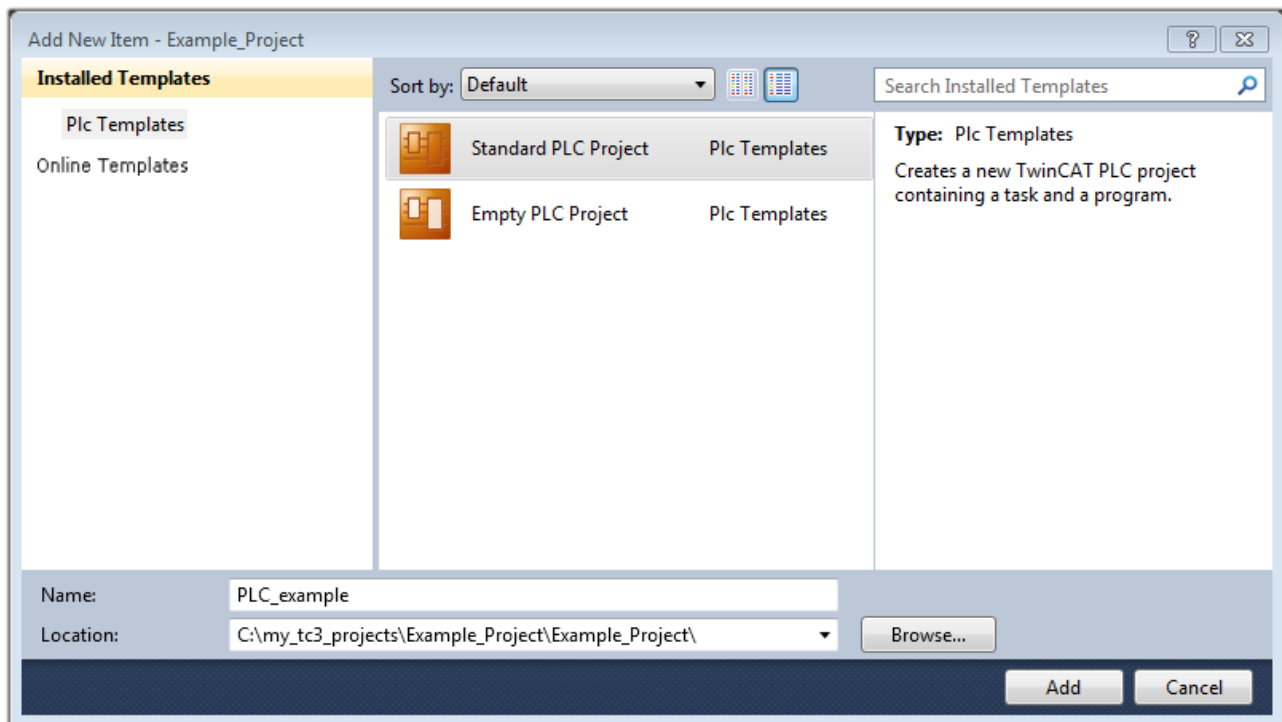


Fig. 54: Specifying the name and directory for the PLC programming environment

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

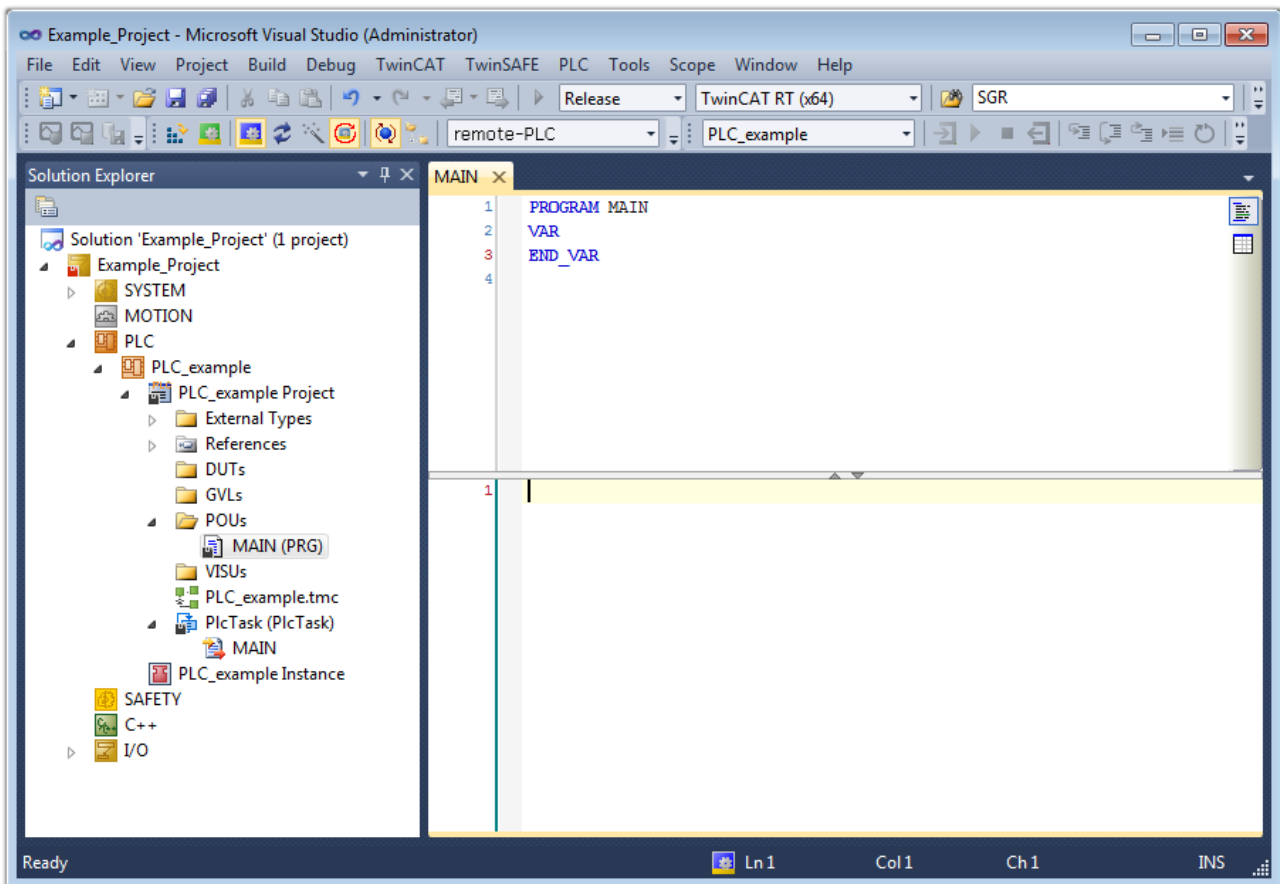


Fig. 55: Initial “Main” program of the standard PLC project

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

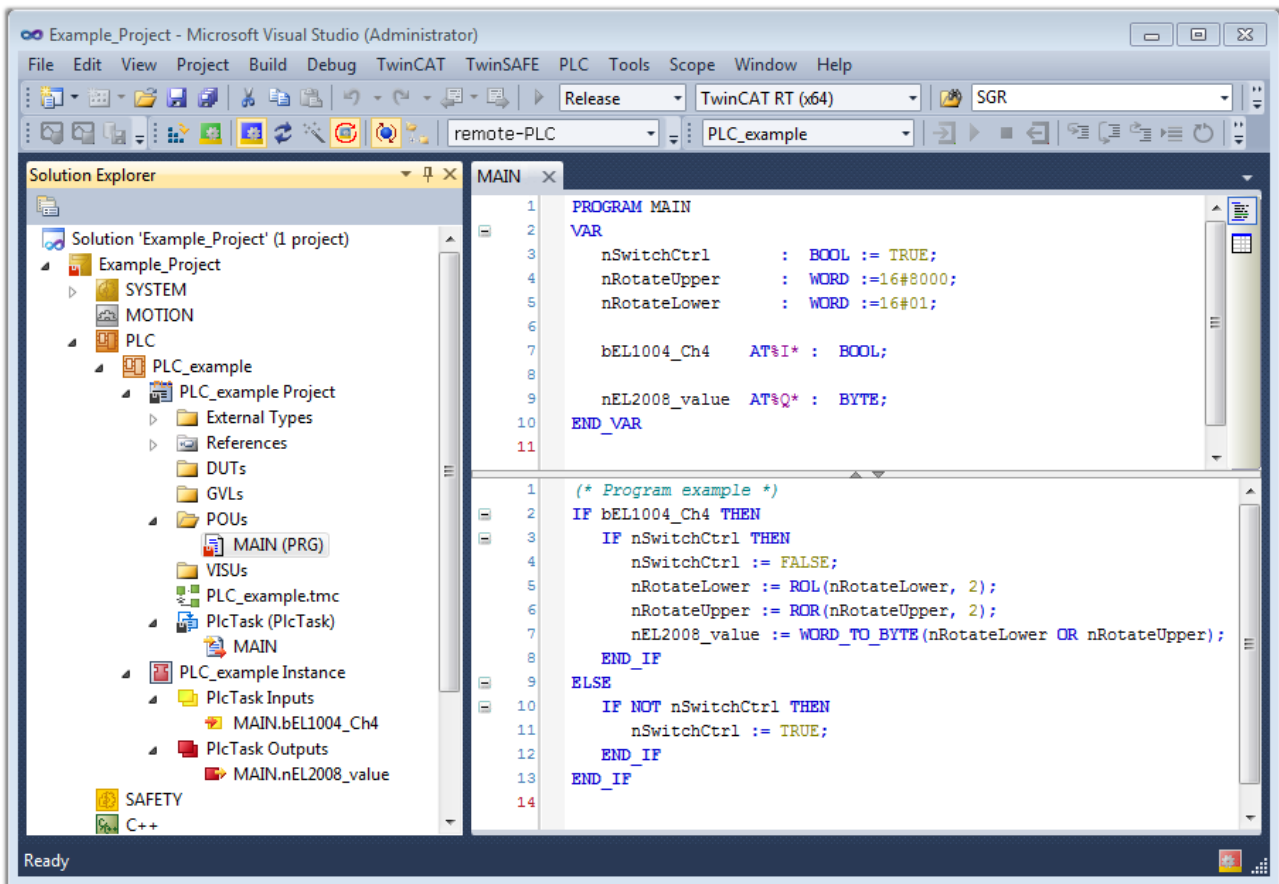


Fig. 56: Sample program with variables after a compile process (without variable integration)

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

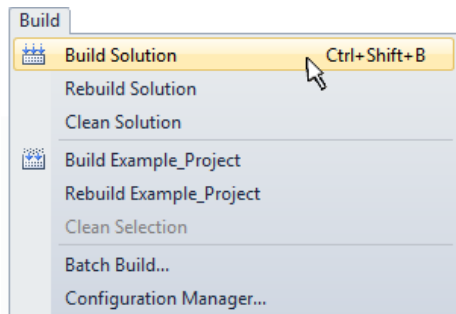
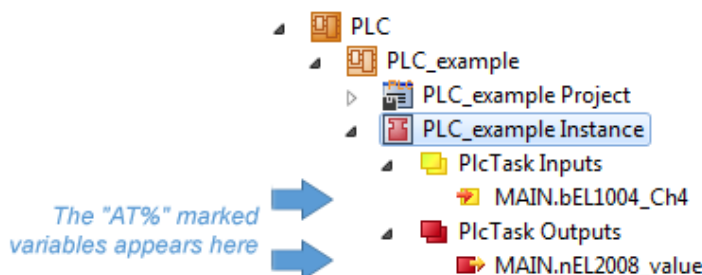


Fig. 57: Start program compilation

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



Assigning variables

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

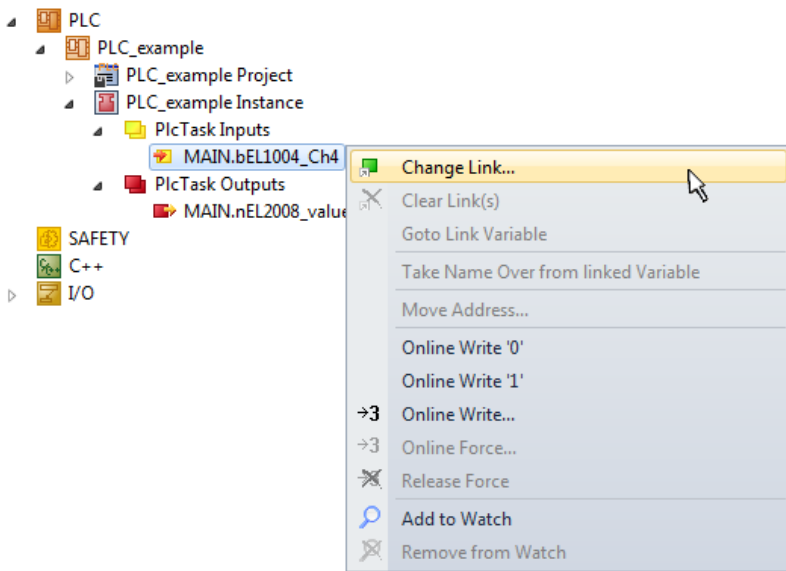


Fig. 58: Creating the links between PLC variables and process objects

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

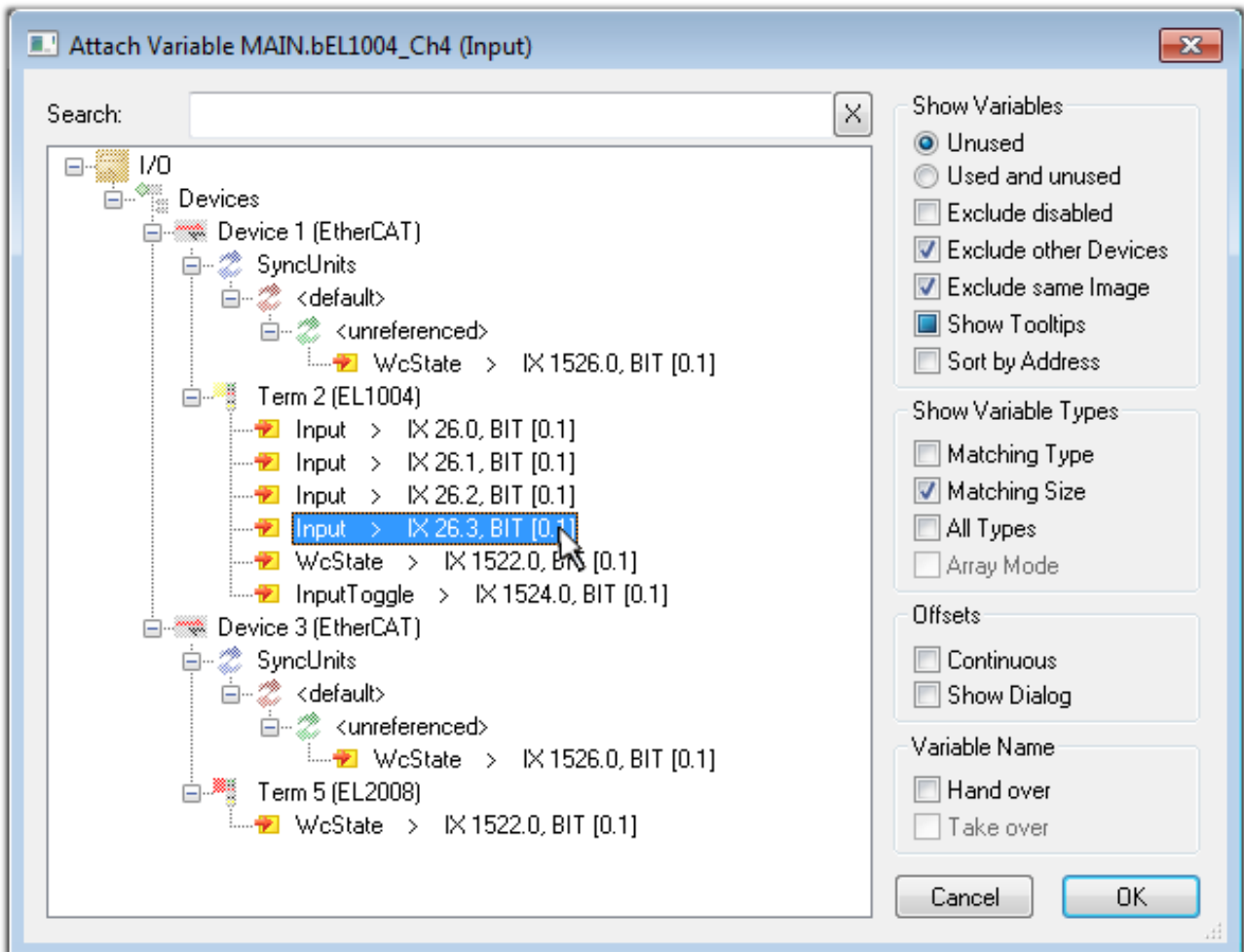


Fig. 59: Selecting PDO of type BOOL

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

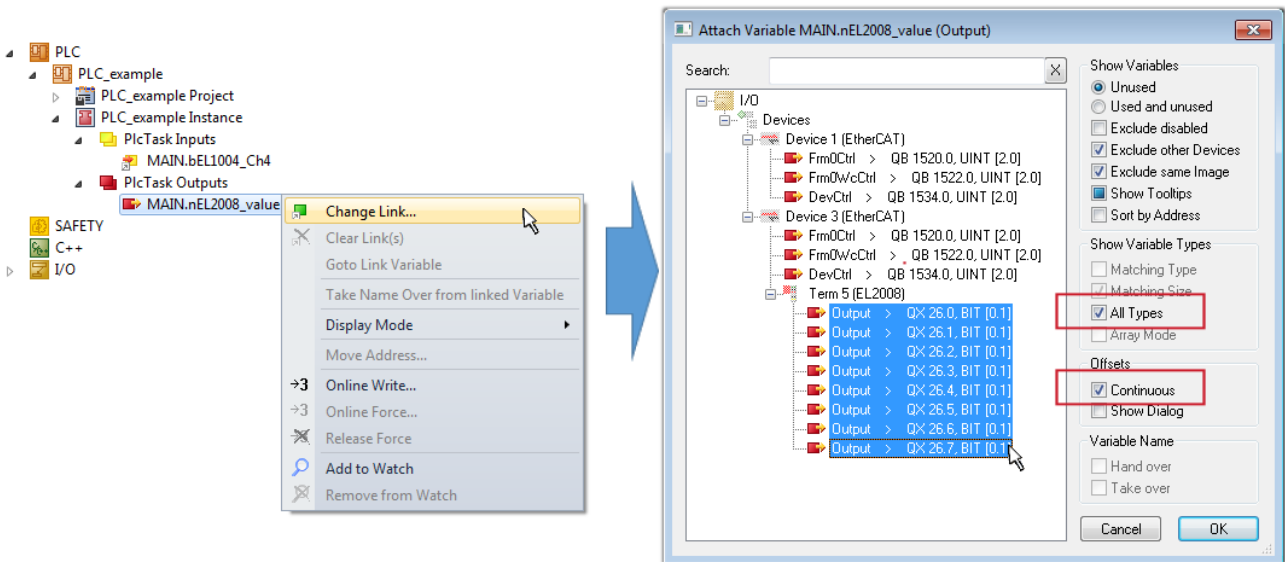



Fig. 60: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

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

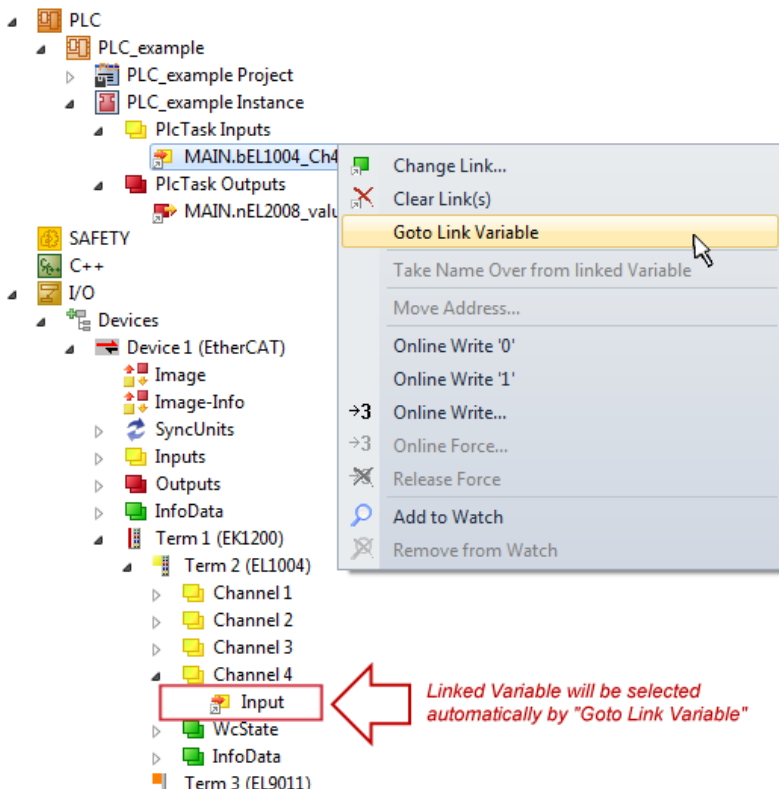


Fig. 61: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

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

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

● Note on the type of variable assignment



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

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

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

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

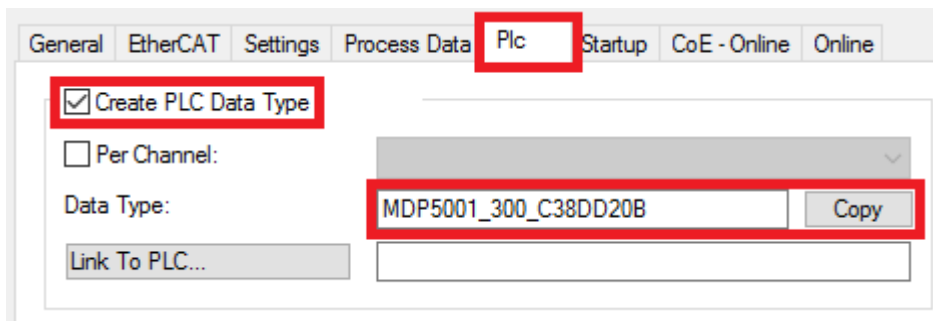


Fig. 62: Creating a PLC data type

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

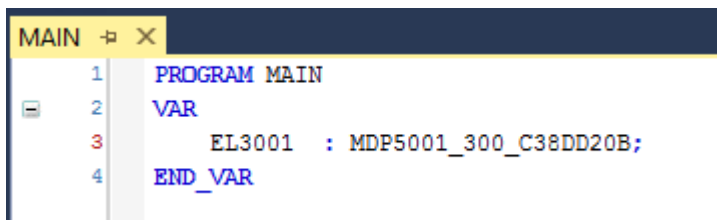


Fig. 63: Instance_of_struct

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.

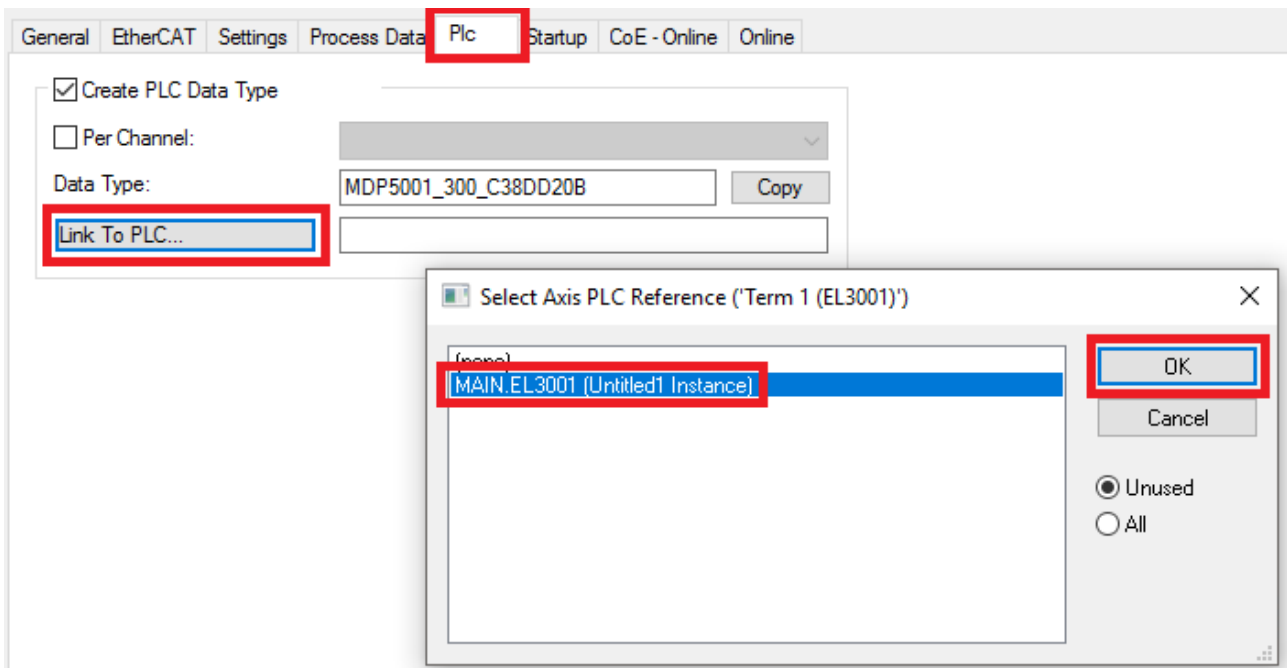


Fig. 64: Linking the structure

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

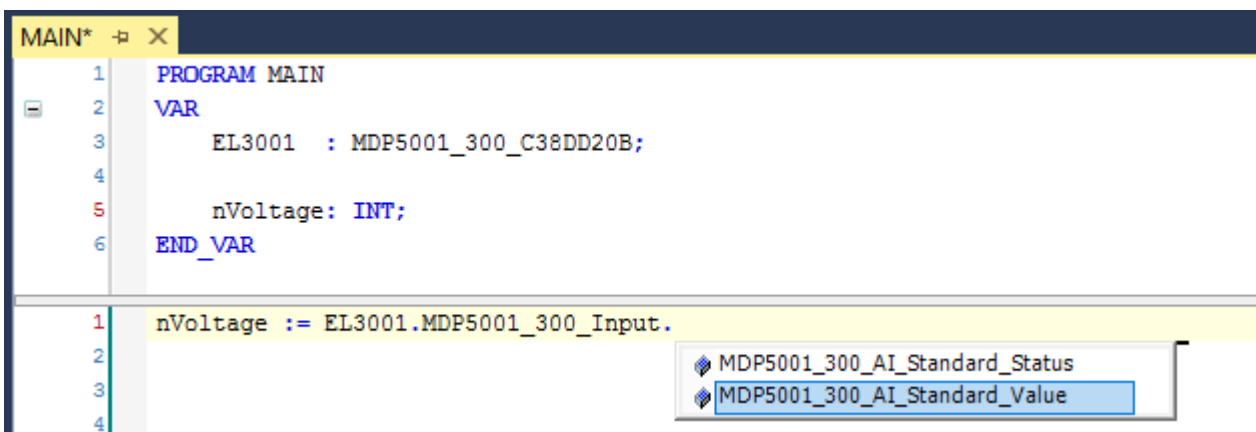
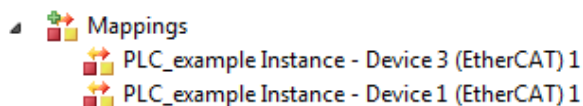


Fig. 65: Reading a variable from the structure of the process data

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs

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





A few seconds later the corresponding status of the Run mode is displayed in the form of a rotating symbol



at the bottom right of the VS shell development environment. The PLC system can then be started as described below.

Starting the controller

Select the menu option “PLC” → “Login” or click on  to link the PLC with the real-time system and load the control program for execution. This results in the message *No program on the controller! Should the new program be loaded?*, which should be acknowledged with “Yes”. The runtime environment is ready for

program start by click on symbol , the “F5” key or via “PLC” in the menu selecting “Start”. The started programming environment shows the runtime values of individual variables:

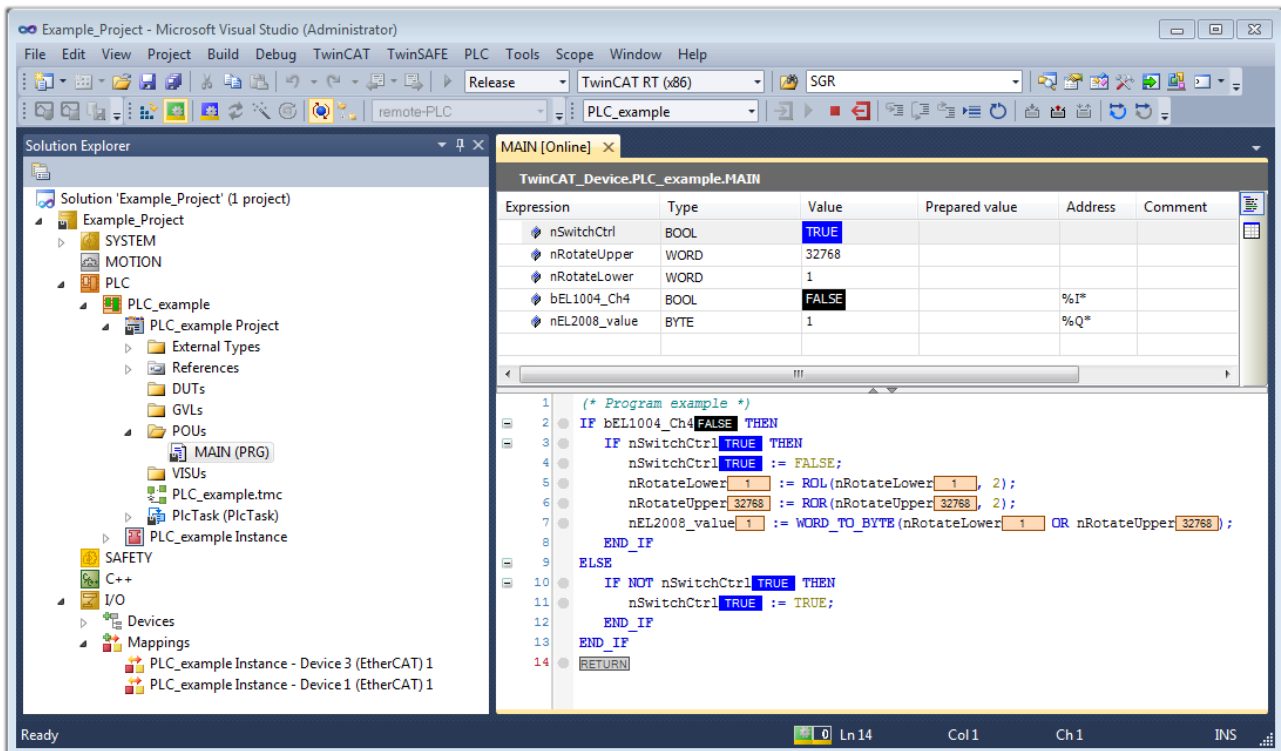


Fig. 66: TwinCAT development environment (VS shell): logged-in, after program startup

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

5.2 TwinCAT Development Environment

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

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

Details:

- **TwinCAT 2:**
 - Connects I/O devices to tasks in a variable-oriented manner
 - Connects tasks to tasks in a variable-oriented manner
 - Supports units at the bit level
 - Supports synchronous or asynchronous relationships
 - Exchange of consistent data areas and process images
 - Datalink on NT - Programs by open Microsoft Standards (OLE, OCX, ActiveX, DCOM+, etc.)

- Integration of IEC 61131-3-Software-SPS, Software- NC and Software-CNC within Windows NT/2000/XP/Vista, Windows 7, NT/XP Embedded, CE
- Interconnection to all common fieldbusses
- More...

Additional features:

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

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

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

5.2.1 Installation of the TwinCAT real-time driver

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

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

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

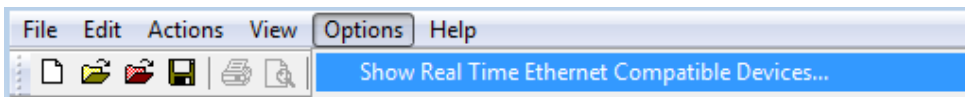


Fig. 67: System Manager “Options” (TwinCAT 2)

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

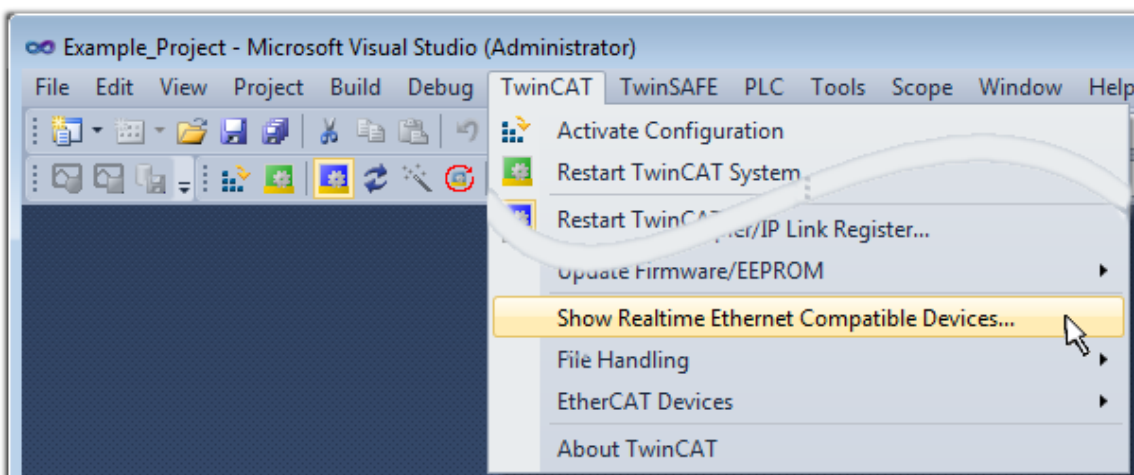


Fig. 68: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

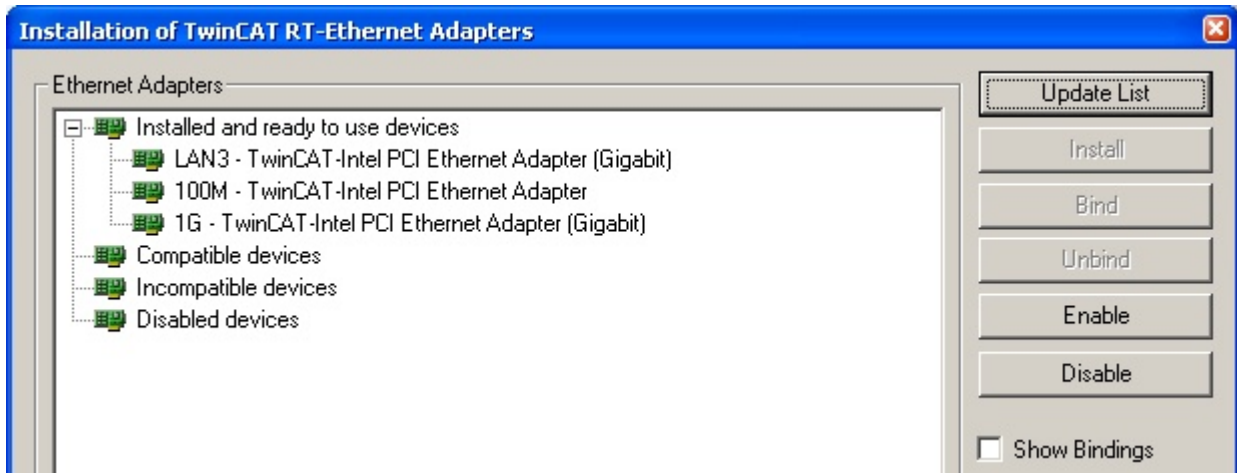


Fig. 69: Overview of network interfaces

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” \[► 73\]](#) in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices...”):



Fig. 70: EtherCAT device properties(TwinCAT 2): click on “Compatible Devices...” of tab “Adapte””

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



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

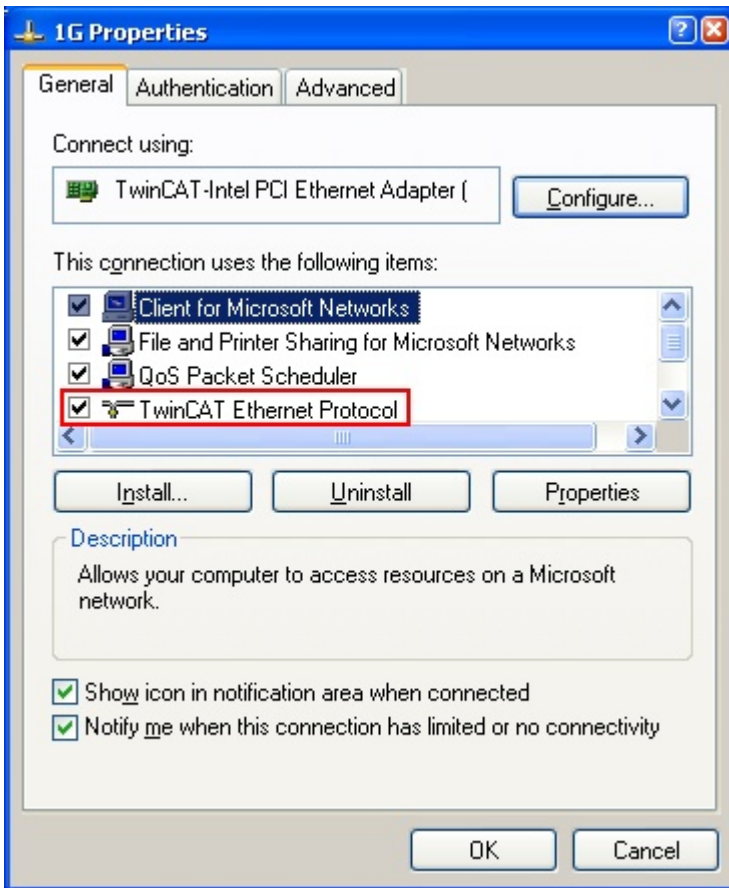


Fig. 71: Windows properties of the network interface

A correct setting of the driver could be:

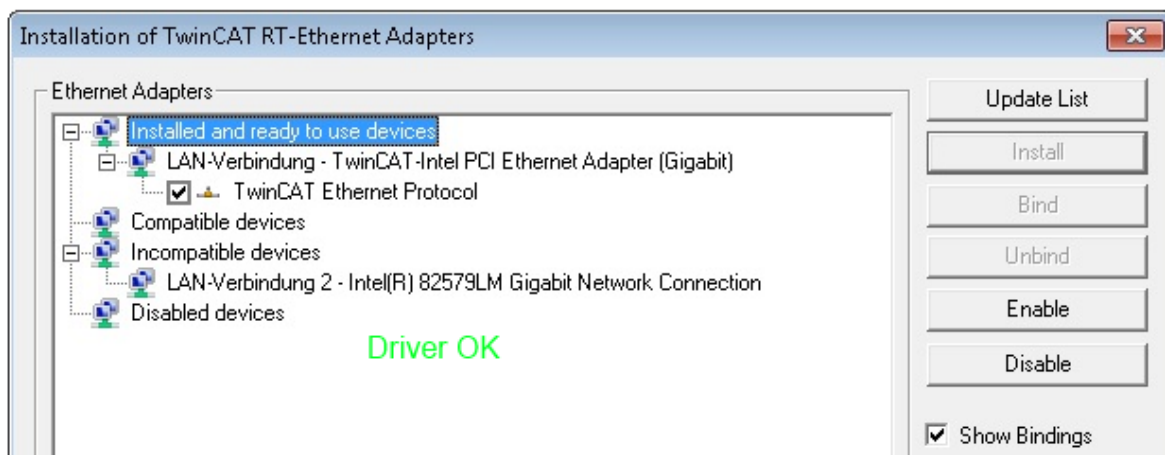


Fig. 72: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

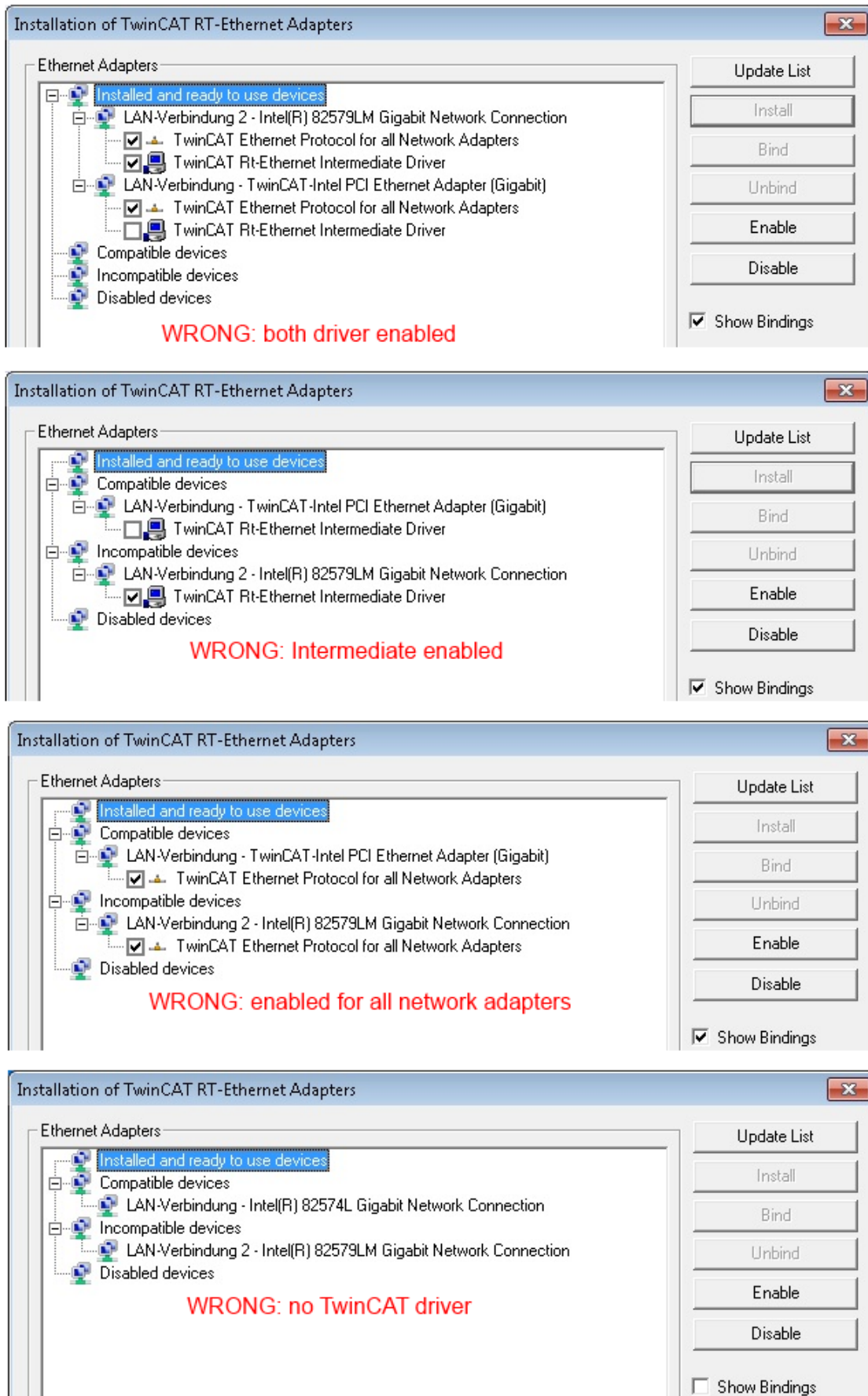


Fig. 73: Incorrect driver settings for the Ethernet port

IP address of the port used

i 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.

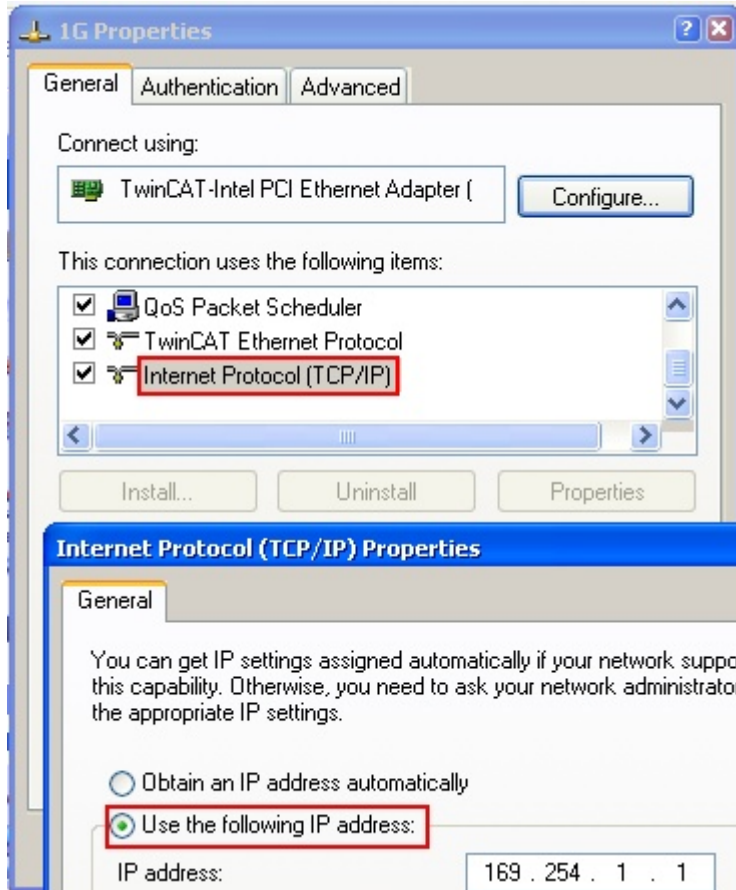


Fig. 74: TCP/IP setting for the Ethernet port

5.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

The TwinCAT EtherCAT master/System Manager needs the device description files for the devices to be used in order to 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 \[► 72\]](#) 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”

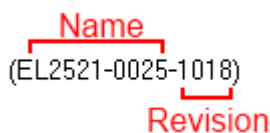


Fig. 75: Identifier structure

The order identifier consisting of name + type (here: EL2521-0010) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes.

Online description

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

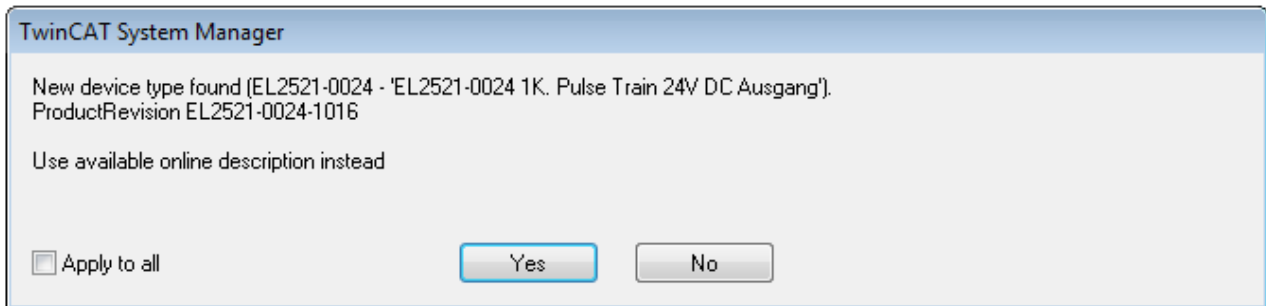


Fig. 76: OnlineDescription information window (TwinCAT 2)

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

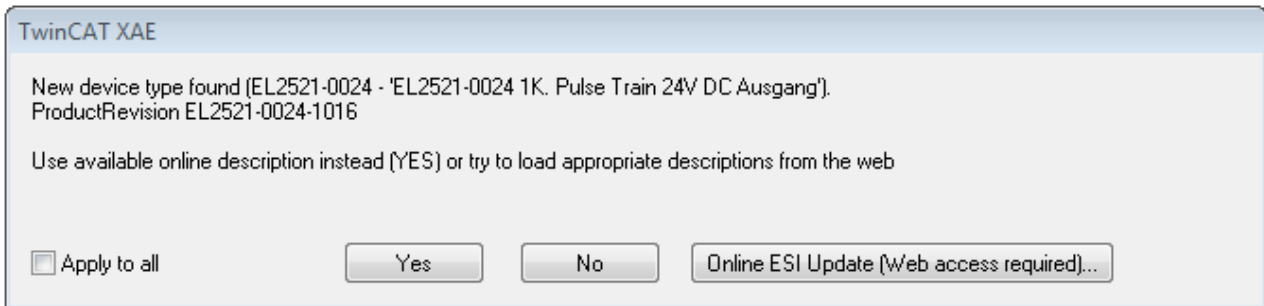


Fig. 77: Information window OnlineDescription (TwinCAT 3)

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

NOTE

Changing the “usual” configuration through a scan

- ✓ If a scan discovers a device that is not yet known to TwinCAT, distinction has to be made between two cases. Taking the example here of the EL2521-0000 in the revision 1019
 - a) no ESI is present for the EL2521-0000 device at all, either for the revision 1019 or for an older revision. The ESI must then be requested from the manufacturer (in this case Beckhoff).
 - b) an ESI is present for the EL2521-0000 device, but only in an older revision, e.g. 1018 or 1017. In this case an in-house check should first be performed to determine whether the spare parts stock allows the integration of the increased revision into the configuration at all. A new/higher revision usually also brings along new features. If these are not to be used, work can continue without reservations with the previous revision 1018 in the configuration. This is also stated by the Beckhoff compatibility rule.

Refer in particular to the chapter “General notes on the use of Beckhoff EtherCAT IO components” and for manual configuration to the chapter “Offline configuration creation [▶ 73]”.

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. 78: File OnlineDescription.xml created by the System Manager

If 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*).

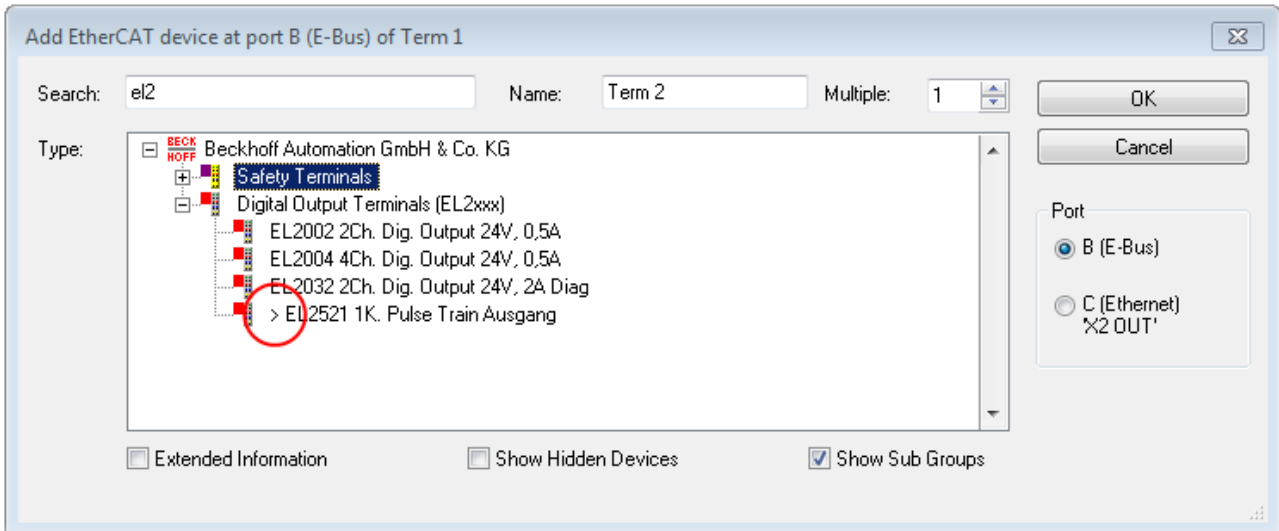


Fig. 79: Indication of an online recorded ESI of EL2521 as an example

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

i 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.

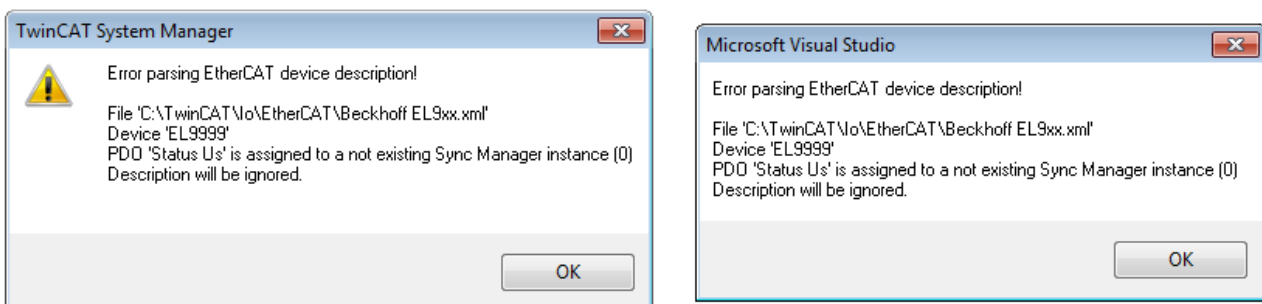


Fig. 80: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

- Structure of the *.xml does not correspond to the associated *.xsd file → check your schematics
- Contents cannot be translated into a device description → contact the file manufacturer

5.2.3 TwinCAT ESI Updater

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

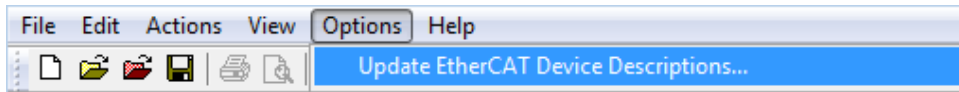


Fig. 81: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

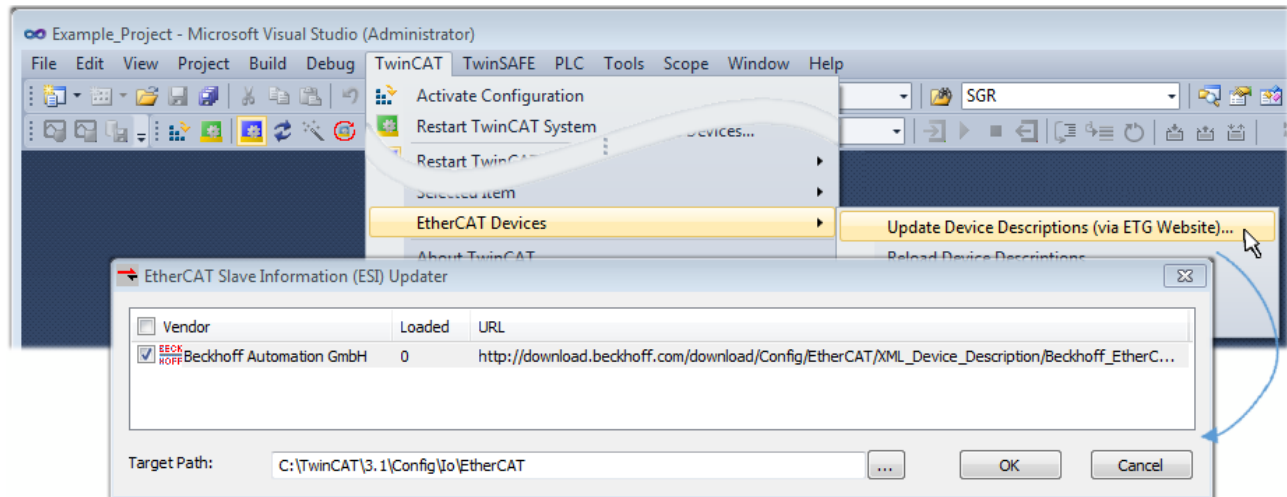


Fig. 82: Using the ESI Updater (TwinCAT 3)

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

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

5.2.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to [note “Installation of the latest ESI-XML device description” \[▶ 68\]](#).

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 [▶ 78] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 79]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 82]

The scan with existing configuration [▶ 83] can also be carried out for comparison.

5.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

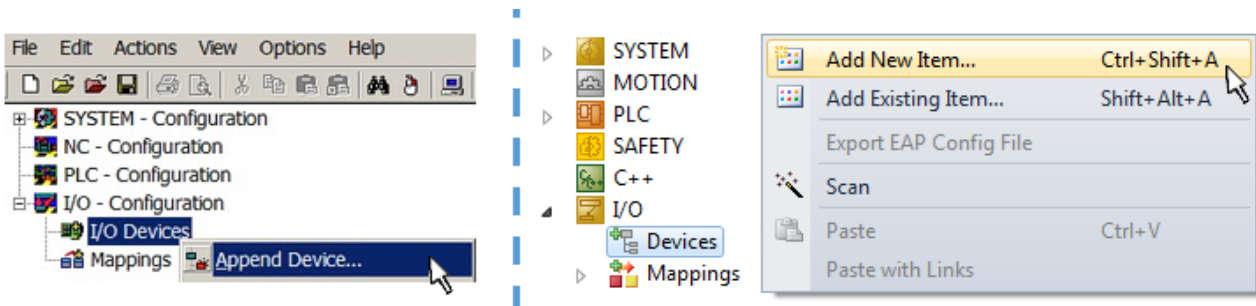


Fig. 83: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

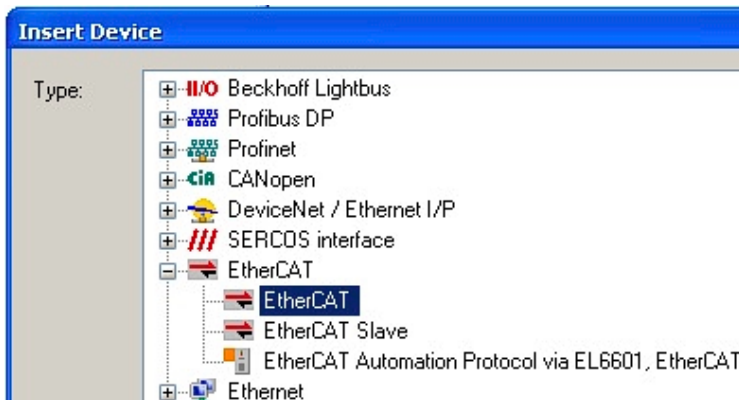


Fig. 84: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

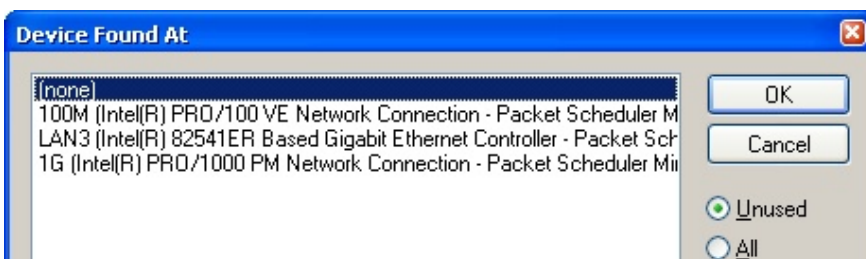


Fig. 85: Selecting the Ethernet port

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)”.

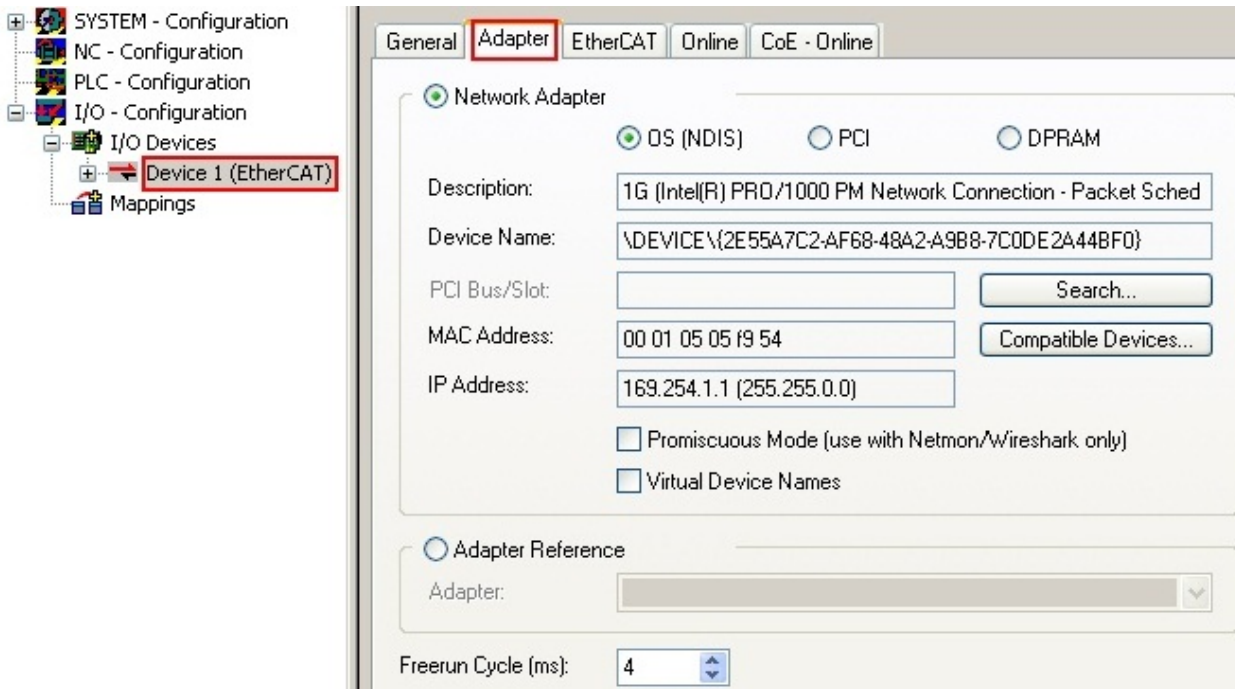
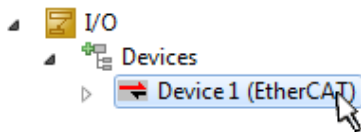


Fig. 86: EtherCAT device properties (TwinCAT 2)

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



i **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 \[▶ 63\]](#).

Defining EtherCAT slaves

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

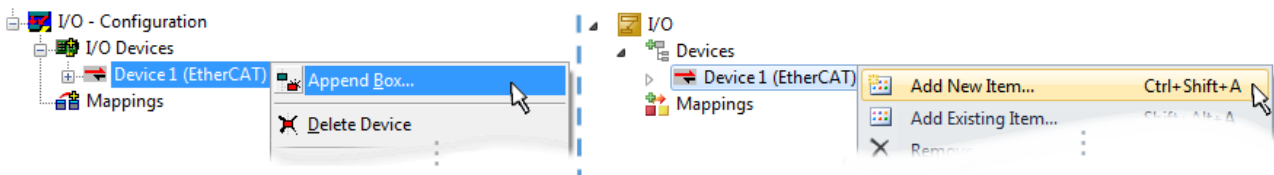


Fig. 87: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

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

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

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

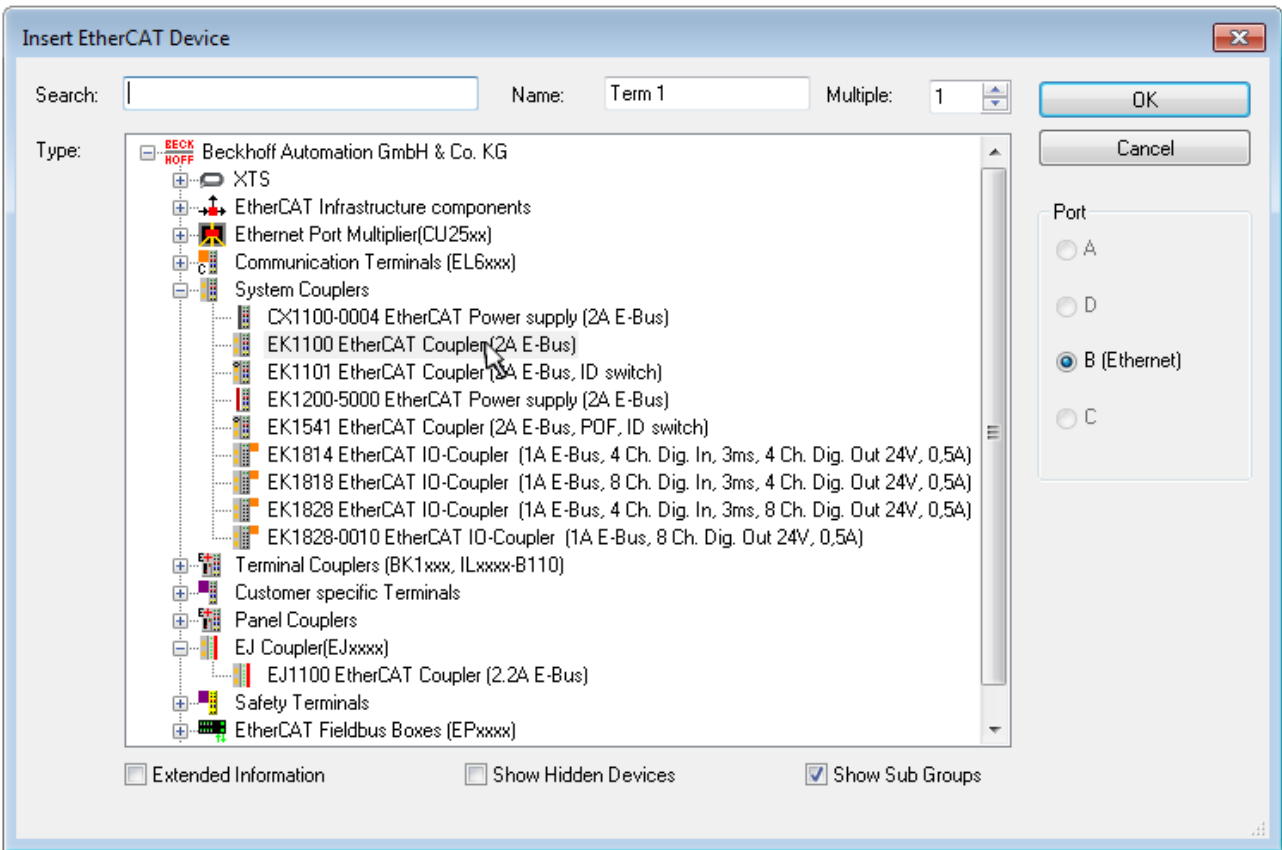


Fig. 88: Selection dialog for new EtherCAT device

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”.

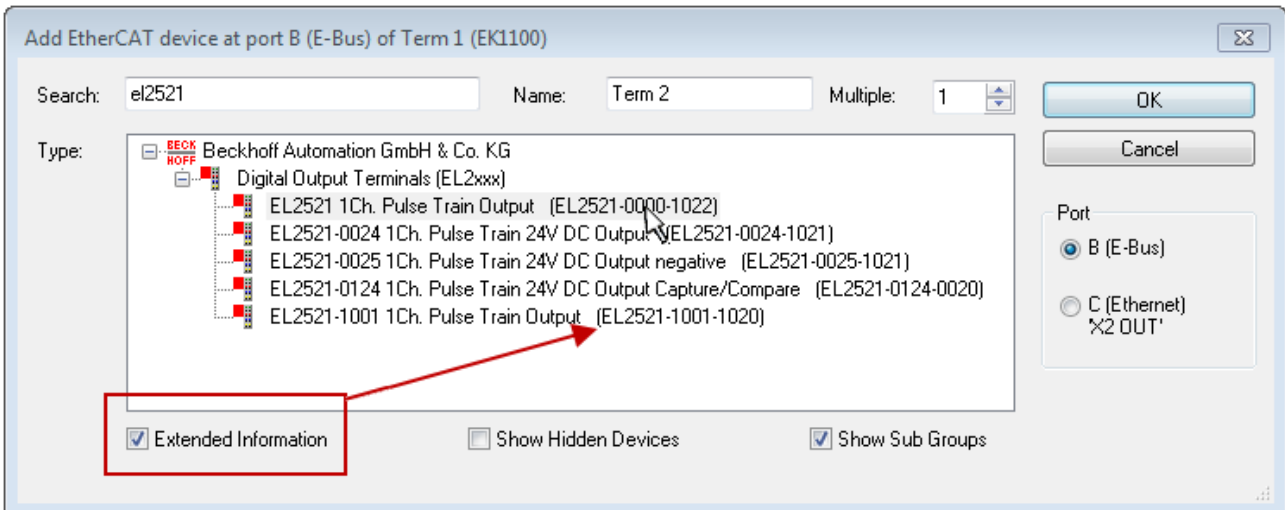


Fig. 89: Display of device revision

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”.

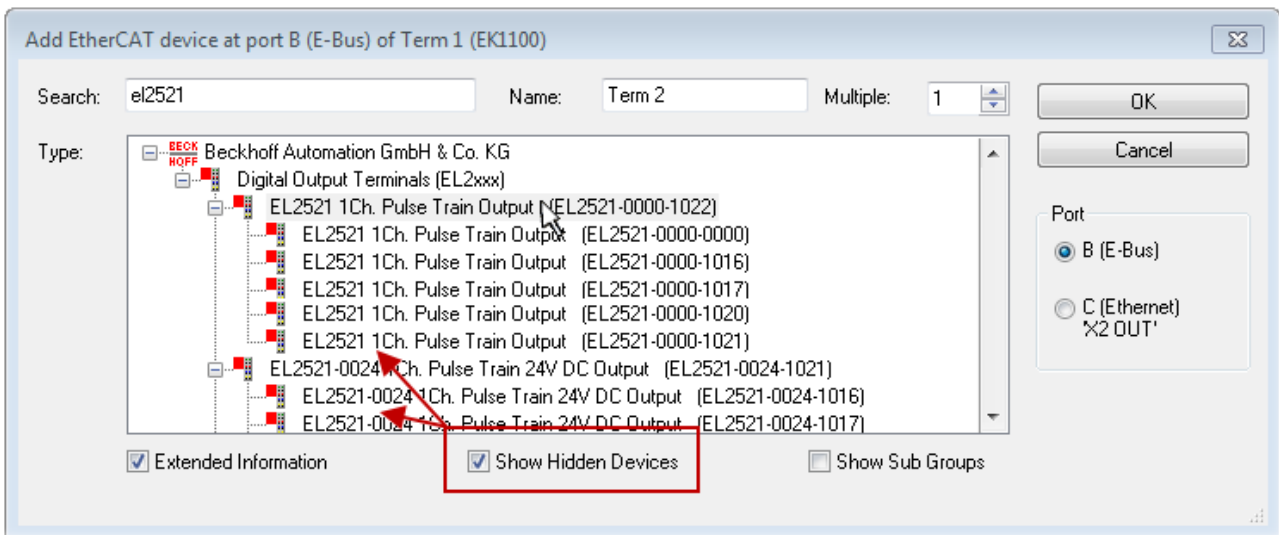


Fig. 90: Display of previous revisions

i 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.

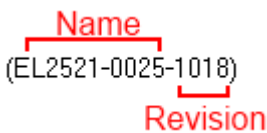


Fig. 91: Name/revision of the terminal

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, ...

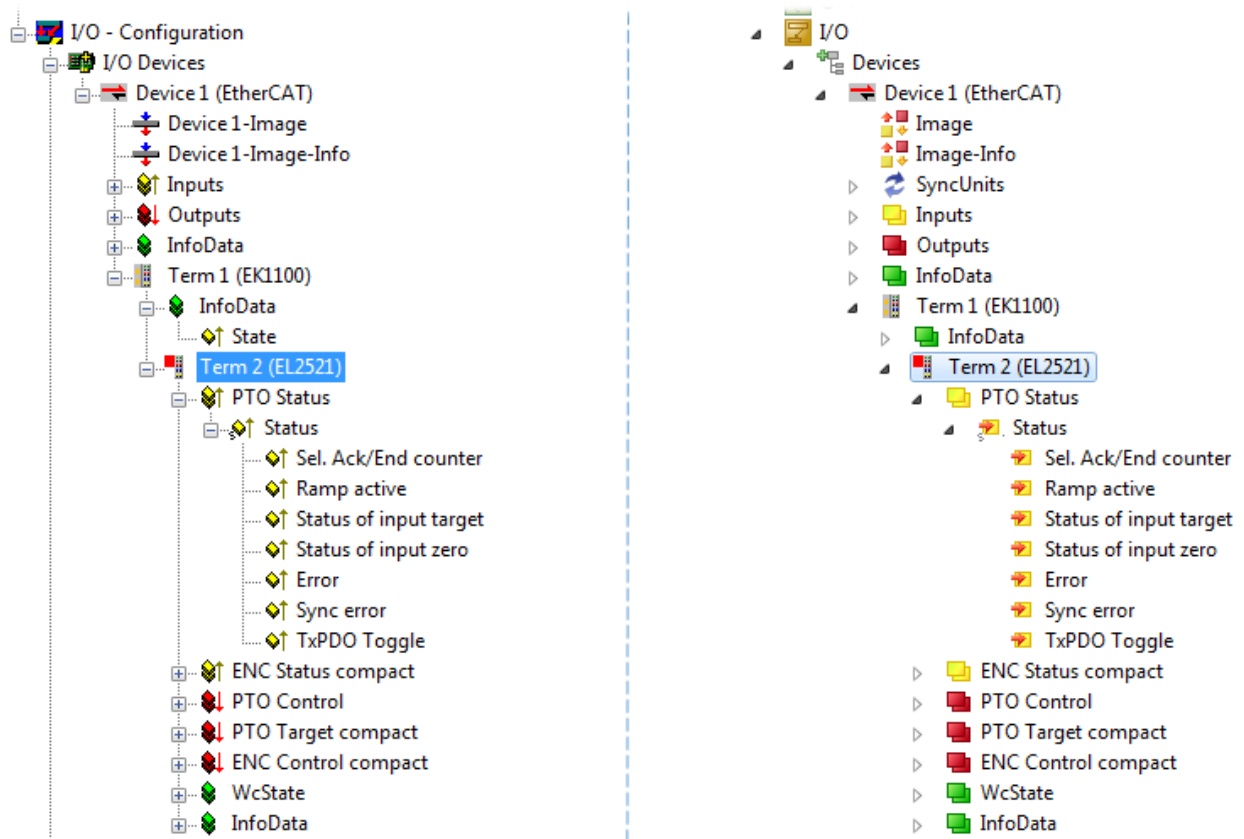




Fig. 92: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



5.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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



- on TwinCAT 2 by a blue display “Config Mode” within the System Manager window:  .
- 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

i 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.

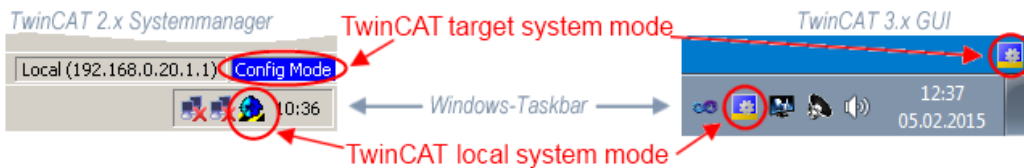


Fig. 93: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

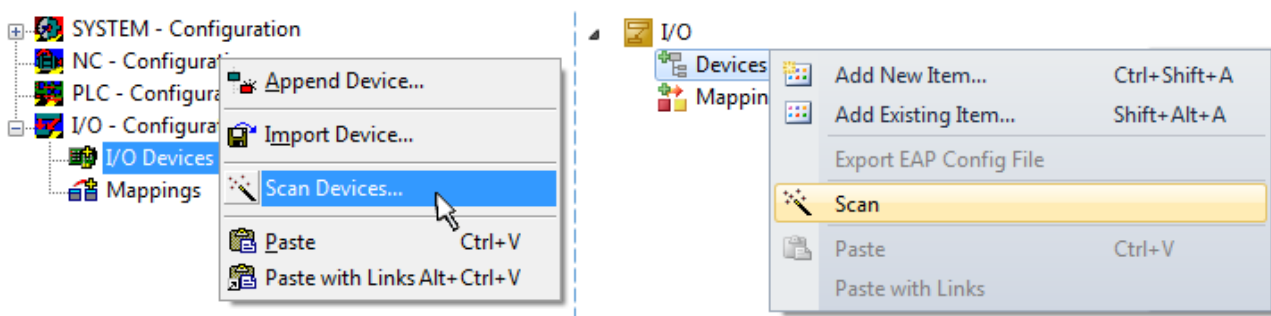


Fig. 94: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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

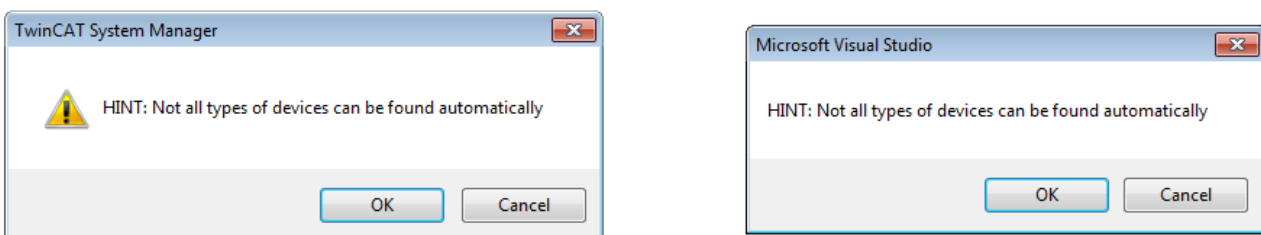


Fig. 95: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

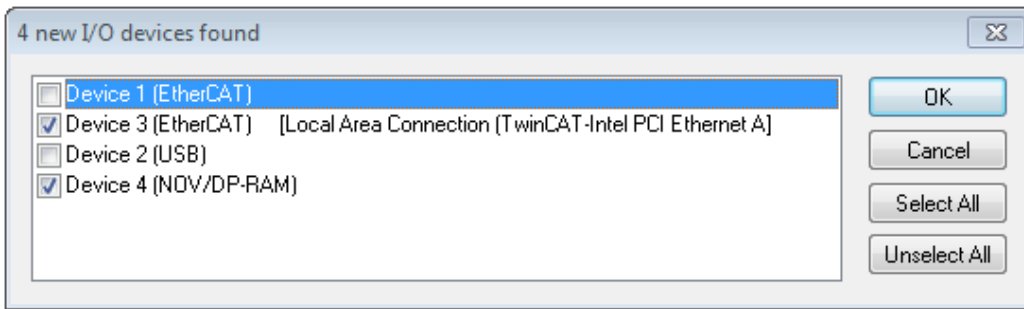


Fig. 96: Detected Ethernet devices

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](#) [▶ 63].

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.

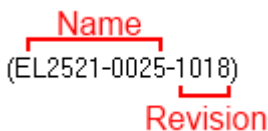


Fig. 97: Example default state

NOTE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for [comparison](#) [▶ 83] 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:

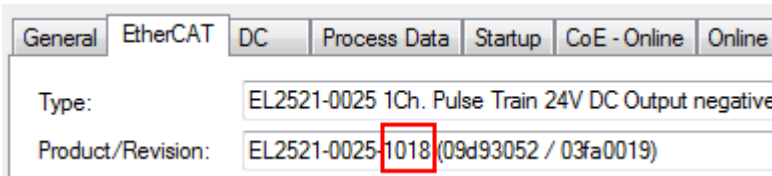


Fig. 98: Installing EthetCAT terminal with revision -1018

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 [► 83] 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:

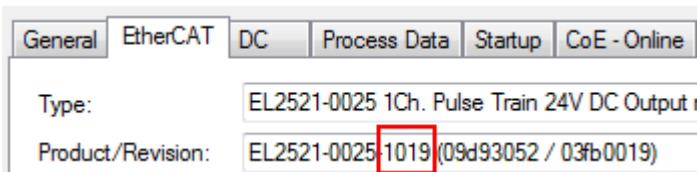


Fig. 99: Detection of EtherCAT terminal with revision -1019

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

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration “B2.tsm” created in this way. 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. 100: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

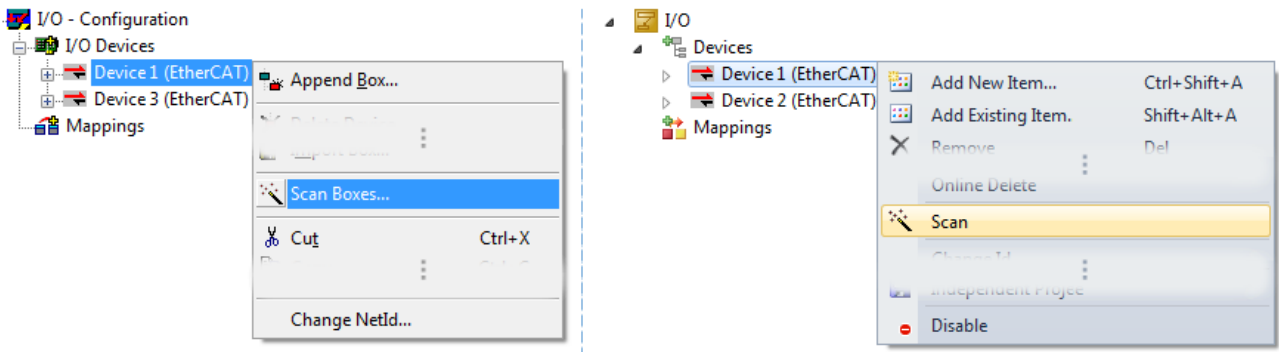


Fig. 101: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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



Fig. 102: Scan progress exemplary by TwinCAT 2

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



Fig. 103: 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. 104: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 105: 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*.

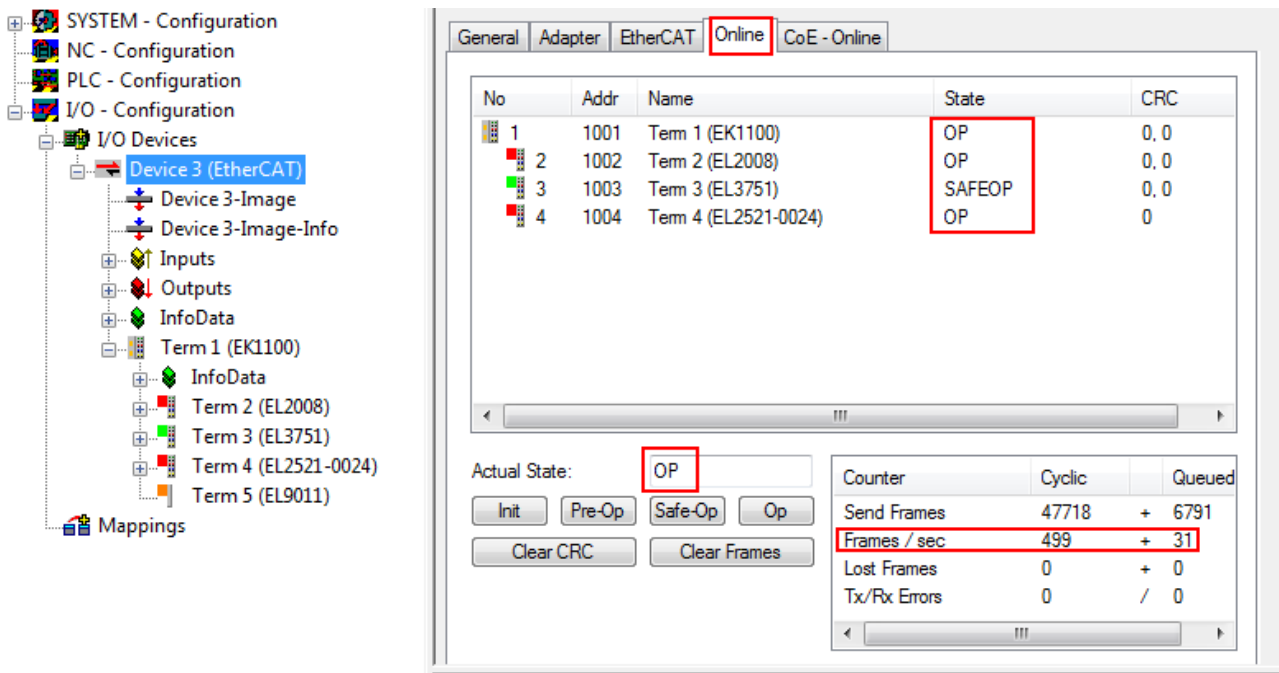


Fig. 106: Online display example

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 \[► 73\]](#).

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.

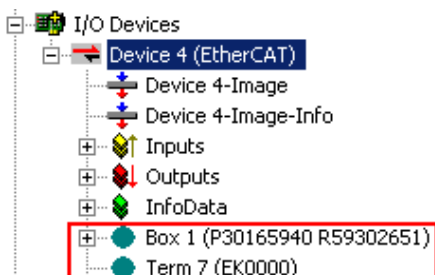


Fig. 107: Faulty identification

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

Scan over existing Configuration

NOTE

Change of the configuration after comparison

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

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



Fig. 108: 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.

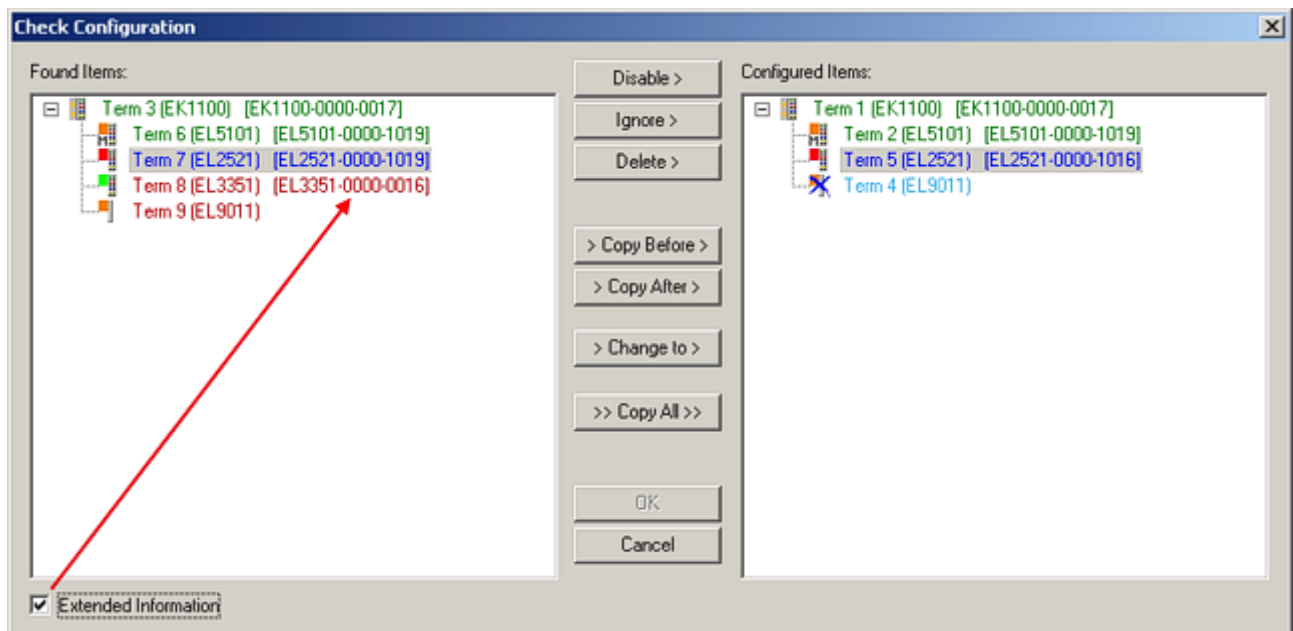


Fig. 109: Correction dialog

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

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored ("Ignore" button)
red	<ul style="list-style-type: none"> 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.

i 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 \geq 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.

Name
(EL2521-0025-1018)
Revision

Fig. 110: 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. 111: 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.

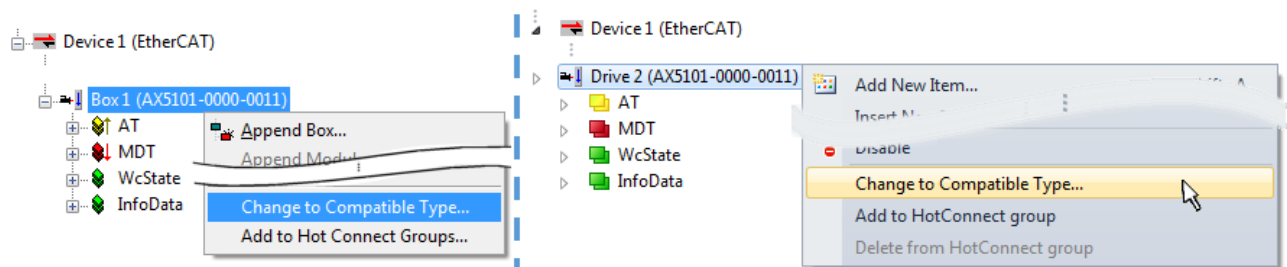


Fig. 112: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

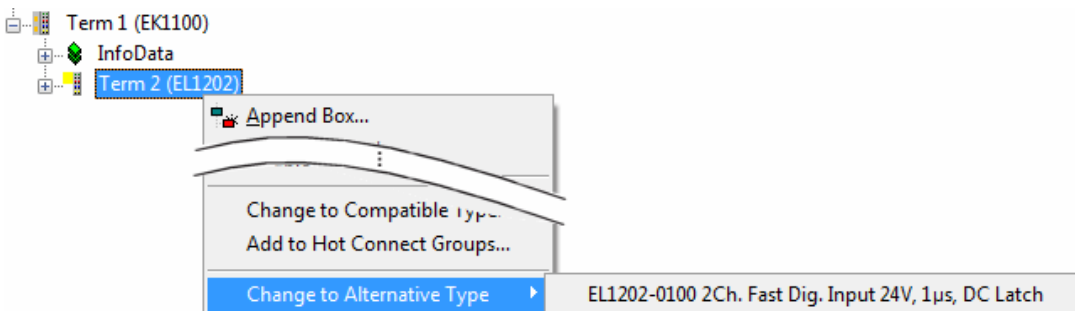


Fig. 113: TwinCAT 2 Dialog Change to Alternative Type

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

5.2.7 EtherCAT subscriber configuration

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

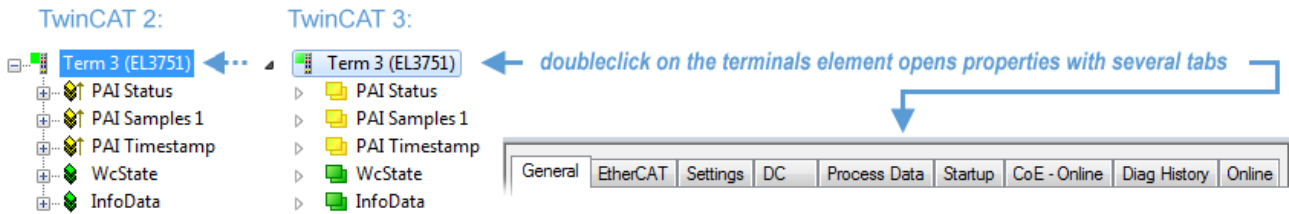


Fig. 114: Branch element as terminal EL3751

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

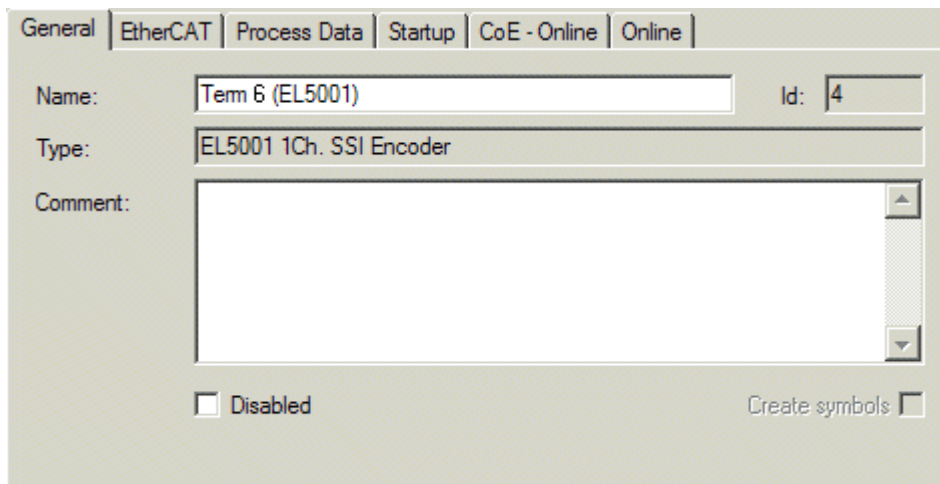


Fig. 115: “General” tab

- 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

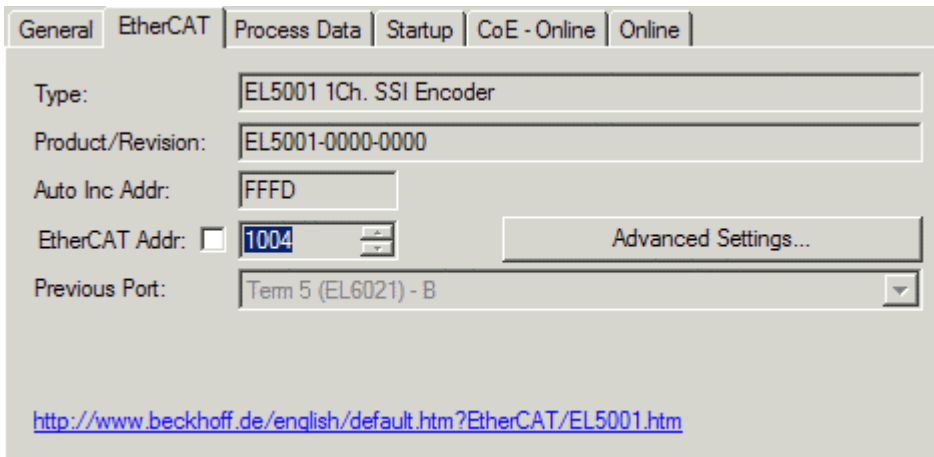


Fig. 116: “EtherCAT” tab

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 ring 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.

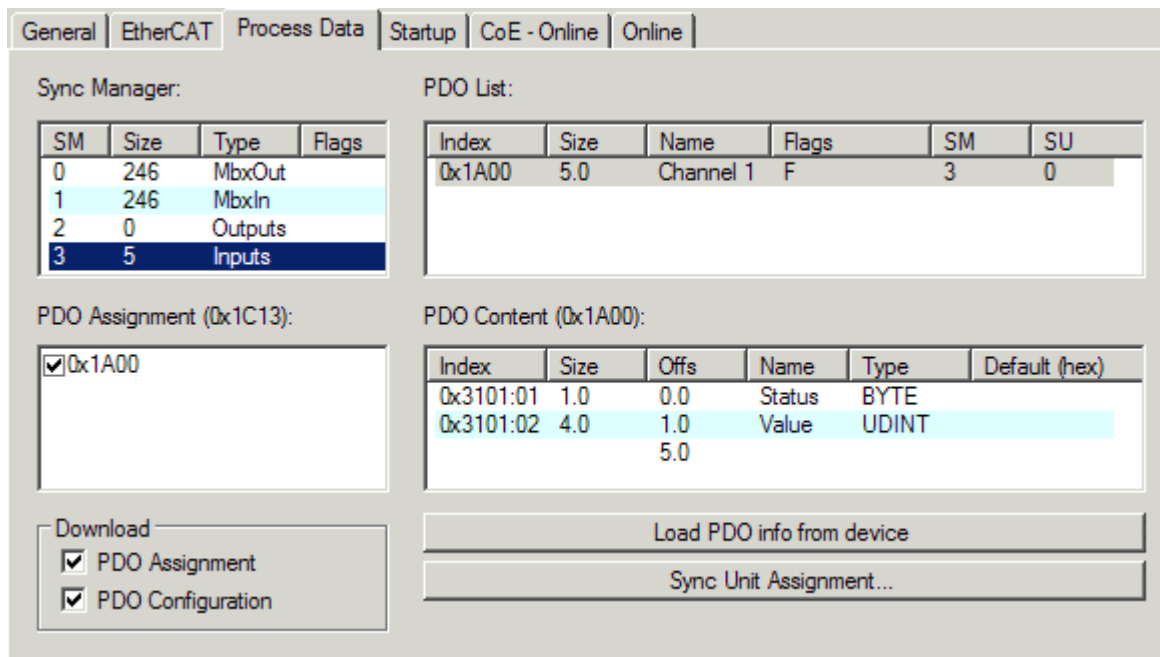


Fig. 117: "Process Data" tab

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").

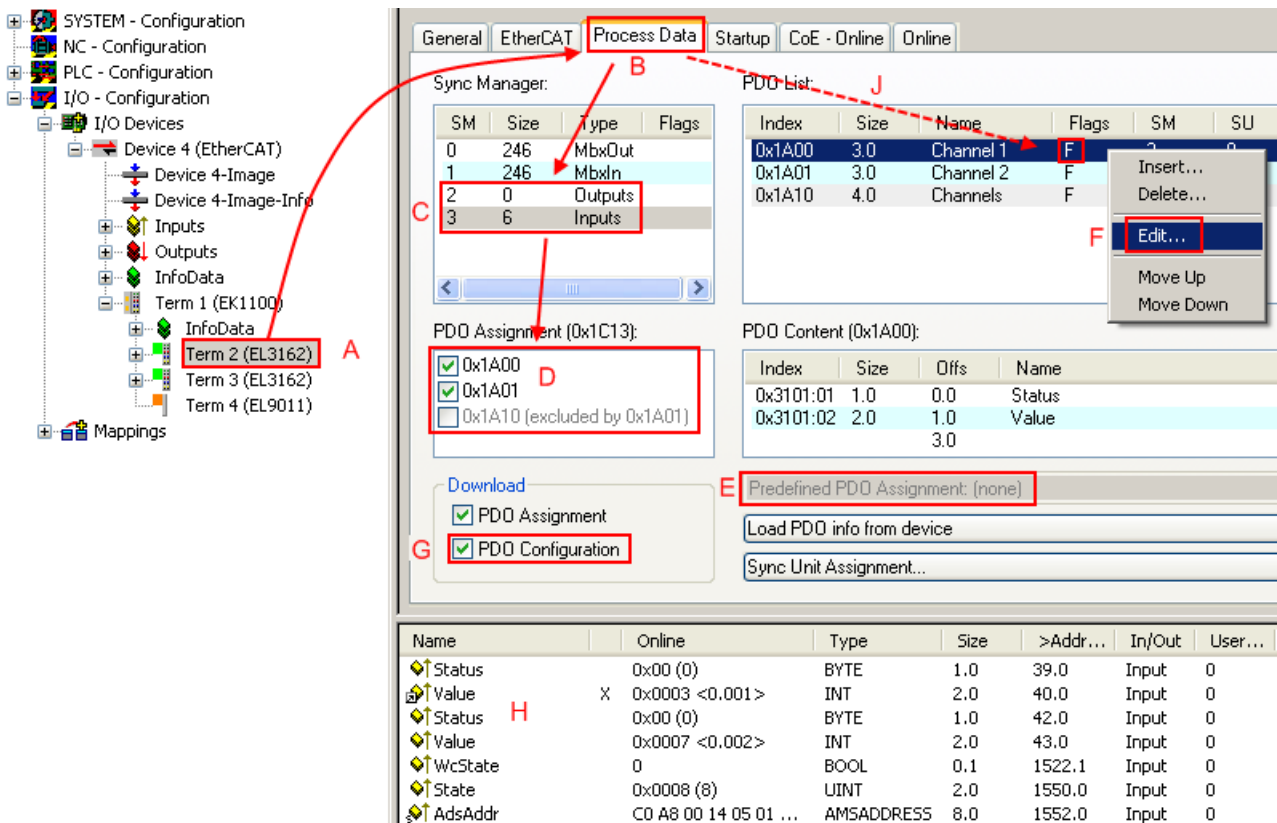


Fig. 118: Configuring the process data

i 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 [► 94] 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.

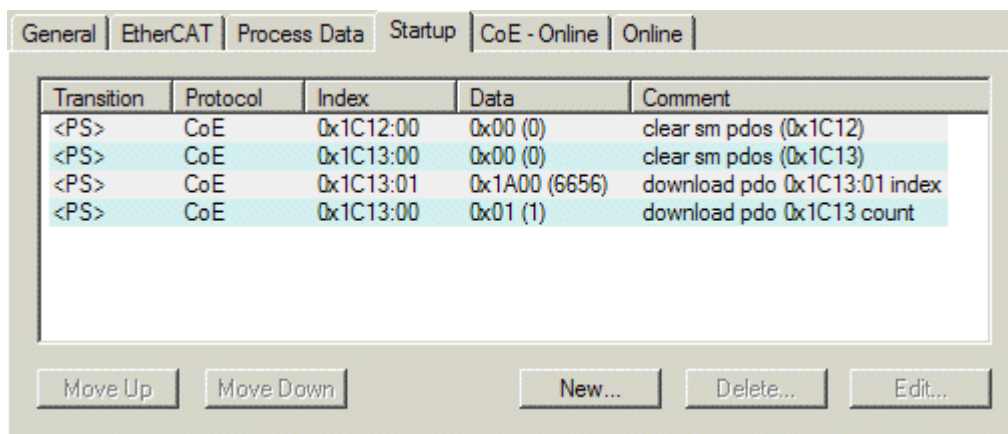


Fig. 119: "Startup" tab

Column	Description
Transition	Transition to which the request is sent. This can either be <ul style="list-style-type: none"> 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.

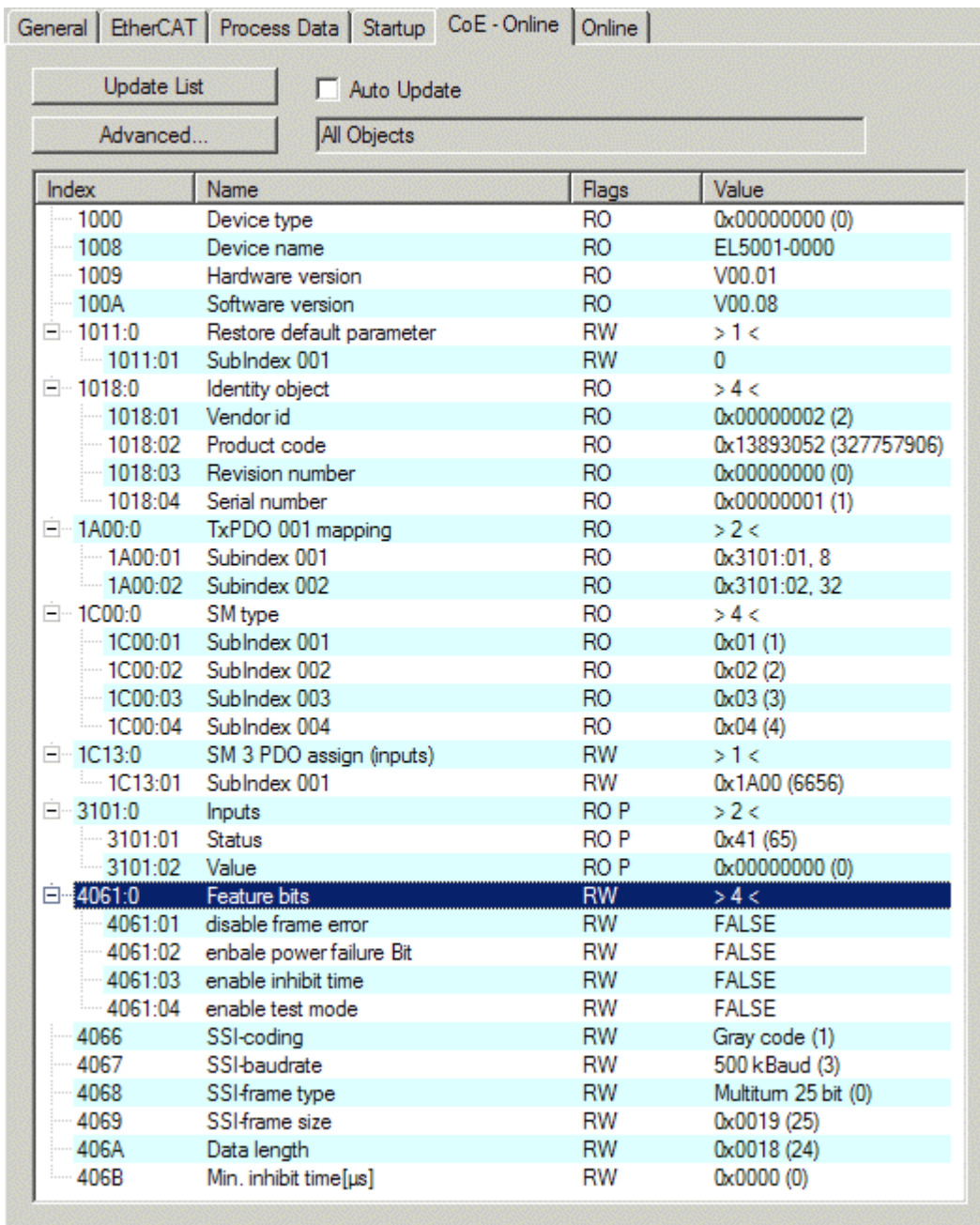


Fig. 120: "CoE - Online" tab

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.

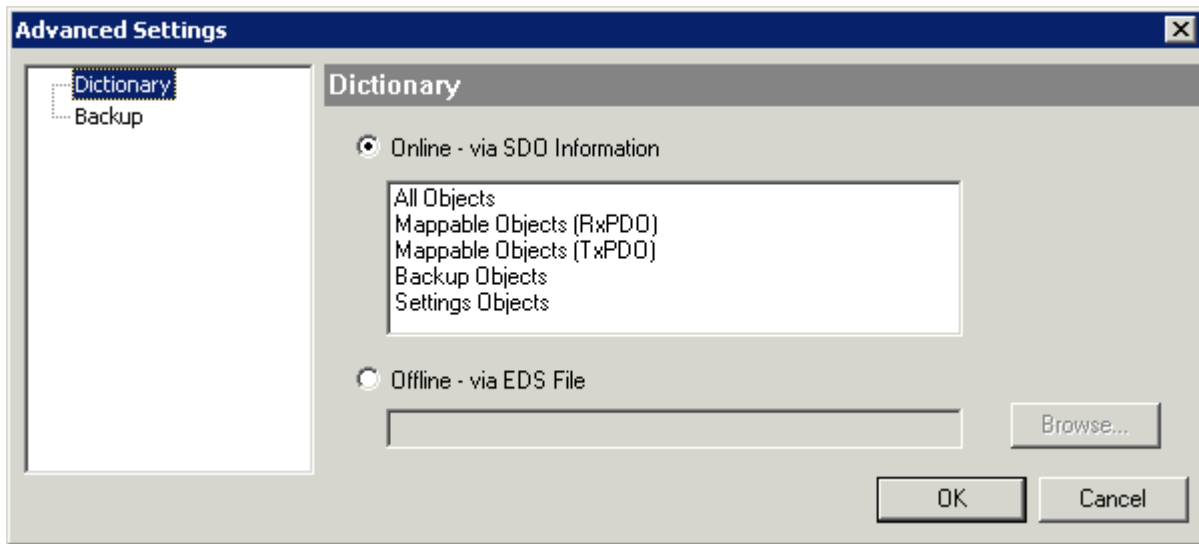


Fig. 121: Dialog “Advanced settings”

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

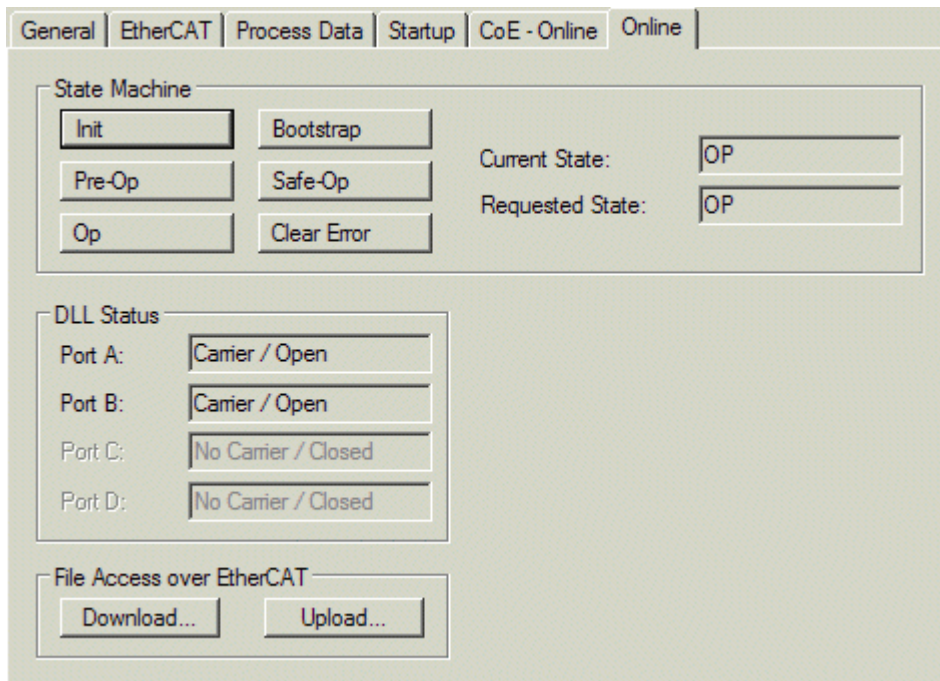


Fig. 122: “Online” tab

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)

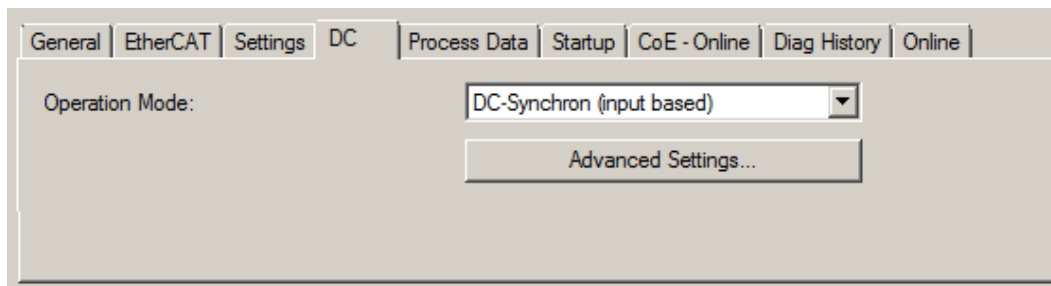


Fig. 123: “DC” tab (Distributed Clocks)

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

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

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

5.2.7.1 Detailed description of Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).

If the EtherCAT device has a mailbox, SM0 is used for the mailbox output (MbxOut) and SM1 for the mailbox input (MbxIn).

SM2 is used for the output process data (outputs) and SM3 (inputs) for the input process data.

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

PDO Assignment



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

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

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

i 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](#) [▶ 92]),
 - 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 <i>PDO Assignment</i> 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 \[► 89\]](#) tab.

PDO Configuration

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

5.2.8 Import/Export of EtherCAT devices with SCI and XTI

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

5.2.8.1 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.

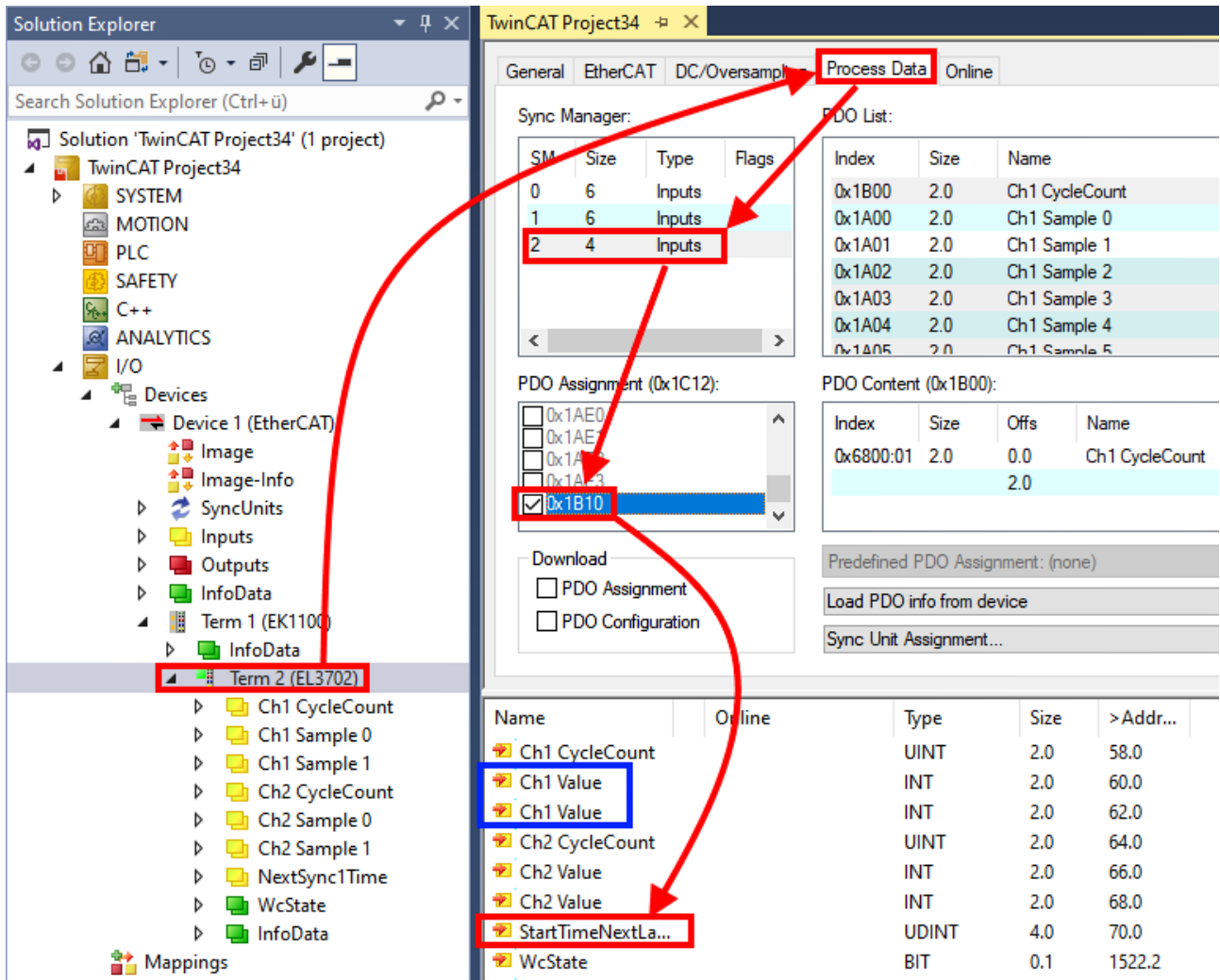
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

- 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 **x**ti file or
- outside, i.e. beyond the TwinCAT limits: Export/Import as **s**ci file.

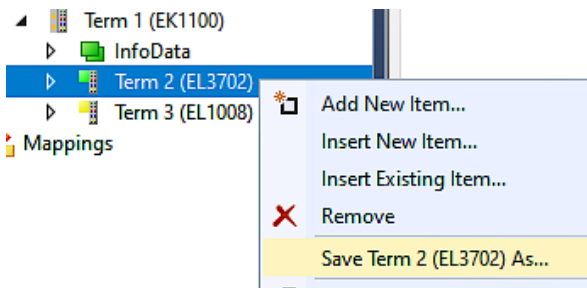
An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):



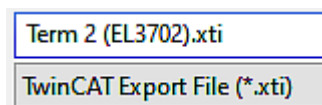
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

5.2.8.2 Procedure within TwinCAT with xti files

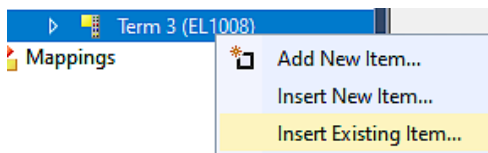
Each IO device can be exported/saved individually:



The xti file can be stored:



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



5.2.8.3 Procedure within and outside TwinCAT with sci file

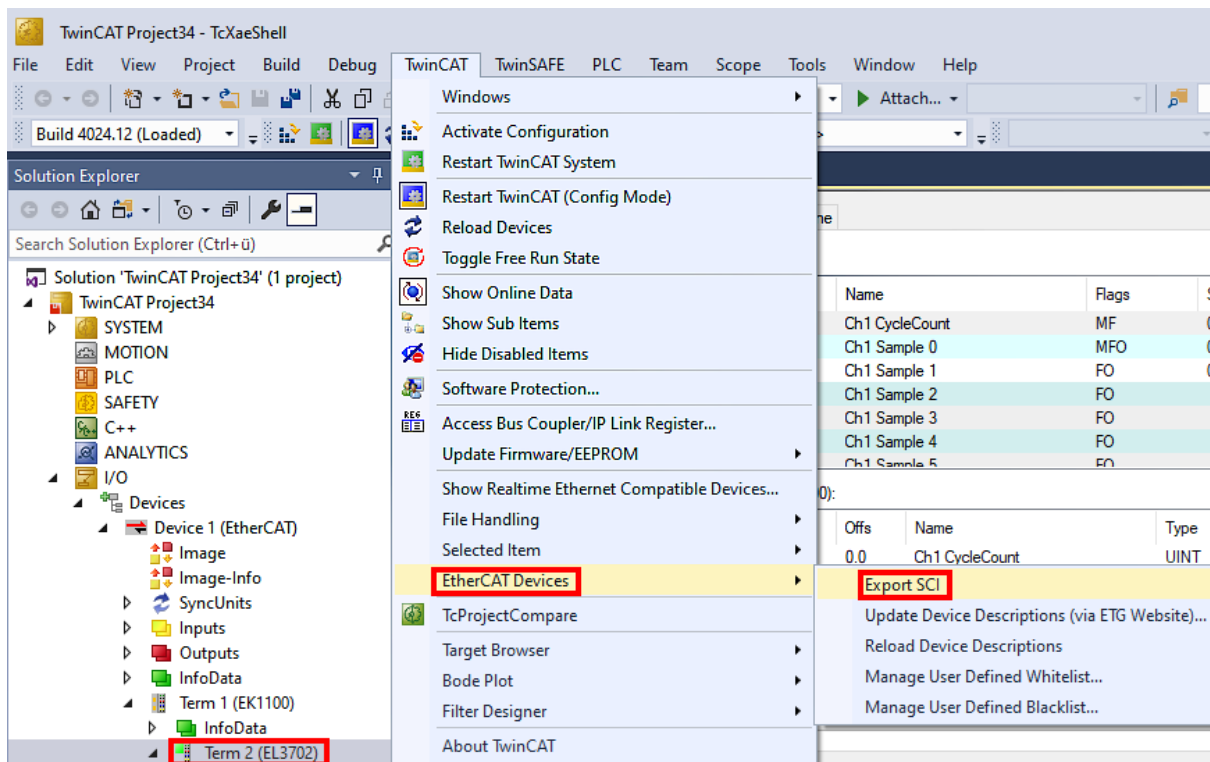
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

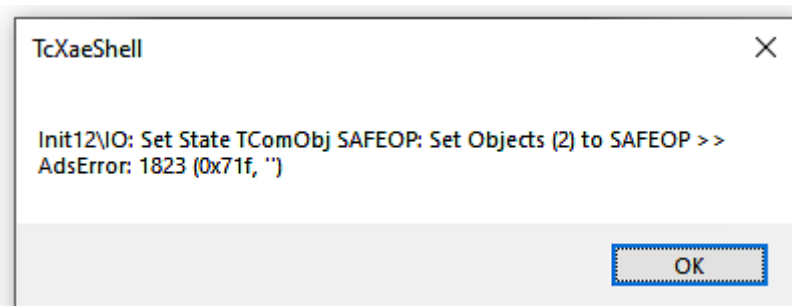
The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

Export:

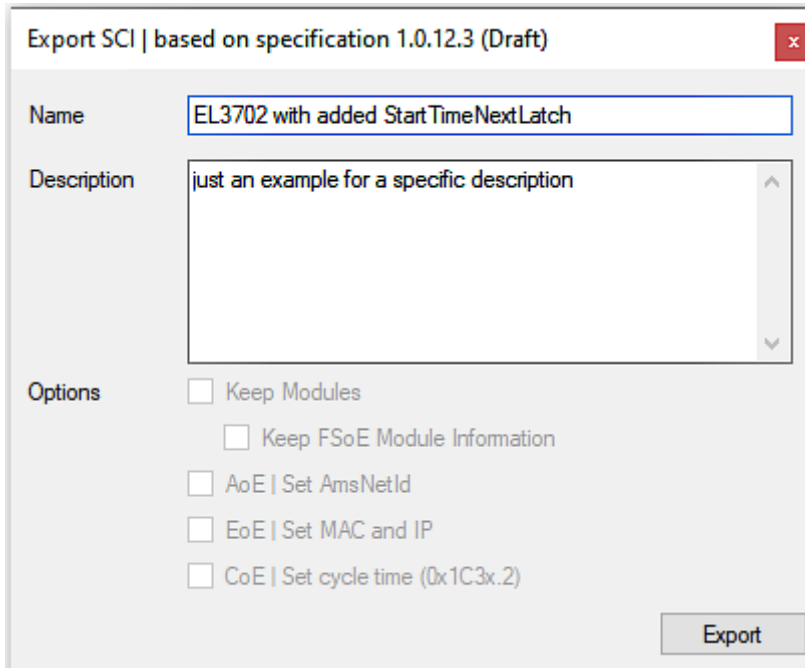
- select a single device via the menu (multiple selection is also possible):
TwinCAT → EtherCAT Devices → Export SCI.



- 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:



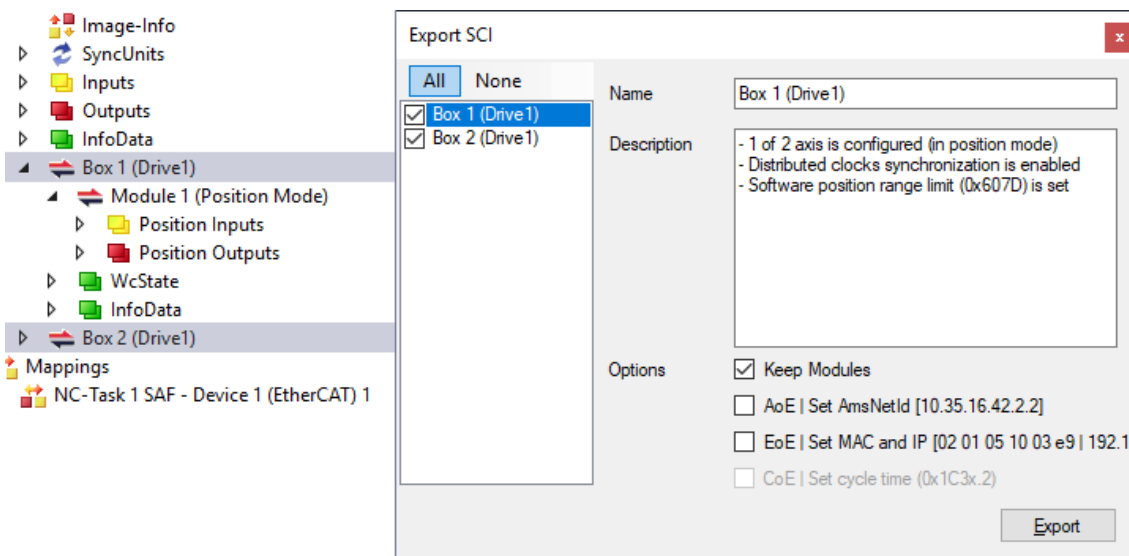
- A description may also be provided:



- 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*):

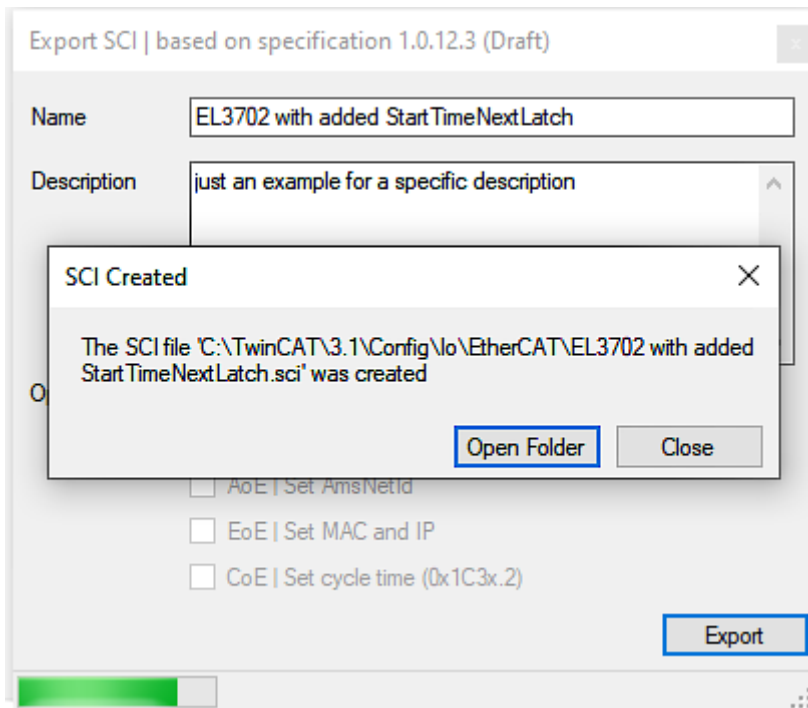


- Selection of the slaves to be exported:
 - All:
All slaves are selected for export.

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

Dateiname:
 Dateityp:

- The export takes place:

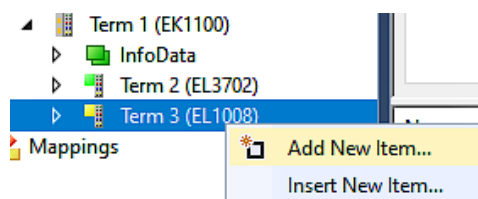


Import

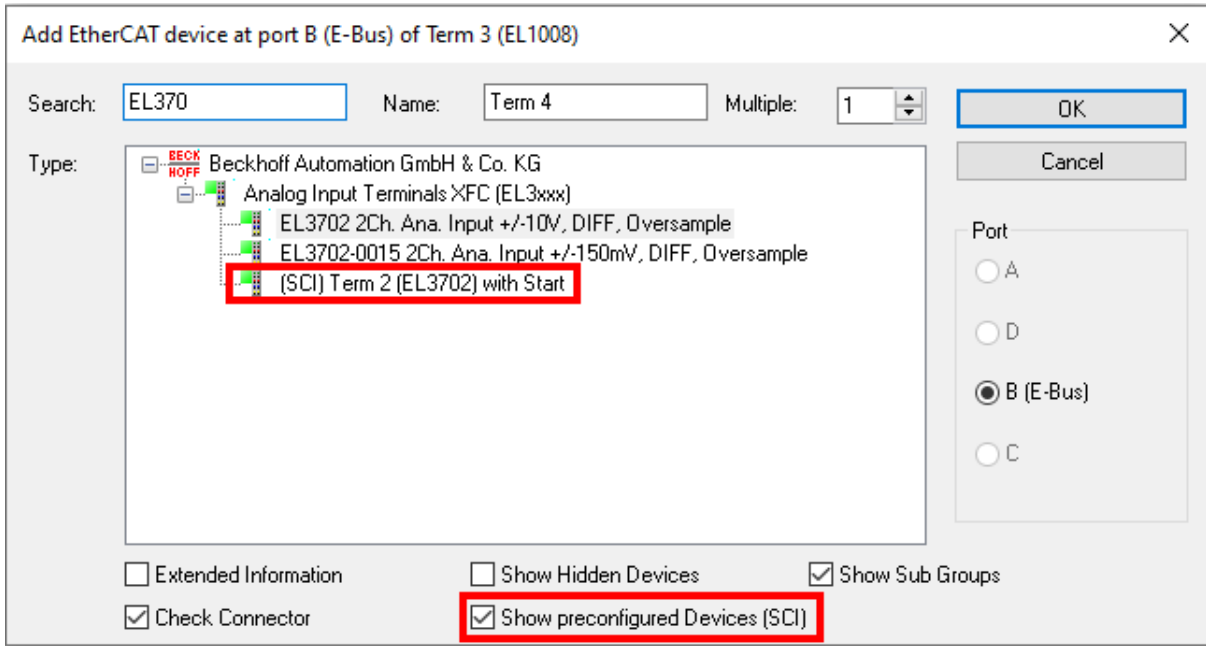
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under:
C:\TwinCAT\3.1\Config\Io\EtherCAT

	EL3702 with added StartTimeNextLatch.sci	11.01.2021 13:29	SCI-Datei	6 KB
--	--	------------------	-----------	------

- Open the selection dialog:

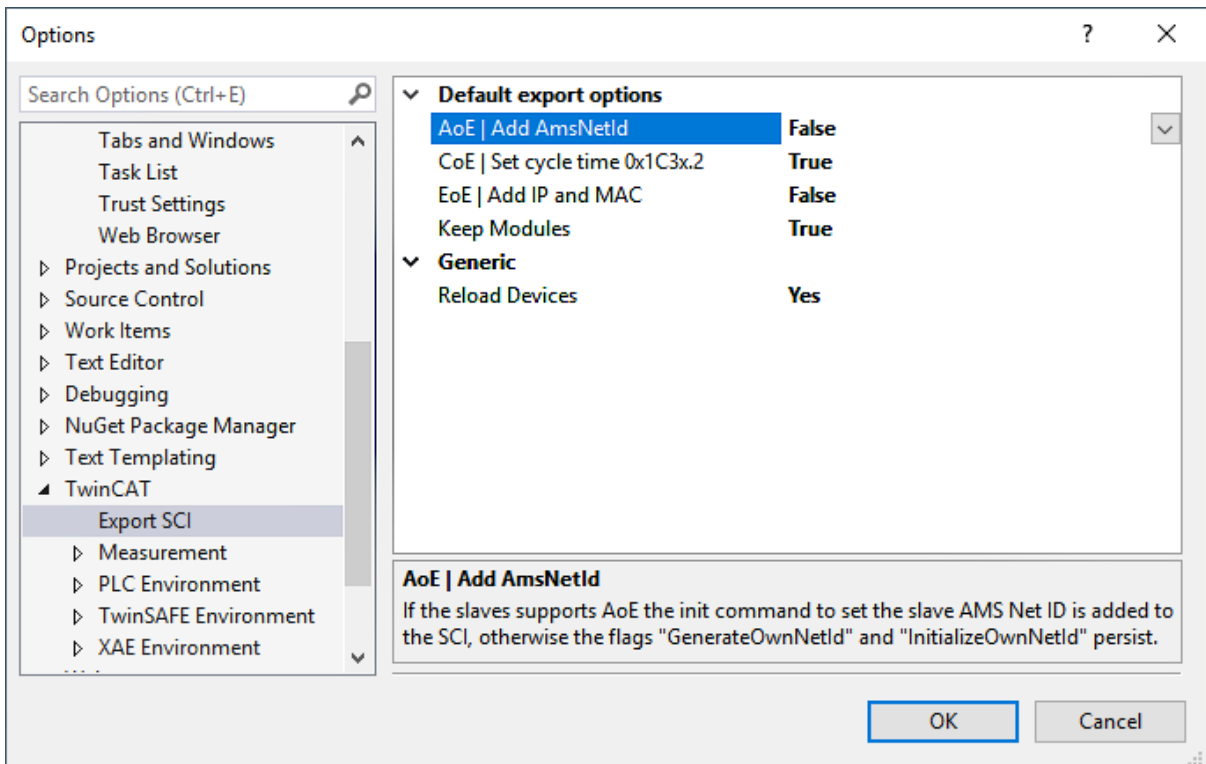


- Display SCI devices and select and insert the desired device:



Additional Notes

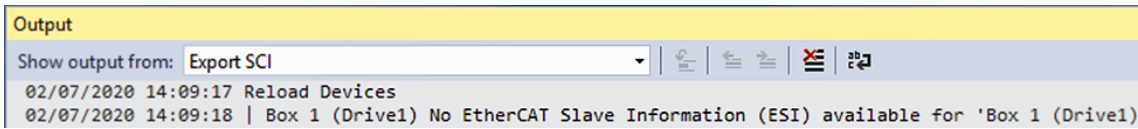
- Settings for the SCI function can be made via the general Options dialog (Tools → Options → TwinCAT → Export SCI):



Explanation of the settings:

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

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



5.3 Quick start

Connecting the terminal

Install the terminal as described in the chapter entitled [Installation](#) [▶ 25].

Parameterizing the terminal

1. If provided, the current ESI (XML file) EL2595-0000-xxxx is to be installed in TwinCAT (usually under C:\TwinCAT\IO\EtherCAT\).
Otherwise, please download the latest ESI from the [Beckhoff-Website](#).
2. If necessary, a firmware update is to be carried out. Check then in the online CoE.
3. Following the firmware update you are urgently recommended to carry out a "Restore default parameter" in the CoE: enter "0x64616F6C" in CoE index 0x1011:01
4. The key data of the intended LED must be determined:
 - Intended forward current (operating current) [mA]
 - Required forward voltage for that [V]
Reason: the current controller regulates the forward voltage itself in order to reach/maintain the desired load current.
Self-heating effects, for example, are thus compensated. So that the load current is as accurate as possible in the first few milliseconds after switch-on, however, the controller needs to know the approximate forward voltage that it has to operate with.
 - These data are to be entered in the online CoE:
 - i. [0x8000:03](#) [▶ 132] actual supply voltage (minimum 12 V permissible)
 - ii. [0x8000:02](#) [▶ 132] desired output current of the LED
 - iii. [0x8000:04](#) [▶ 132] necessary forward voltage on the LED
5. Further useful CoE settings
 - Supply voltage and output voltage are monitored. Exceeding/undershooting these limits, which must be set (in ± % of the desired value), produces first a warning, then an error. This can look like the following, for example:

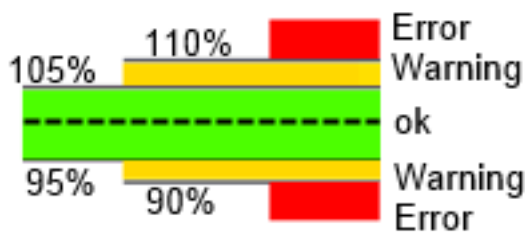


Fig. 124: Representation of warning thresholds

The percentage value (e.g. here: warning: 5 for ±5%, error: 10 for ±10%) must be entered in the CoE index [0x8000:11-14](#) [▶ 132].

- If the EtherCAT connection fails, the EL2595 can automatically go into an auto-flash mode. The failure is detected by the SM watchdog (default setting 100 ms).
 - i. [0x8002:04](#) [▶ 133] = TRUE, Enable emergency flash
 - II. [0x8003:ff](#) [▶ 133] the flash sequence is set here as on/off times [ms]
0 = deactivated; an on/off pair must always be used

- Limitation of the pulse time for all operation modes to protect the connected HW, can also be used for cooling the LED
 - i. Set in [0x8000:09](#) [[▶ 132](#)] and [0x8000:0A](#) [[▶ 132](#)]
 - ii. 0 = deactivated, no monitoring takes place
 - iii. The *ReadyToActivate* PDO is FALSE as long as the OFF time is running
- EnableAutomaticResetOnError in [0x8002:05](#) [[▶ 133](#)]
If an error occurred during the pulse (signaled in the PDO and DiagData), the error message is automatically deleted if the output is set to FALSE. Otherwise a manual Reset via the PDO is necessary.
- EnableAutomaticSavingOfCounter in [0x8002:06](#) [[▶ 133](#)]
If activated, operating hour counter [0x9000:11](#) [[▶ 134](#)] (unit [minute]) and switching counter [0x9000:12](#) [[▶ 134](#)] are saved locally every 15 minutes. Saving also takes place on the state transition OP → INIT or on request by command
- The information to be transmitted for the real-time diagnosis can be set in [0x8002:11](#) [[▶ 133](#)] and [0x8002:19](#) [[▶ 133](#)].
- Operation mode of the input

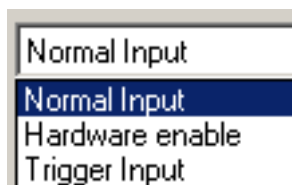


Fig. 125: Selection of the mode of operation

- i. Normal input: no particular function for the LED function, standard input
 - ii. HW Enable: if this function is set, the Emergency auto-flash mode is only active if TRUE is present at the input.
 - iii. Trigger:
 1. If the PulseOn/Off time from [0x8000:09](#) /[0A](#) = 0 → each rising edge at the input toggles the LED status
 2. If these times are $\neq 0$, a pulse is output as defined in the times
 3. The jitter for the internal acceptance of the trigger signal is about 5 μ s
 - Diagnosis over CoE
 - i. [0x9000:01](#) [[▶ 134](#)], bit-by-bit collection of various data, list to follow
 - ii. [0x9000:08](#) [[▶ 134](#)], current output voltage
 - iii. [0x9000:09](#) [[▶ 134](#)], last measured current value in the ON state
 - iv. [0x9000:11](#) [[▶ 134](#)], totaled On operating time of the LED in [min]; all ON times are added together, even if a pulse is only a few μ s long
 - v. [0xA000:01](#) [[▶ 134](#)], saturated: Output stage is overloaded, target voltage can no longer be maintained
 - vi. [0xA000:04](#) [[▶ 134](#)], [05](#) [[▶ 134](#)], [0A](#) [[▶ 134](#)], [0B](#) [[▶ 134](#)]: Warning based on the thresholds from [0x8000:11-14](#) [[▶ 132](#)]; if [0xA000:09](#) [[▶ 134](#)] is set at the same time, it is the corresponding error message
 - vii. [0xA000:06](#) [[▶ 134](#)], current > 4.2 A for several cycles is evaluated as an external short-circuit
 - viii. [0xF900](#) [[▶ 135](#)], supply current: Maximum value since last reading; it is deleted after reading
6. For safety's sake these CoE data can also be entered into the Startup list. If the EL2595 is exchanged later, the online CoE data are no longer available. The system manager then writes the data to the new terminal.
7. The output can now be switched on via the process data (PDO) *Control.Output* = TRUE or a pulse of a defined length can be output in DC mode via a timestamp.
8. The present set output current can also be specified from the controller by cyclic PDO:
- Activate PDO

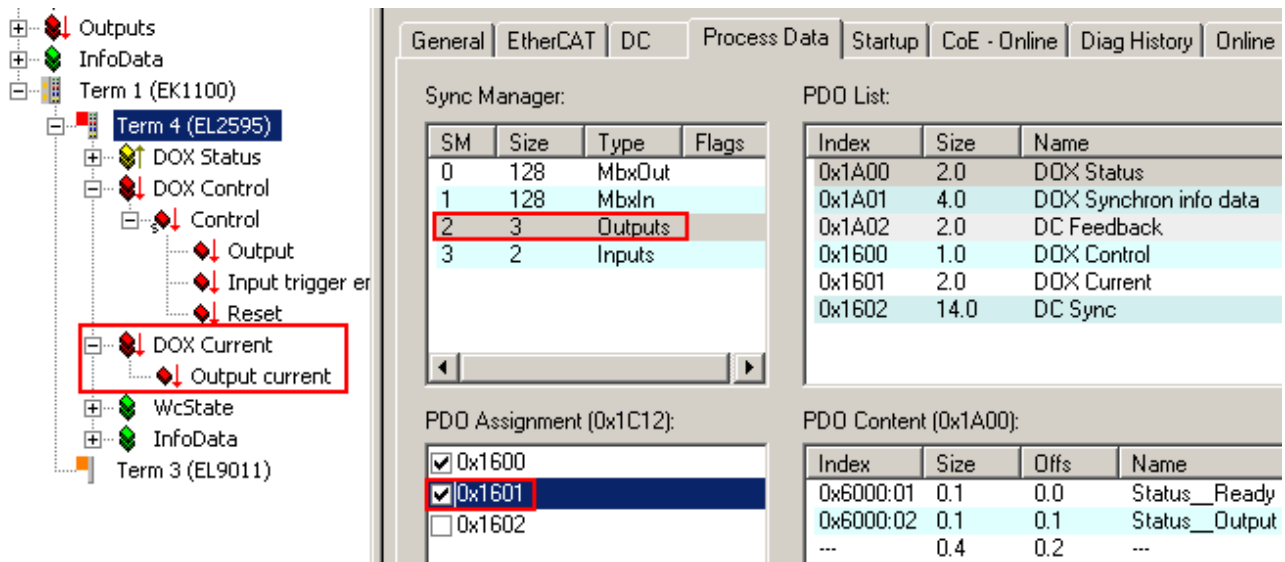


Fig. 126: Activating PDO

- To switch on the terminal, specify a current value [mA] and then switch on the output by `Control.Output = TRUE`.
- The CoE values `0x8000:02 [▶_132]`, `0x8000:04 [▶_132]` are then the upper limiters for the operation. Nevertheless, a forward voltage suitable for the load must be specified in the CoE (see above) for a correct forward current.

Additional Notes

- 2013-04-19: Serial/parallel switching of several EL2595s in parallel via the SyncIn/Out connection is not yet implemented.
- The output voltage U_{out} may not lie within a window of $U_{in} \pm 10\%$.

5.4 General Notes - EtherCAT Slave Application

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

Any application that relies on I/O data from a fieldbus being correct and up to date must make diagnostic access to the corresponding underlying layers. EtherCAT and the TwinCAT System Manager offer comprehensive diagnostic elements of this kind. Those diagnostic elements that are helpful to the controlling task for diagnosis that is accurate for the current cycle when in operation (not during commissioning) are discussed below.

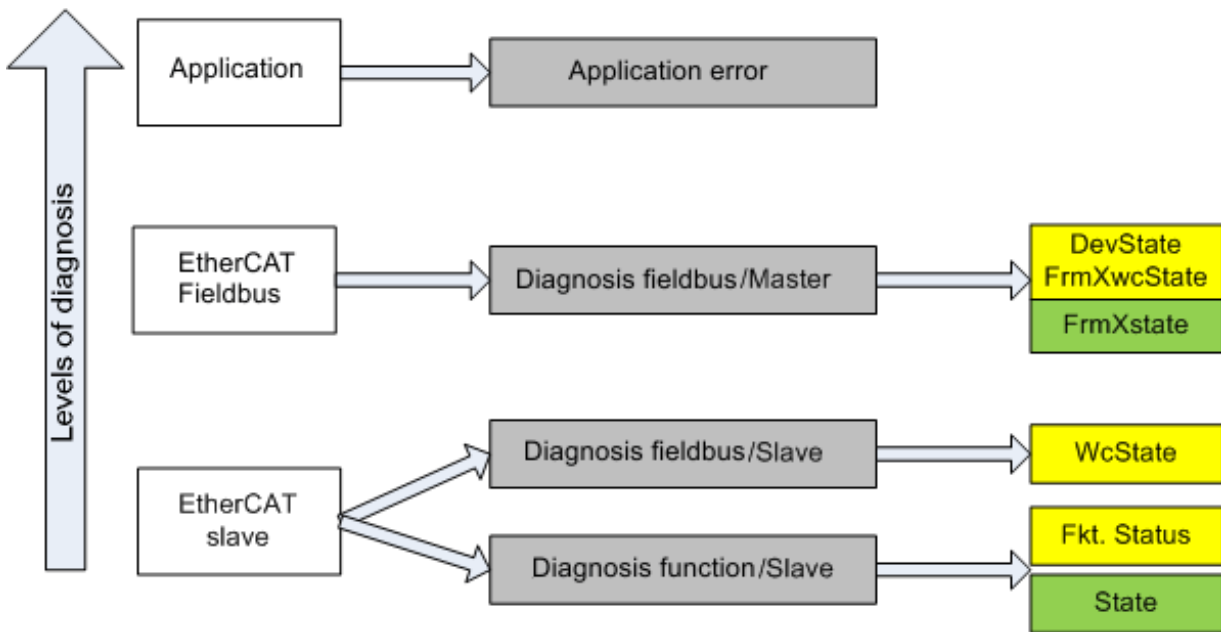


Fig. 127: Selection of the diagnostic information of an EtherCAT Slave

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
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.

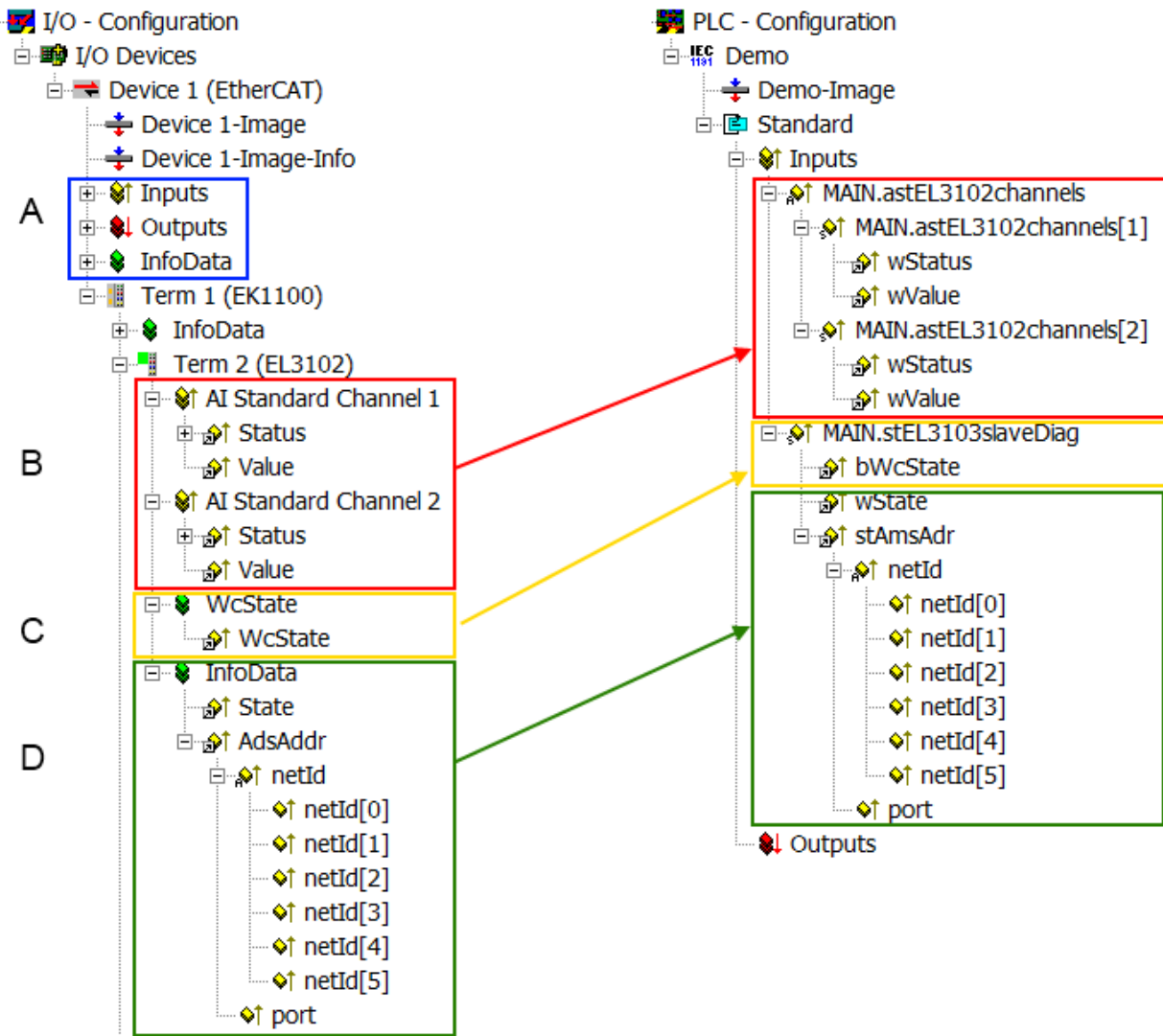


Fig. 128: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: <ul style="list-style-type: none"> • CoE in the Master for communication with/through the Slaves • Functions from <i>TcEtherCAT.lib</i> • Perform an OnlineScan
B	In the example chosen (EL3102) the EL3102 comprises two analogue input channels that transmit a single function status for the most recent cycle.	Status <ul style="list-style-type: none"> • 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 <ol style="list-style-type: none"> 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 <ul style="list-style-type: none"> • 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. <i>AdsAddr</i> 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 <i>port</i> (= EtherCAT address).	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore possible to read such variables through ADS.

NOTE

Diagnostic information

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

CoE Parameter Directory

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

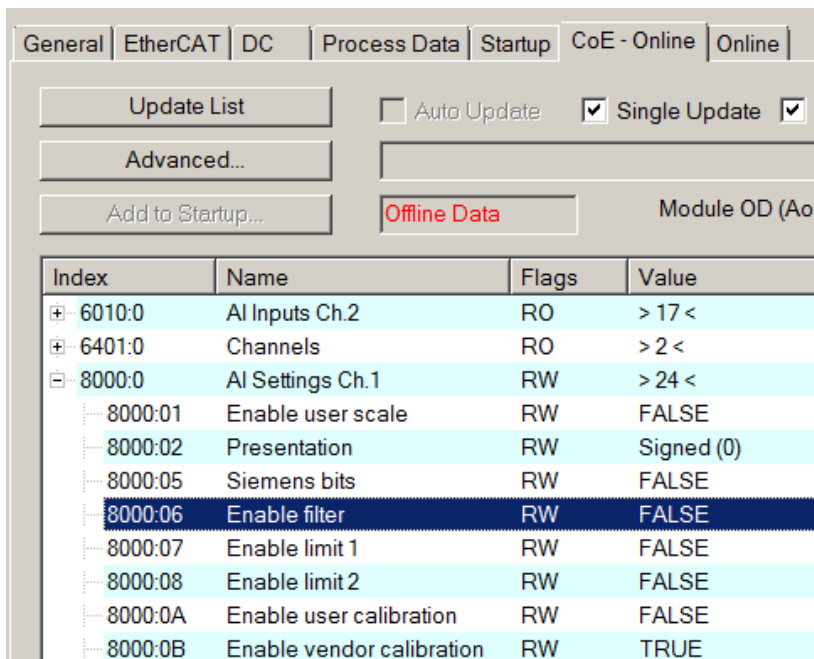


Fig. 129: EL3102, CoE directory

i 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.

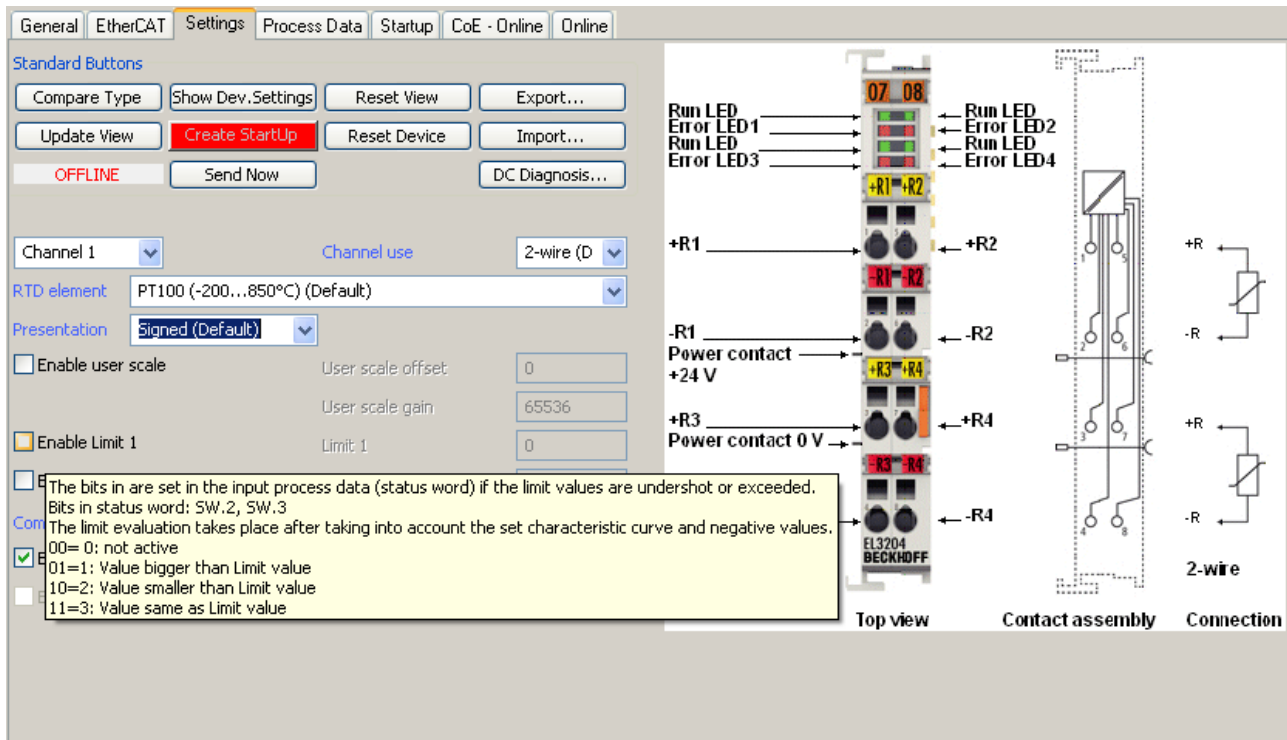


Fig. 130: Example of commissioning aid for a EL3204

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 \[▶ 17\]](#)" 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.

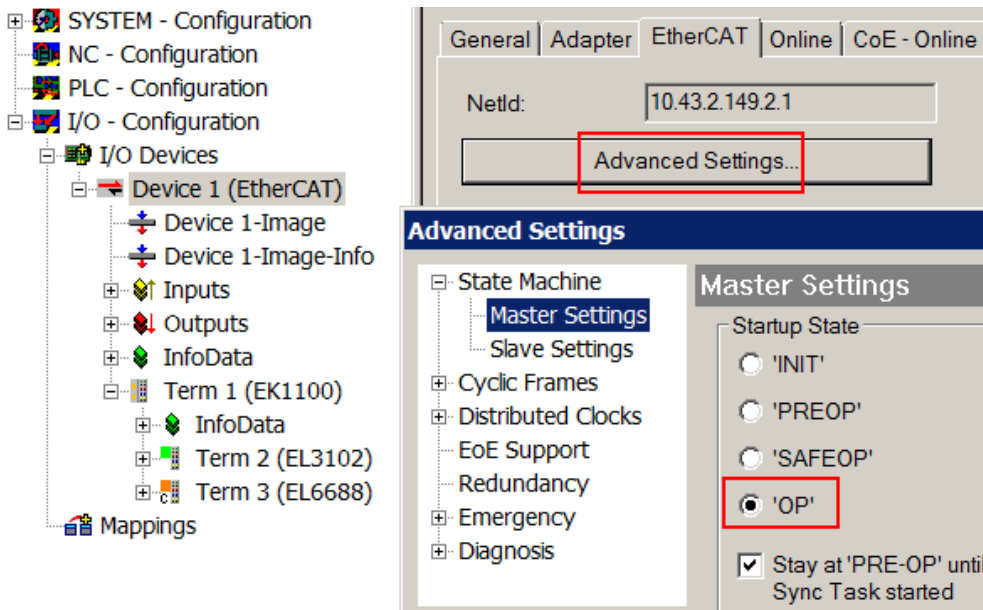


Fig. 131: Default behaviour of the System Manager

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

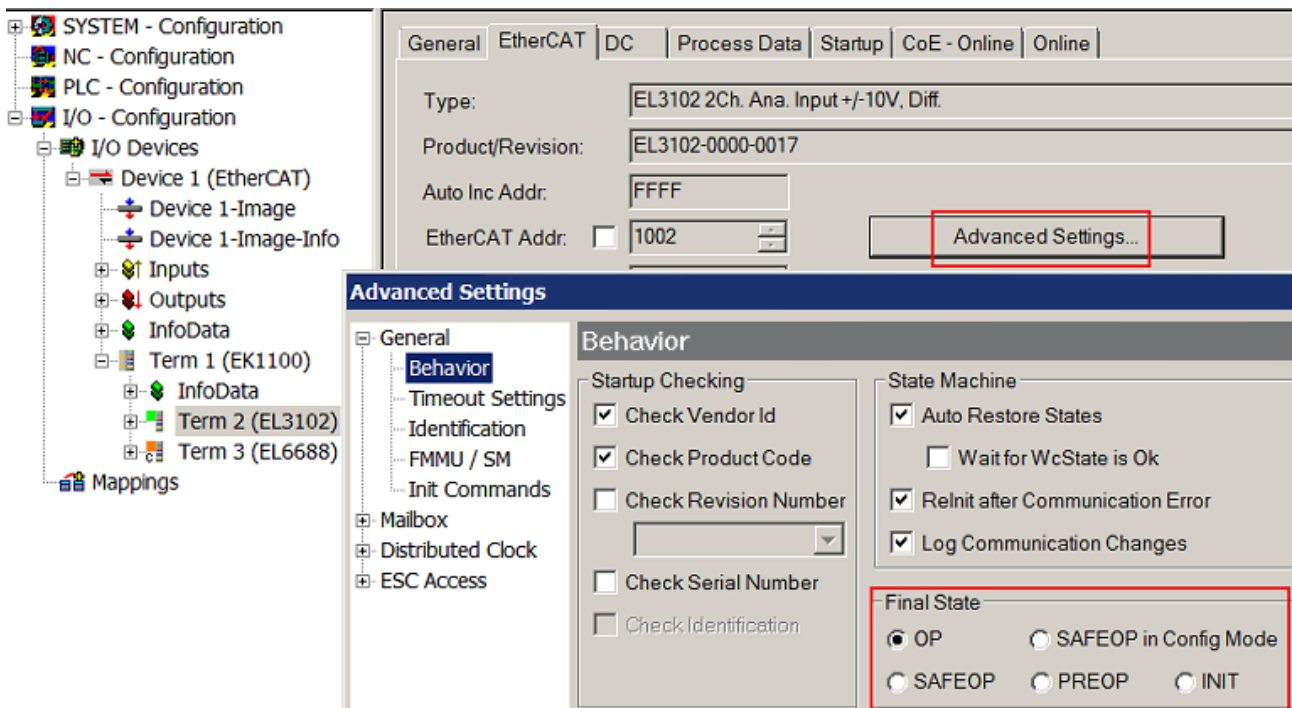


Fig. 132: Default target state in the Slave

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.

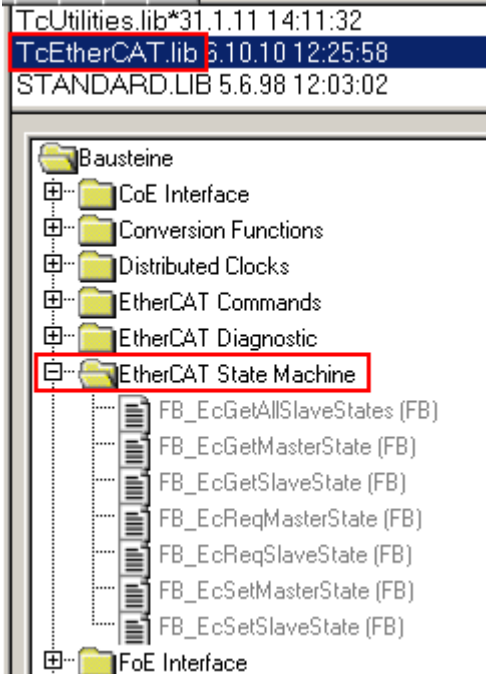


Fig. 133: PLC function blocks

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.

General Adapter EtherCAT Online CoE - Online						
NetId:		10.43.2.149.2.1		Advanced Settings...		
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. 134: 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:

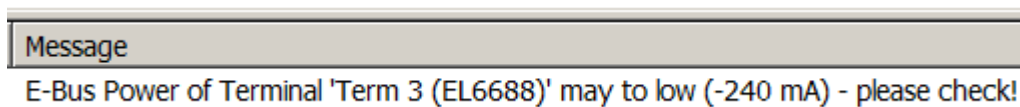


Fig. 135: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

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

5.5 Basic Function Principles

The EL2595 offers the user various possibilities to switch the digital output.

The different functions are described below:

5.5.1 General procedure

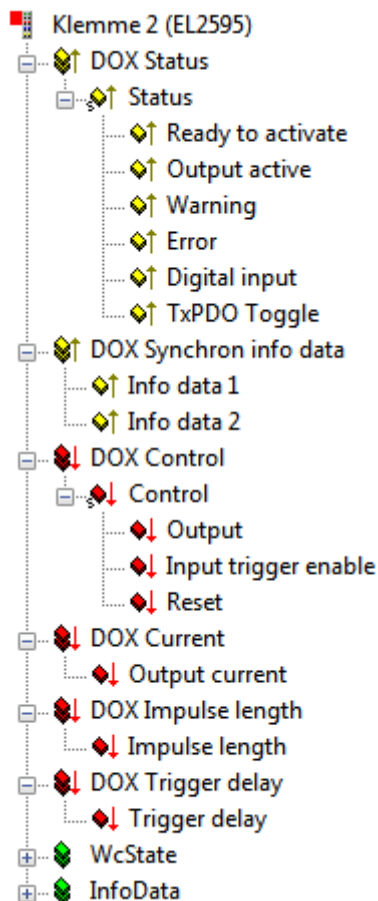


Fig. 136: All PDOs for the synchronous operation

All signals that can be used to control the digital output are shown in the figure *PDOs for synchronous operation*. In standard applications only the PDOs "DOX Status" and "DOX Control" are required as the minimum.

The user gets a "Ready to activate" first. The terminal is then ready to switch the output. If this bit is not set they may be an **Error**. In this case the Diagnostic Objects Index 0x9000, 0xA000 and 0xF900 should be checked and an error evaluation carried out. A further possibility is that the "Pulse off time" has not yet expired and the load is being given time to cool down.

If the terminal is ready, the desired "**Output current**" can now be specified with a resolution of 1 mA. If only one pulse is to be output, a time can be specified by writing the "**Impulse length**" with a resolution of 1 μ s.

4 cases can be distinguished in the combination of the PDO and CoE values:

1. "PDO: **Impulse length**" = 0, "CoE: **Pulse on time**" = 0
-> output permanently active
2. "PDO: **Impulse length**" > 0, "CoE: **Pulse on time**" = 0
-> output active for the time of "**Impulse length**"
3. "PDO: **Impulse length**" > 0, "CoE: **Pulse on time**" > 0
-> output active for the time "**Impulse length**", but maximally for the time "Pulse on time"
4. "PDO: **Impulse length**" = 0, "CoE: **Pulse on time**" > 0
-> output is active for the time "**Pulse on time**"

Now only "Output" needs to be set in order to activate the output. For the period in which the output is active, the user receives feedback in "Output active".

● Priorities

i When outputting pulses it must be noted that "Output" always has the highest priority. This means that the output is always deactivated if "Output" is deleted, even if a preset impulse time has not yet expired.

● Limitation of process data

i The process data values "Output current", "Impulse length" and "Trigger delay" are upwardly limited in their range of values by their corresponding entry in the CoE. The CoE parameters are therefore to be configured in such a way that the connected load can be destroyed by a bad combination of the different process data.

NOTE

Avoidance of incorrect warning and error messages due to dynamically specified output currents

With dynamically specified output currents, note that the "Warning Level [▶ 142]" and the "Error Level [▶ 142]" must be adapted or deactivated so that an undesired warning or error state is not falsely generated.

The control of the output current is optimized to a static current value. This means that the edge steepness of the current is considerably larger in the case of a fixed output current than with a dynamically changed current.

5.5.2 Normal digital output

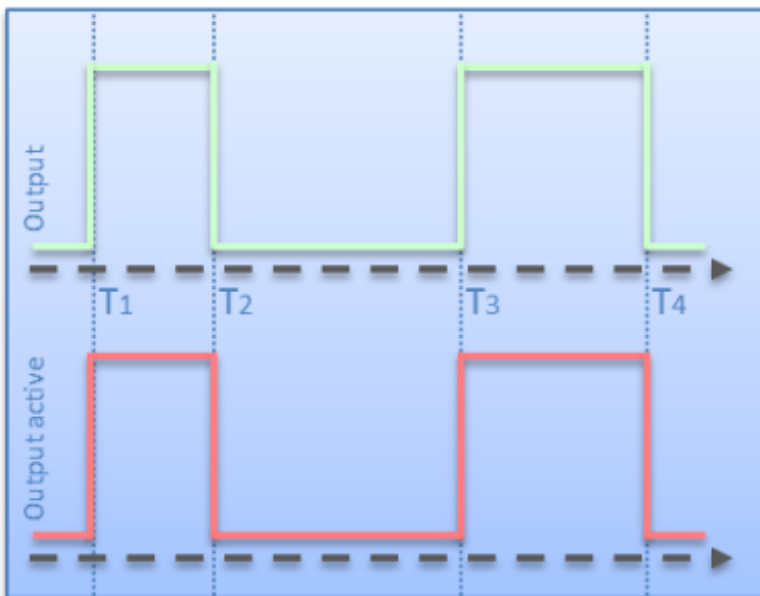


Fig. 137: Normal digital output

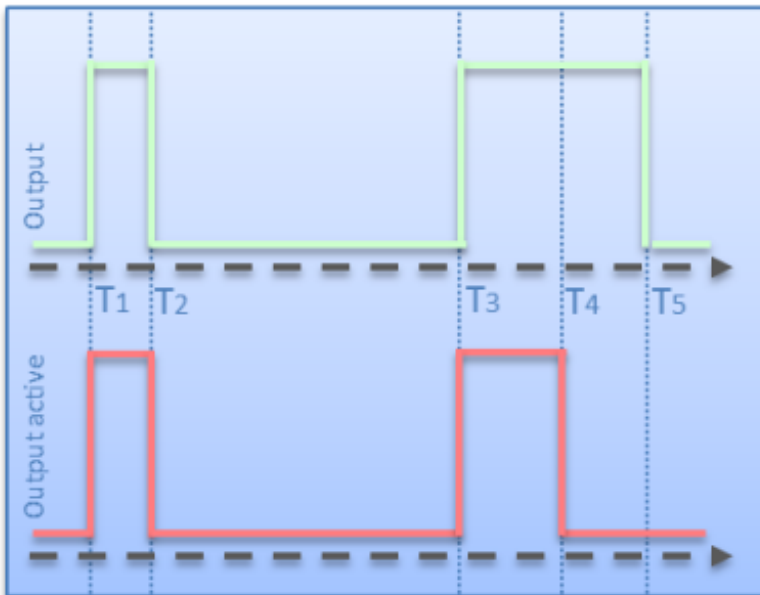


Fig. 138: Digital output with specification of the pulse length

In the case of the "normal digital output", the input signal "**Output**" is switched directly through to the driver stage and returned as feedback in "**Output active**" (see fig. *Normal digital output*).

T1 and T3 thereby represent the switch-on times and T2 and T4 the switch-off times.

The temporal behavior with "**Impulse length**" additionally activated is shown in fig. *Digital output with preset pulse length*:

- T1 and T3 are the switch-on times
- T4 - T3 is the "**Impulse length**"
- T2 - T1 has a shortened duty cycle due to the premature resetting of "**Output**"
- The resetting of "**Output**" at time T5 has no effect on the output, since the output time has already expired

5.5.3 External trigger input

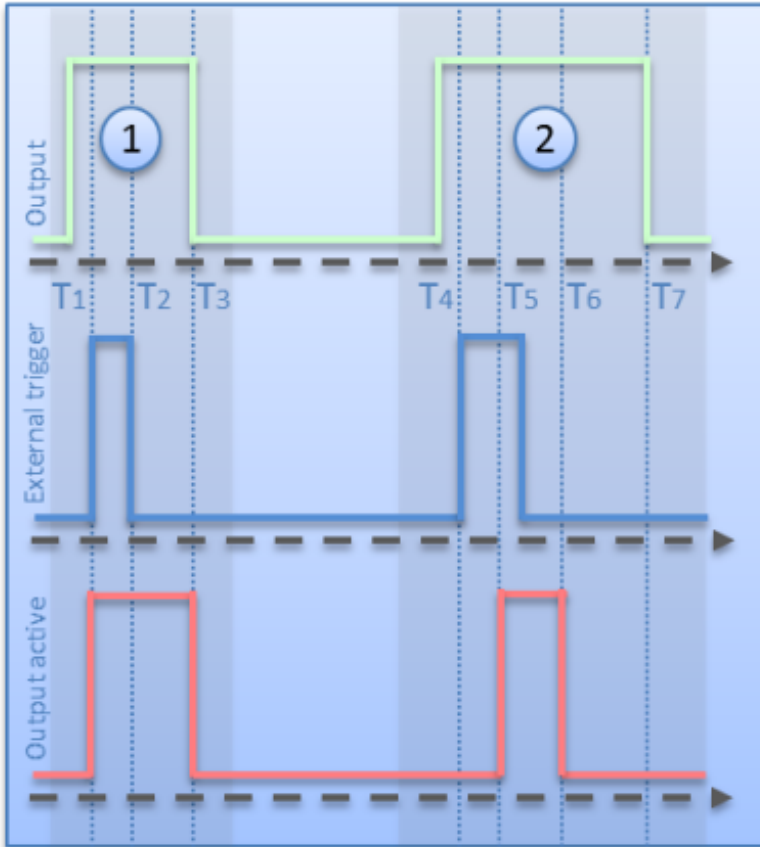


Fig. 139: Digital output with external trigger and specification of pulse length and trigger delay

When using the digital input as a trigger pulse, this must first be configured as **"Trigger input"** in the CoE (see [DOX Features](#) |> 143).

In turn, **"Output"** and additionally **"Input trigger enable"** must be activated first in order to enable the trigger function.

Two different cases of the output of a pulse by trigger input are shown in the fig. *Digital output with external trigger and preset pulse length and trigger delay*:

Case 1:

- **"Output"** and **"Input trigger enable"** are already activated at T1 and the external trigger event is detected.
- **"Output active"** is set immediately, since **"Trigger delay"** has the value 0.
- After the **"Impulse Length"** T3 - T1 or if **"Output"** is deleted, **"Output active"** is likewise reset.

Case 2:

- **"Output"** and **"Input trigger enable"** are already activated at T4 and the external trigger event is detected.
- **"Output active"** is set for the duration of **"Impulse Length"** T6 - T5 on expiry of the time **"Trigger delay"**.
- The resetting of **"Output"** at time T7 has no effect on the output, since the output time has already expired.

4 cases can be distinguished in the combination of the PDO and CoE values:

1. "PDO: Trigger delay" = 0, "CoE: Trigger delay" = 0
-> no delay of the digital input

2. "PDO: Trigger delay" > 0, "CoE: Trigger delay" = 0
-> delay of the digital input specified by the PDO value
3. "PDO: Trigger delay" > 0, "CoE: Trigger delay" > 0
-> delay of the digital input specified by the PDO value, but maximally for the time configured in the CoE
4. "PDO: Trigger delay" = 0, "CoE: Trigger delay" > 0
-> no delay of the digital input

● Limitation of process data

I Unlike in the case of "**Impulse length**", "**Trigger delay**" is not overwritten with the associated CoE value if the process data value is 0, but is left as 0.

● Trigger function with active watchdog

I By means of the configuration of "**Trigger input (Watchdog disabled)**" in [Index 0x8002:32 \[▶ 133\]](#) (Function for input), the trigger input can be kept active in the case of a bus error->Watchdog active and pulses can still be output.

5.5.4 Distributed Clocks controlled output

The description of the control of the output by Distributed Clocks is somewhat more extensive and is therefore described in a [separate document \[▶ 125\]](#).

5.5.5 Emergency Flash

In case of a bus error the internal watchdog is activated. The terminal can be configured in such a way ("**Enable emergency flash**" = 1, see [Index 0x8002:04 \[▶ 133\]](#)) that it outputs a pulse sequence in this state - the Emergency Flashing. This sequence can consist of a maximum of 6 pulses. The On/Off times are set in the object "[DOX Flash Timing \[▶ 144\]](#)".

Note that a complete pair of "**ON-Time**" and "**OFF-Time**" must always be set. At the first time that is written with 0 the terminal starts again automatically with the first pulse.

5.5.6 Operating hour and switching cycle counter

The terminal internally counts the duty cycle ([Index 0x9000:11 \[▶ 134\]](#)) (resolution: 1 minute, accuracy: approx. 50 µs) and the number of switching cycles ([Index 0x9000:12 \[▶ 134\]](#)) of the connected load. These values are cyclically saved every 15 minutes in the internal memory if "[Enable automatic saving of counter \[▶ 143\]](#)" ([Index 0x8002:06 \[▶ 133\]](#)) is activated.

For the application it may be necessary for this saving to be acyclically triggered as well. The user can use the "[Command \[▶ 145\]](#)" object ([Index FB00 \[▶ 136\]](#)) for this. The two counters can be saved or deleted with the commands 0x0001 and 0x0002 respectively.

5.6 Process data

The process data overview lists the detailed PDO selection. These data are not usually necessary for operation under TwinCAT, since they can be simply configured from the configuration interface via the process data preselection.

5.6.1 Preselection of process data

An EtherCAT device usually offers several different process data objects (PDO) for input and output data, which can be configured in the System Manager, i.e. they can be activated or deactivated for cyclic transmission. See further below for the corresponding overview. Attention is thereby to be paid to the compatibility of input and output PDO.

From TwinCAT 2.11 with the EtherCAT devices intended for the purpose according to the ESI/XML description, the process data for input and output can be activated simultaneously by appropriate predefined sentences, "Predefined PDO".

On the "Process data" tab, the EL2595 has



Fig. 140: "Process data" tab

the following 'predefined PDO' sentences:

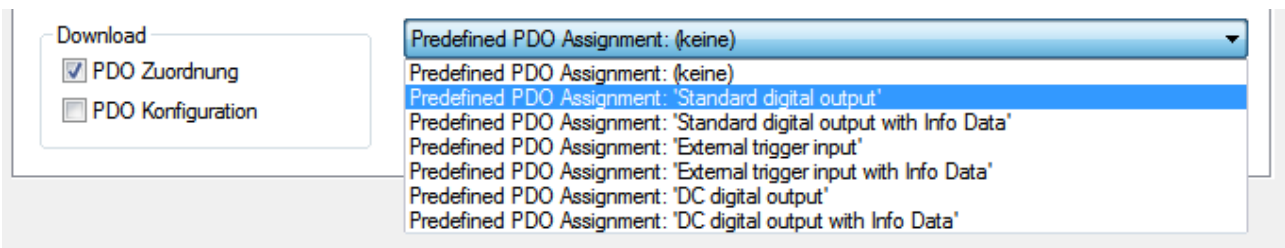


Fig. 141: TwinCAT System Manager with the PDO selection

In detail the sentences are composed as follows:

Operation mode	Name	SM2, PDO assignment	SM3, PDO assignment
SM-synchronous "frame-triggered"	Standard digital output (default setting)	0x1600	0x1A00
	Standard digital output with InfoData	0x1600	0x1A00 0x1A01
	External trigger input	0x1600 0x1602 0x1603	0x1A00
	External trigger input with InfoData	0x1600 0x1602 0x1603	0x1A00 0x1A01
DistributedClocks- triggered	DC Digital output	0x1600 0x1602 0x1604	0x1A00 0x1A02
	DC Digital output with InfoData	0x1600 0x1602 0x1604	0x1A00 0x1A01 0x1A02

5.6.2 Explanation of the process data

Standard digital output (default)

The outputs can be written directly with this standard PDO assignment, i.e. the connected actuators can be switched frame-triggered directly.

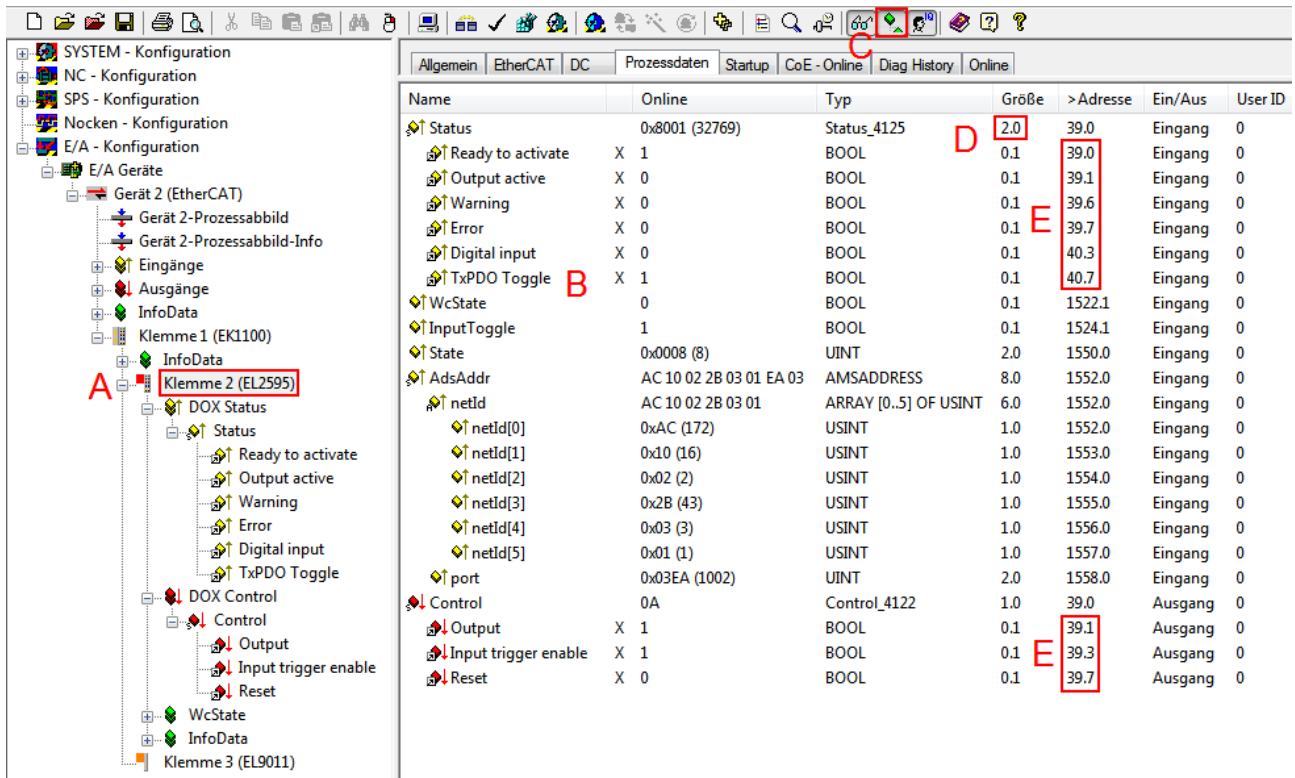


Fig. 142: Standard EL2595 process image

The EL2595 (A) has 2-byte variables at its disposal with different bit meanings. These can be seen by expanding the tree (A). They are also displayed in the detail view (B) if the appropriate display function (C) is activated.

The bit meaning i.e. offset position can then also be taken from the memory assignment display (E), taking into account the variable size (D).

Both the collective name e.g. *Status* and the individual bit variable e.g. *"Output active"* can be linked, but not both at the same time.

Input data			
Collective name	Name	Description / function	Bit position [0..15]
Status	Ready to activate	The terminal signals its readiness for operation here.	0
	Output active	The output is actively switched.	1
	Warning	A warning has occurred - > evaluate "Diag data" (index 0xA000).	6
	Error	An error has occurred and the output driver is deactivated - > evaluate "Diag data" (index 0xA000).	7
	Digital input	The status of the digital input is returned here.	11
	TxPDO Toggle	Changes its state each time process data are exchanged.	15
WcState		Setpoint during operation: 0 Each datagram of the EL2595 indicates its processing state here. This allows the EL2595 to be monitored for correct process data communication.	
InfoData (State)		Setpoint during operation: 8 Status display of the "EtherCAT state machine"	
AdsAddr		AMS address of the responsible EtherCAT Master in the format "0.0.0.0.0". In addition, the port number valid for this Slave. Required for acyclic accesses to the CoE at runtime.	

Output data			
Collective name	Name	Description / function	Bit position [0..15]
Control	Output	Switches the output active	1
	Input trigger enable	"Arms" the digital input as a trigger input	3
	Reset	Resets an error	7

Standard digital output with InfoData

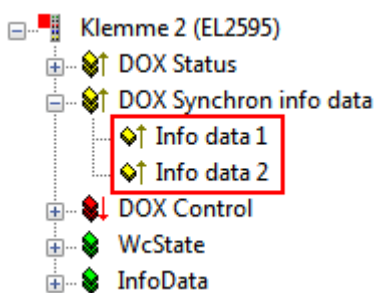


Fig. 143: Additional information data

Two further cyclic data words can be displayed per channel for more exact information about the states of the actuators or the driver stage. The respective selection is to be configured via the corresponding Index [0x8002:11 \[▶ 133\]](#) or [0x8002:19 \[▶ 133\]](#) in the CoE. Among other things, the interior temperature of the terminal or the momentary current through the connected actuator can be selected, for example.

Input data	
Name	Description / function
Info data 1	Additional channel information, definition in 0x8002:11 [▶ 133]
Info data 2	Additional channel information, definition in 0x8002:19 [▶ 133]

External trigger input

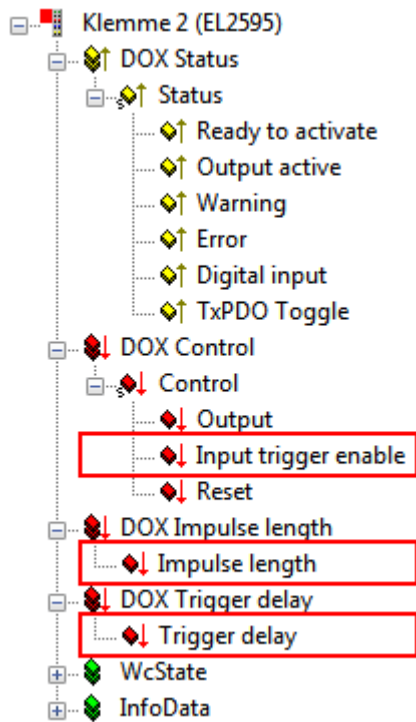


Fig. 144: Additional process data in the "External trigger input" mode

In addition to the variables of the "Standard digital output" mode, there are also the following variables:

Output data	
Name	Description / function
Input trigger enable	"Arms" the digital input as a trigger input
Impulse length	Specifies the length of the output pulses. The unit is 1 μ s. The data type UDINT (double word) provides the user with maximum flexibility and can specify pulses from 1 μ s to approx. 1 hr with a high resolution.
Trigger delay	Specifies the length of the delay from the time of the digital input to the switching of the output. The unit here is similarly 1 μ s.

External trigger input with InfoData

Like both standard data types, additional information data can also be displayed in "External trigger input" mode. See above.

DC Digital output

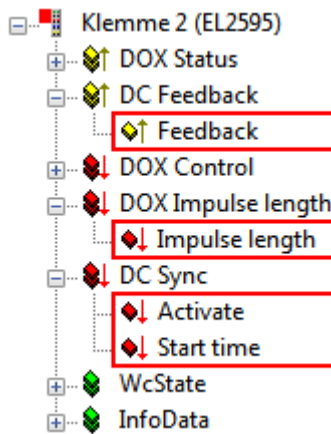


Fig. 145: Additional process data for Distributed Clocks mode

In Distributed Clocks mode the EL2595 works according to timestamp order. The process image is structured correspondingly, as in the fig. *Additional process data for Distributed Clocks mode.*

Input data	
Name	Description / function
Feedback	The terminal returns the current state of the output channels in this byte. This can be used to check a switching order. Bit 0: Output channel 1

Output data	
Name	Description / function
Activate	This byte activates a new start time in the terminal through the transition 0 --> 3. The sequence: 1. in an EtherCAT cycle <i>Activate</i> = 0, transmit the target state of the two output channels and desired start time to the EL2595 2. activate the switching order in the following EtherCAT cycle with <i>Activate</i> = 3
Start time	64-bit value of the next desired switching event. The data of the DC time: • Start time 1.1.2000 • Resolution 1 bit = 1 ns

● Switching time

i The desired switching time must be far "enough" in the future, as seen from the time of calculation, in the NC/PLC in order to be capable of being conveyed to the terminal, including activation. Since 2 EtherCAT cycles are required for the activation, it is recommended not to make this interval smaller than 3 cycles.

DC Digital output with InfoData

Like both standard data types, additional information data can also be displayed in DC mode. See above.

5.6.3 Process data overview

Manual process data assignment is necessary for TwinCAT up to version 2.10.

Sync Manager (SM)

The extent of the process data that is made available can be changed via the "Process data" tab (see Fig. *Process data tab SM2, EL2595 (default)*).

The PDOs from the range 0x160n (0x1600 to 0x1604) can be assigned to the Output SyncManager 2, see fig. *Process data tab SM2, EL2595 (default)*.

The PDOs from the range 0x1A0n (0x1A00 to 0x1A02) can be assigned to the Input SyncManager 3. See fig. *Process Data tab SM3, EL2595 (default)*.

Not all combinations are technically useful.

The screenshot shows the 'Process Data' configuration window for Sync Manager 2. The top navigation bar includes 'Allgemein', 'EtherCAT', 'DC', 'Prozessdaten', 'Startup', 'CoE - Online', 'Diag History', and 'Online'. The 'Prozessdaten' tab is active.

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	1	Outputs	
3	2	Inputs	

PDO Liste:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F	3	0
0x1A01	4.0	DOX Synchron info data	F		0
0x1A02	2.0	DC Feedback	F		0
0x1600	2.0	DOX Control	F	2	0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	10.0	DC Sync	F		0

PDO Zuordnung (0x1C12):

- 0x1600
- 0x1601
- 0x1602
- 0x1603
- 0x1604

PDO Inhalt (0x1600):

Index	Size	Offs	Name	Type	Default (hex)
--	0.1	0.0	--		
0x7000:02	0.1	0.1	Control__Output	BOOL	
--	0.1	0.2	--		
0x7000:04	0.1	0.3	Control__Input trigger enable	BOOL	
--	0.3	0.4	--		
0x7000:08	0.1	0.7	Control__Reset	BOOL	
--	1.0	1.0	--		
		2.0			

Download

- PDO Zuordnung
- PDO Konfiguration

Predefined PDO Assignment: 'Standard digital output'

Lade PDO Info aus dem Gerät

Sync Unit Zuordnung...

Fig. 146: Process Data tab SM2, EL2595 (default)

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	1	Outputs	
3	2	Inputs	

PDO Liste:

Index	Size	Name	Flags	SM	SU
0x1A00	2.0	DOX Status	F	3	0
0x1A01	4.0	DOX Synchron info data	F		0
0x1A02	2.0	DC Feedback	F		0
0x1600	2.0	DOX Control	F	2	0
0x1601	2.0	DOX Current	F		0
0x1602	4.0	DOX Impulse length	F		0
0x1603	4.0	DOX Trigger delay	F		0
0x1604	10.0	DC Sync	F		0

PDO Zuordnung (0x1C13):

- 0x1A00
- 0x1A01
- 0x1A02

PDO Inhalt (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x6000:01	0.1	0.0	Status__Ready to activate	BOOL	
0x6000:02	0.1	0.1	Status__Output active	BOOL	
---	0.4	0.2	---		
0x6000:07	0.1	0.6	Status__Warning	BOOL	
0x6000:08	0.1	0.7	Status__Error	BOOL	
---	0.3	1.0	---		
0x6000:0C	0.1	1.3	Status__Digital input	BOOL	
---	0.3	1.4	---		
0x6000:10	0.1	1.7	Status__TxPDO Toggle	BOOL	
		2.0			

Download

PDO Zuordnung

PDO Konfiguration

Predefined PDO Assignment: 'Standard digital output'

Lade PDO Info aus dem Gerät

Sync Unit Zuordnung...

Fig. 147: Process Data tab SM3, EL2595 (default)

Manual PDO Assignment

To configure the process data, select the required Sync Manager (SM 2 + 3) in the "Sync Manager" field at the top left (see Fig. *Process data tab SM3, EL2595 (default)*). The process data assigned to this Sync Manager can then be switched on or off in the "PDO Assignment" box underneath. Restarting the EtherCAT system, or reloading the configuration in Config mode (F4), causes the EtherCAT communication to restart, and the process data is transferred from the terminal.

SM2, PDO Assignment 0x1C12				
Index	Index of excluded PDOs	Size (byte.bit)	Name	PDO Content
0x1600 (default)	-	2.0	DOX Control	Index 0x7000:02 [▶ 131] - Output Index 0x7000:04 [▶ 131] - Input trigger active Index 0x7000:08 [▶ 131] - Reset
0x1601	-	2.0	DOX Current	Index 0x7000:11 [▶ 131]- Output current
0x1602	-	4.0	DOX Impulse length	Index 0x7000:12 [▶ 131] - Impulse length
0x1603	-	4.0	DOX Trigger delay	Index 0x7000:13 [▶ 131] - Trigger delay
0x1604	-	10.0	DC Sync possible exclusively in conjunction with DC operation mode!	Index 0xF700:01 [▶ 135] - Activate Index 0xF700:03 [▶ 135] - Start time

SM3, PDO Assignment 0x1C13				
Index	Index of excluded PDOs	Size (byte.bit)	Name	PDO Content
0x1A00 (default)	-	2.0	DOX Status	Index 0x6000:01 [▶ 131] - Ready to activate Index 0x6000:02 [▶ 131] - Output active Index 0x6000:07 [▶ 131] - Warning Index 0x6000:08 [▶ 131] - Error Index 0x6000:0C [▶ 131] - Digital input Index 0x6000:10 [▶ 131] - TxPDO Toggle
0x1A01	-	4.0	DOX Synchron info data	Index 0x6000:11 [▶ 131] - Info data 1 Index 0x6000:12 [▶ 131] - Info data 2
0x1A02	-	2.0	DC Feedback possible exclusively in conjunction with DC operation mode!	Index 0xF600:01 [▶ 134] - Feedback

5.7 Distributed Clocks settings

5.7.1 Basic principles

The EL2595 has a special feature in DC mode: usually the DC unit in the ESC (EtherCAT Slave Controller) is managed by the EtherCAT master. In the case of the EL2595, however, the local controller manages the start time and sets appropriate values in the ESC.

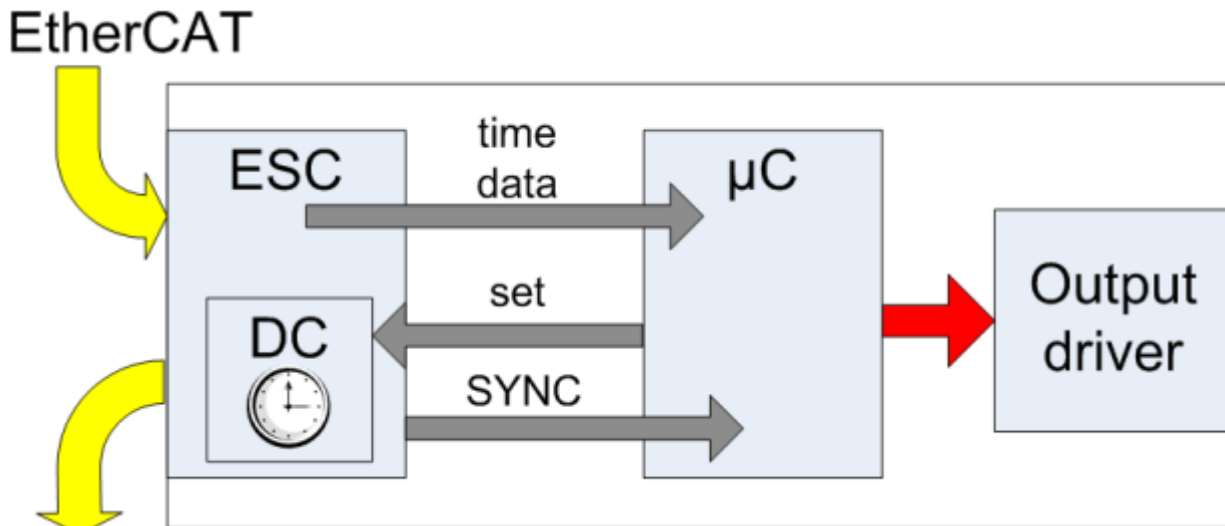


Fig. 148: EL2595 DC schematic

The controller (B) receives timestamps, activation and target states as normal process data from the ESC (A). It parameterizes the ESC to the next start time. At the set time the ESC informs the controller of this via the SYNC signal and the controller then activates the output stages accordingly.

For this reason the corresponding DC entries cannot be changed in the TwinCAT System Manager despite DC mode, but the EL2595 can nevertheless be used as a ReferenceClock.

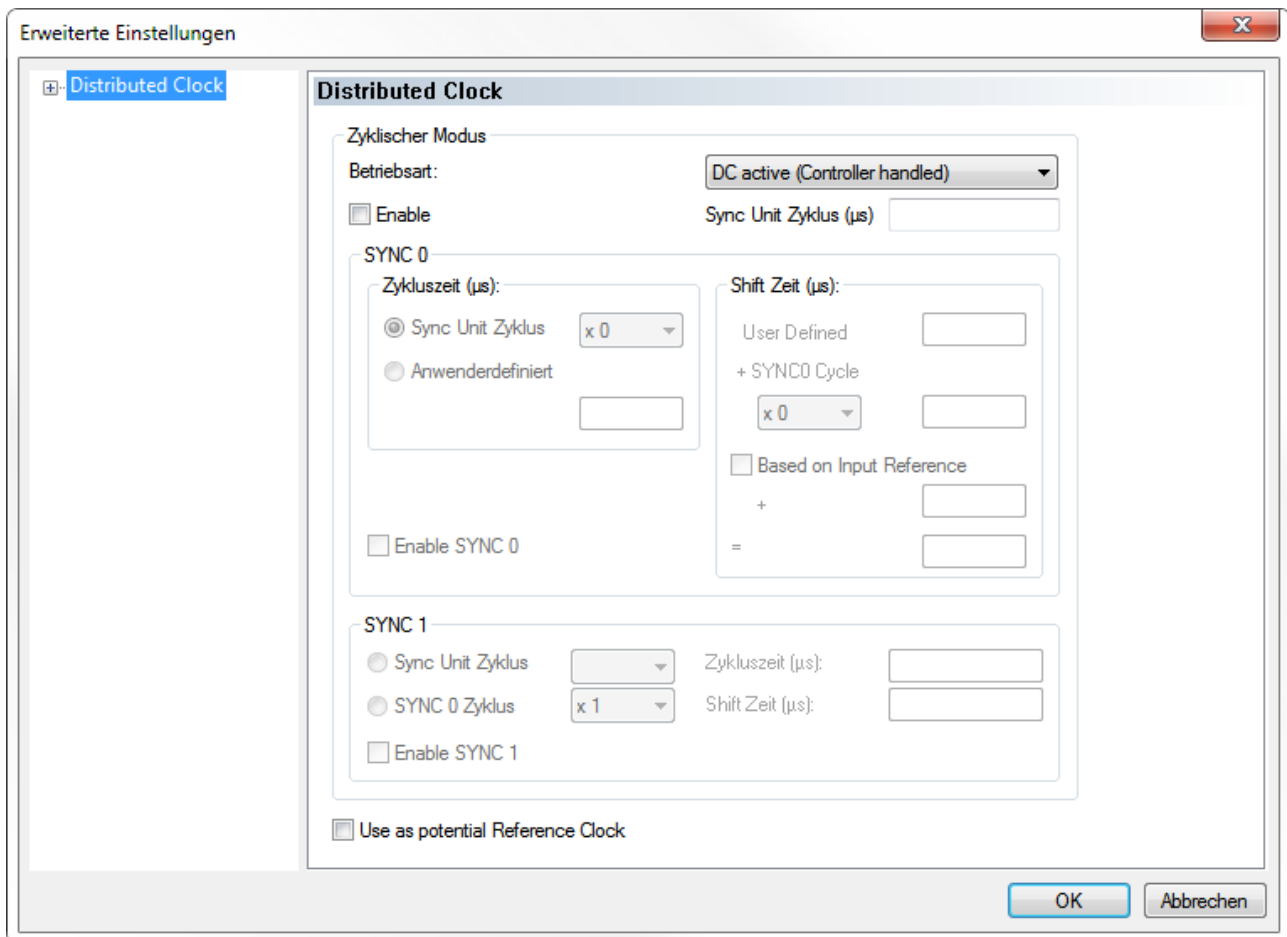


Fig. 149: EL2595 DC settings

5.7.2 Determination of the current Distributed Clock time

If an up-to-date statement of the Distributed Clock time is needed in a PLC cycle, this can be linked via the input variable of the EtherCAT master.

● Functions for data types with 64-bit width

i A selection of functions for handling 64-bit numbers is available under Beckhoff TwinCAT in the TcUtilities.lib library. Longer execution times are required here than is the case with standard, 32-bit data types. A data type with a width of 64-bit is defined in TcEthercat.lib as T_DCTIME or in TcUtilities.lib as T_LARGE_INTEGER.

The EtherCAT master can display a copy of the current master distributed clock. To this end activate the setting "Show DC System Time (64-bit)" in the *EtherCAT device --> EtherCAT tab --> Advanced Settings --> Distributed Clocks* (Fig. *Activation of the master distributed clock display*).

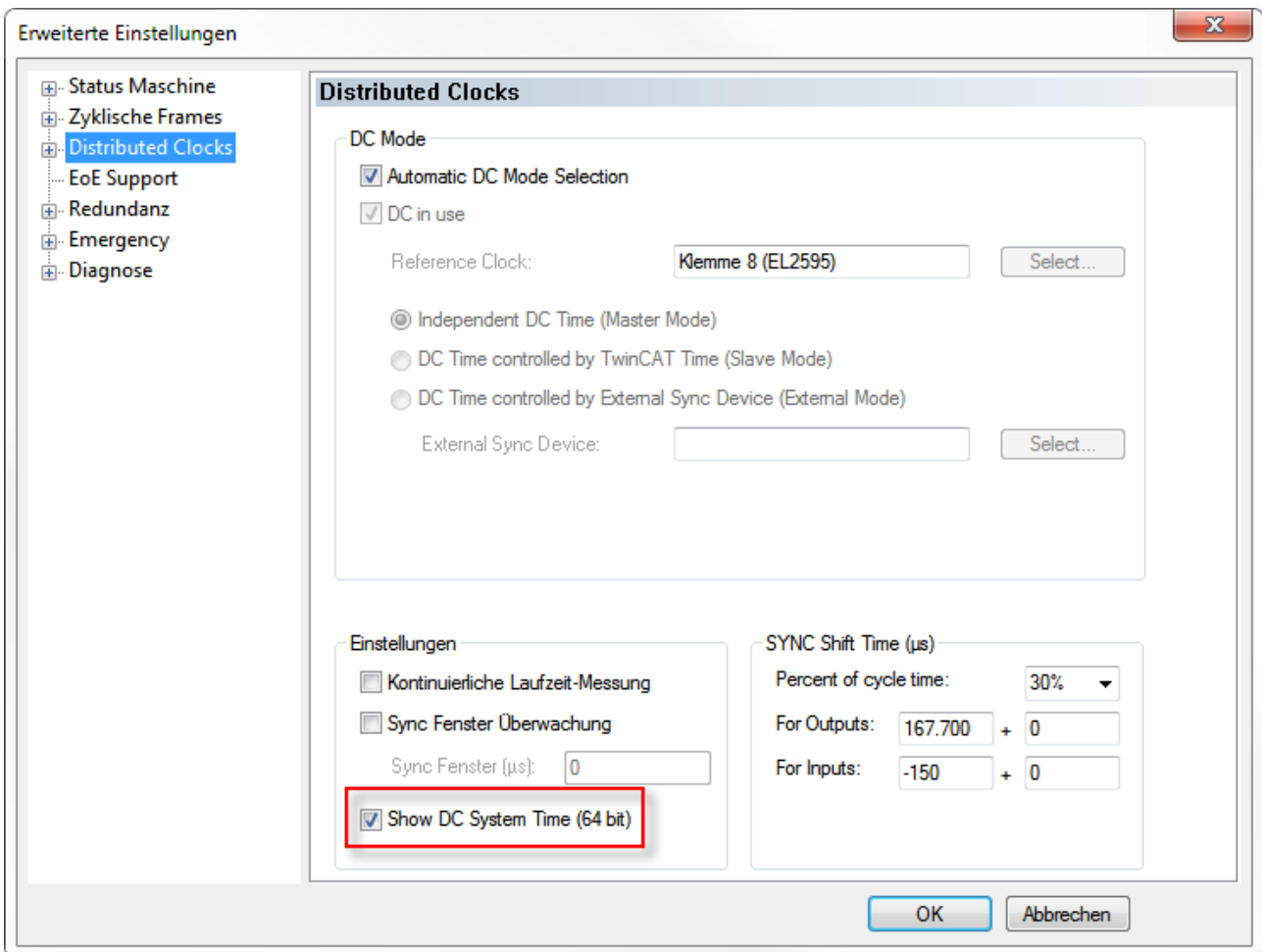


Fig. 150: Activation of the master distributed clock display

The process image of the EtherCAT master now looks as shown in Fig. *Extended process image of the EtherCAT master*:

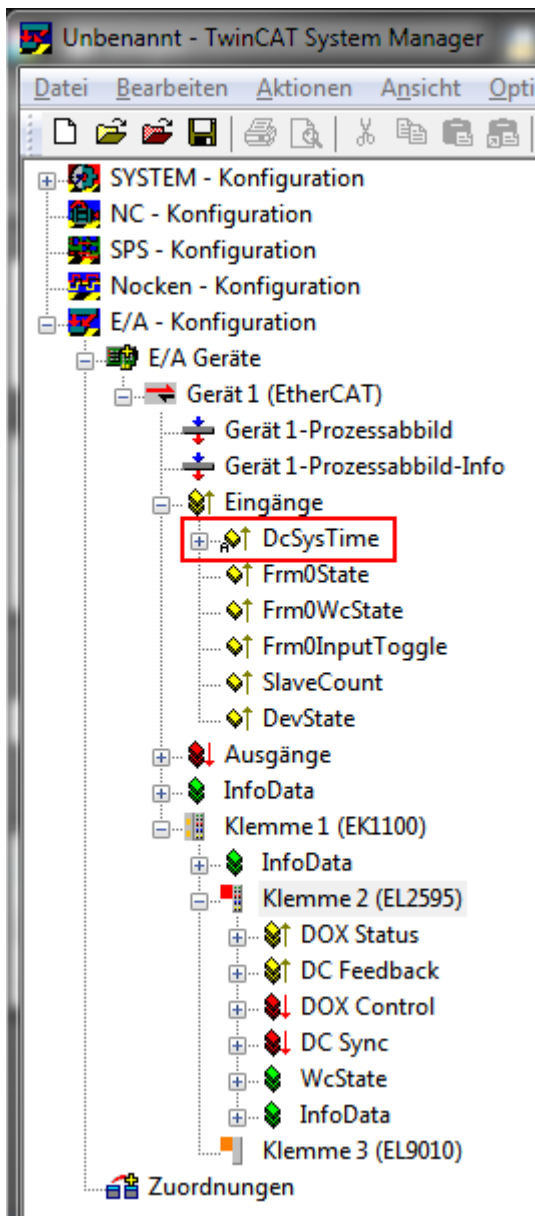


Fig. 151: Extended process image of the EtherCAT master

i SYSTIME

The EtherCAT master value DcSysTime is to be used with care and should serve only as a coarse indication as to which time range (order of magnitude: 1-2 task cycles) the Distributed Clocks system is currently in.

This is due to the fact that the local time in the terminal is read into associated buffers several μ s before retrieving frame. The EtherCAT strand then requires some time for transporting the process data to the master. Due to the character of the visualization, the online display in the TwinCAT System Manager is accurate at best to a three-figure millisecond range. Within a PLC the SysTime process data may also be outdated to a certain degree: depending on when the analysis takes place in the program code, the process data may be several μ s/ms older. If responses of an EtherCAT segment (e.g. an EL2212 output terminal) are to be based on a SysTime read in this way, 2-3 PLC cycles should be taken into account as a buffer.

It is more appropriate to use an EL2595 with other EtherCAT slaves that generate a time stamp based on ambient influences, such as the EL1252.

Alternatively, functions can also be called at NC/PLC runtime that immediately return the current DC time, e.g. F_GetCurDcTickTime. Refer to the notes on the TwinCAT time sources in the EtherCAT system documentation.

5.7.3 Process data

For the operation of the EL2595 the activation of the process data [▶ 116] provided for this is necessary.

5.7.4 Example of a sequence with activated Distributed Clocks

The following sequence is to be adhered to when using the EL2595 with activated *Distributed Clocks*:

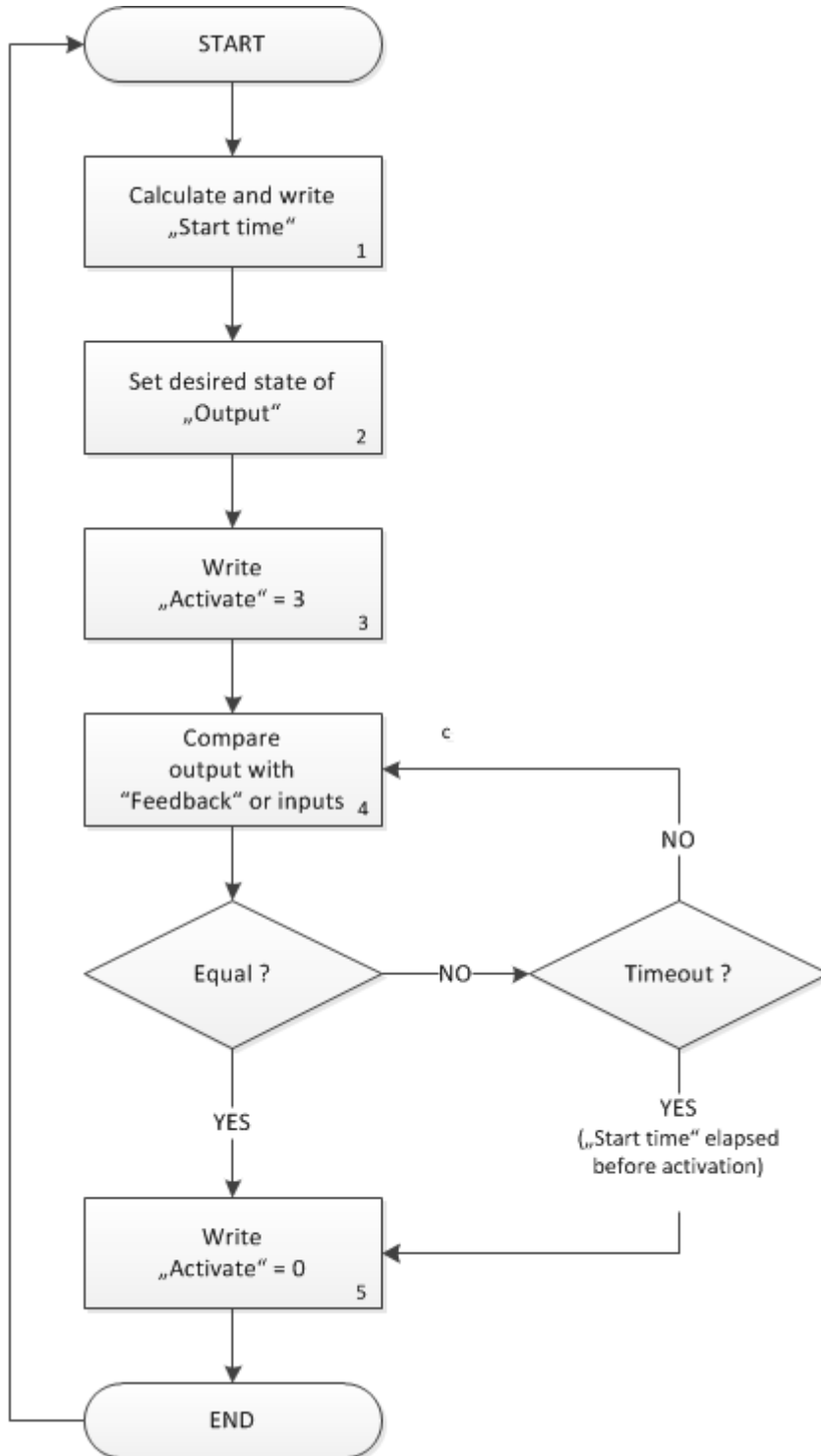


Fig. 152: Sequence of a switching operation with activated DistributedClock

1. Calculation of the output time:

- Determination of the current Distributed Clock time in 64-bit format, e.g. through the activated DC System Time of the EtherCAT master
 - Calculation and writing of the desired output time Start time with 64-bit operations.
2. Setting the desired output state of the digital signal:
 - the Output output data
 - at the same time Activate must be 0
 3. Transmission of these output data to the EL2595 by the EtherCAT cycle
 4. In the subsequent cycle: activation of the output time:
 - Set Activate to 3 (transition from 0 to 3 activates the start time)
 5. The EL2595 now waits as described above until the start time occurs and then send the output data to the power stage
 - until the switching event occurs, Activate and the Output output bit must be maintained in accordance with the target state
 - in the event of a power loss of the terminal being switched off, the start time is lost and must be reset by the control system
 6. Optional: Comparison of the input and output data:
 - Compare the feedback byte with the desired output image in order to register the switching operation
 - if necessary, check whether the start time has already elapsed, if the writing of the start time took place too time-critically
 7. Deactivation of the start time:
 - Set Activate to 0 in order to prepare for the subsequent activation phase

For each EtherCAT cycle only one switching event can be defined.

It is permissible to carry out the writing of the output data target output states, start time and Activate = 0 in the same EtherCAT cycle.

5.8 Object description and parameterization

● EtherCAT XML Device Description

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

● Parameterization via the CoE list (CAN over EtherCAT)

i The EtherCAT device is parameterized via the CoE-Online tab [► 90] (double-click on the respective object) or via the Process Data tab [► 87](allocation of PDOs). Please note the following general CoE notes [► 19] when using/manipulating the CoE parameters:

- Keep a startup list if components have to be replaced
- Differentiation between online/offline dictionary, existence of current XML description
- use “CoE reload” for resetting changes

5.8.1 Profile-specific objects

Index 6000 DOX Inputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
6000:0	DOX Inputs Ch.1	Max. Subindex	UINT8	RO	0x12 (18 _{dec})
6000:01	Ready to activate	Driver stage is ready for activation	BOOLEAN	RO	0x00 (0 _{dec})
6000:02	Output active	Output is activated	BOOLEAN	RO	0x00 (0 _{dec})
6000:07	Warning	a warning has occurred (see index App0)	BOOLEAN	RO	0x00 (0 _{dec})
6000:08	Error	An error has occurred (see index App0)	BOOLEAN	RO	0x00 (0 _{dec})
6000:0C	Digital input	Digital input	BOOLEAN	RO	0x00 (0 _{dec})
6000:10	TxPDO Toggle	Toggle bit	BOOLEAN	RO	0x00 (0 _{dec})
6000:11	Info data 1	Synchronous information (selection via subindex 0x8001:11)	UINT16	RO	0x0000 (0 _{dec})
6000:12	Info data 2	Synchronous information (selection via subindex 0x8001:19)	UINT16	RO	0x0000 (0 _{dec})

Index 7000 DOX Outputs Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
7000:0	DOX Outputs Ch.1	Max. Subindex	UINT8	RO	0x13 (19 _{dec})
7000:02	Output	activates the output stage	BOOLEAN	RO	0x00 (0 _{dec})
7000:04	Input trigger enable	activates the trigger function of the digital input	BOOLEAN	RO	0x00 (0 _{dec})
7000:08	Reset	all errors that may have occurred are reset by setting this bit (rising edge)	BOOLEAN	RO	0x00 (0 _{dec})
7000:11	Output current	variable specification of the output current (unit: 1 mA)	UINT16	RO	0x0000 (0 _{dec})
7000:12	Impulse length	variable specification of the output pulse (unit: 1 µs)	UINT32	RO	0x00000000 (0 _{dec})
7000:13	Trigger delay	variable specification of the delay of the trigger input (unit: 1 µs)	UINT32	RO	0x00000000 (0 _{dec})

Index 8000 DOX Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
8000:0	DOX Settings Ch.1	Max. Subindex	UINT8	RO	0x14 (20 _{dec})
8000:02	Hold current	Nominal current (unit: 1 mA)	UINT16	RW	0x01F4 (500 _{dec})
8000:03	Supply voltage	Nominal voltage (supply voltage) (unit: 0.01 V)	UINT16	RW	0x0960 (2400 _{dec})
8000:04	Output voltage	Nominal output voltage (load voltage) (unit: 0.01 V)	UINT16	RW	0x03E8 (1000 _{dec})
8000:09	Pulse on time (max.)	maximum length of the output pulse (unit: 1 μs)	UINT32	RW	0x00000000 (0 _{dec})
8000:0A	Pulse off time (min.)	minimum length of the interpulse period (unit: 1 μs)	UINT32	RW	0x00000000 (0 _{dec})
8000:0B	Trigger delay (switch on)	Specification of the delay of the trigger input (unit: 1 μs)	UINT32	RW	0x00000000 (0 _{dec})
8000:11	Warning level (supply voltage)	Warning threshold of the supply voltage (unit: 1%)	UINT8	RW	0x05 (5 _{dec})
8000:12	Error level (supply voltage)	Error threshold of the supply voltage (unit: 1%)	UINT8	RW	0x14 (20 _{dec})
8000:13	Warning level (output voltage)	Warning threshold of the output voltage (unit: 1%)	UINT8	RW	0x05 (5 _{dec})
8000:14	Error level (output voltage)	Error threshold of the output voltage (unit: 1%)	UINT8	RW	0x14 (20 _{dec})

Index 8001 DOX Controller Settings Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
8001:0	DOX Controller Settings Ch.1	Max. Subindex	UINT8	RO	0x03 (3 _{dec})
8001:01	Kp factor (curr.)	Kp control factor of the current controller	UINT16	RW	0x03E8 (1000 _{dec})
8001:02	Ki factor (curr.)	Ki control factor of the current controller	UINT16	RW	0x000A (10 _{dec})
8001:03	Kd factor (curr.)	Kd control factor of the current controller	UINT16	RW	0x0000 (0 _{dec})

Index 8002 DOX Features Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default	
8002:0	DOX Features Ch.1	Max. Subindex	UINT8	RO	0x32 (50 _{dec})	
8002:04	Enable emergency flash	Activation of the emergency flash routine	BOOLEAN	RW	0x00 (0 _{dec})	
8002:05	Enable automatic re-set on error	automatic clearing of an error activated	BOOLEAN	RW	0x01 (1 _{dec})	
8002:06	Enable automatic saving of counter	automatic saving of the internal counters activated (operating hours, switching cycles)	BOOLEAN	RW	0x01 (1 _{dec})	
8002:11	Select info data 1	Selection "Info data 1":	UINT8	RW	0x07 (7 _{dec})	
		0				Status word
		7				Output voltage
		8				Output current
		101				Internal temperature
		104				Supply voltage
		105				Supply current
8002:19	Select info data 2	Selection "Info data 2":	UINT8	RW	0x08 (8 _{dec})	
		0				Status word
		7				Output voltage
		8				Output current
		101				Internal temperature
		104				Supply voltage
		105				Supply current
8002:30	Invert digital input	Activates the inversion of the digital input	BOOLEAN	RW	0x00 (0 _{dec})	
8002:32	Function for input	Function selection of the digital input:	BIT4	RW	0x00 (0 _{dec})	
		0				Normal Input
		1				Hardware enable
		2				Trigger Input
		3				Trigger Input (Watchdog disabled)

Index 8003 DOX Flash timing Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
8003:0	DOX Flash timing Ch.1	Max. Subindex	UINT8	RO	0x0C (12 _{dec})
8003:01	ON-Time 1	Length of the ON-Time 1 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:02	OFF-Time 1	Length of the OFF-Time 1 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:03	ON-Time 2	Length of the ON-Time 2 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:04	OFF-Time 2	Length of the OFF-Time 2 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:05	ON-Time 3	Length of the ON-Time 3 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:06	OFF-Time 3	Length of the OFF-Time 3 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:07	ON-Time 4	Length of the ON-Time 4 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:08	OFF-Time 4	Length of the OFF-Time 4 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:09	ON-Time 5	Length of the ON-Time 5 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:0A	OFF-Time 5	Length of the OFF-Time 5 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:0B	ON-Time 6	Length of the ON-Time 6 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})
8003:0C	OFF-Time 6	Length of the OFF-Time 6 (unit: 1 µs)	UINT16	RW	0x0000 (0 _{dec})

Index 800F DOX Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
800F:0	DOX Vendor data	Max. Subindex	UINT8	RO	0x04 (4 _{dec})
800F:01	Offset (output voltage)	Vendor - offset calibration of the output voltage	INT16	RW	0x0000 (0 _{dec})
800F:02	Gain (output voltage)	Vendor - gain calibration of the output voltage	UINT16	RW	0x4000 (16384 _{dec})
800F:03	Offset (output current)	Vendor - offset calibration of the output current	INT16	RW	0x0000 (0 _{dec})
800F:04	Gain (output current)	Vendor - gain calibration of the output current	UINT16	RW	0x4000 (16384 _{dec})

Index 9000 DOX Info data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
9000:0	DOX Info data Ch.1	Max. Subindex	UINT8	RO	0x12 (18 _{dec})
9000:01	Status word	Status word (see Index App0)	UINT16	RO	0x0000 (0 _{dec})
9000:08	Output voltage	current output voltage (internal measuring point, unit: 0.01 V)	UINT16	RO	0x0000 (0 _{dec})
9000:09	Output current	current output current (internal measuring point, unit: 1 mA)	UINT16	RO	0x0000 (0 _{dec})
9000:11	Operating hour counter	Operating hour counter for the connected load (unit: 1 minute)	UINT32	RO	0x00000000 (0 _{dec})
9000:12	Switching operation counter	Switching cycle counter for the connected load	UINT32	RO	0x00000000 (0 _{dec})

Index A000 DOX Diag data Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
A000:0	DOX Diag data Ch.1	Max. Subindex	UINT8	RO	0x0B (11 _{dec})
A000:01	Saturated	Driver stage operates with maximum duty cycle	BOOLEAN	RO	0x00 (0 _{dec})
A000:02	Over temperature	Internal terminal temperature is greater than 80 °C	BOOLEAN	RO	0x00 (0 _{dec})
A000:04	Under voltage (Supply)	Supply voltage too low	BOOLEAN	RO	0x00 (0 _{dec})
A000:05	Over voltage (Supply)	Supply voltage too high	BOOLEAN	RO	0x00 (0 _{dec})
A000:06	Short circuit	Short circuit / overcurrent in the driver stage	BOOLEAN	RO	0x00 (0 _{dec})
A000:08	No control power	No power supply to driver stage	BOOLEAN	RO	0x00 (0 _{dec})
A000:09	Misc error	Internal temperature of the terminal is higher than 100 °C or short-circuit of the driver stage or supply voltage outside the permissible range or output voltage outside the permissible range	BOOLEAN	RO	0x00 (0 _{dec})
A000:0A	Under voltage (Output)	Output voltage too low	BOOLEAN	RO	0x00 (0 _{dec})
A000:0B	Over voltage (Output)	Output voltage too high	BOOLEAN	RO	0x00 (0 _{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2 _{dec})
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16 _{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0001 (1 _{dec})

Index F008 Code word

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

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Max. Subindex	UINT8	RW	0x01 (1 _{dec})
F010:01	SubIndex 001	Profile 210	UINT32	RW	0x000000D2 (210 _{dec})

Index F600 DC Inputs

Index (hex)	Name	Meaning	Data type	Flags	Default
F600:0	DC Inputs	Max. Subindex	UINT8	RO	0x01 (1 _{dec})
F600:01	Feedback	fed back output value (bit 0: Index 0x6000:02)	UINT8	RO	0x00 (0 _{dec})

Index F700 DC Outputs

Index (hex)	Name	Meaning	Data type	Flags	Default
F700:0	DC Outputs	Max. Subindex	UINT8	RO	0x03 (3 _{dec})
F700:01	Activate	This byte activates a new start time in the terminal through the transition 0 --> 3	UINT8	RO	0x00 (0 _{dec})
F700:03	Start time	Start time of the next desired switching event	UINT64	RO	0x0000000000 0000000 (0 _{dec})

Index F80F DOX Vendor data

Index (hex)	Name	Meaning	Data type	Flags	Default
F80F:0	DOX Vendor data	Max. Subindex	UINT8	RO	0x07 (7 _{dec})
F80F:01	PWM Frequency	DC link frequency (unit: 1 Hz)	UINT32	RW	0x0001D4C0 (120000 _{dec})
F80F:02	Deadtime	Dead time for pulse width modulation	UINT16	RW	0x0652 (1618 _{dec})
F80F:03	Deadtime space	Duty cycle limitation	UINT16	RW	0x0206 (518 _{dec})
F80F:04	Warning temperature	Threshold of the temperature warning (unit: 1 °C)	INT8	RW	0x50 (80 _{dec})
F80F:05	Switch off temperature	Switch-off temperature (unit: 1°C)	INT8	RW	0x64 (100 _{dec})
F80F:06	Analog trigger point	Trigger point for AD conversion	UINT16	RW	0x0032 (50 _{dec})
F80F:07	SYNC 0 shift time	Offset of the SYNC0 pulse	INT16	RW	0x0000 (0 _{dec})

Index F900 DOX Info data

Index (hex)	Name	Meaning	Data type	Flags	Default
F900:0	DOX Info data	Max. Subindex	UINT8	RO	0x06 (6 _{dec})
F900:01	Software version (driver)	Software version of the driver card	STRING	RO	
F900:02	Internal temperature	Internal terminal temperature (unit: 1°C)	INT8	RO	0x00 (0 _{dec})
F900:05	Supply voltage	Supply voltage (unit: 0.01 V)	UINT16	RO	0x0000 (0 _{dec})
F900:06	Supply current	Total current of the supply voltage (unit: 1 mA)	UINT16	RO	0x0000 (0 _{dec})

Index FB00 DOX Command

Index (hex)	Name	Meaning	Data type	Flags	Default	
FB00:0	DOX Command	Max. Subindex	UINT8	RO	0x03 (3 _{dec})	
FB00:01	Request	0x0001	Save operating hour counter manual saving of the operating hour counter	OCTET-STRING[2]	RW	{0}
		0x0002	Delete operating hour counter manual deletion of the operating hour counter			
		0x0103	Calibrate Carries out an internal calibration; the value of the current output voltage (Index 0x9000:08 [► 134]) is transferred into the configured output voltage (Index 0x8000:04 [► 132])			
		0x1000	Clear the Diag History Clear the Diag History			
		0x1100	Get build number Read out the build number			
		0x1101	Get build date Read out the build date			
		0x1102	Get build time Read out the build time			
		0x8000	Software reset Carry out a software reset (hardware is re-initialized with the current CoE configuration)			
FB00:02	Status	0	Finished, no error, no response Command terminated without error and without response	UINT8	RO	0x00 (0 _{dec})
		1	Finished, no error, response Command terminated without error and with response			
		2	Finished, error, no response Command terminated with error and without response			
		3	Finished, error, response Command terminated with error and with response			
		255	Executing Command is being executed			
FB00:03	Response	dependent on the request	OCTET STRING[4]	RO	{0}	

5.8.2 Standard objects

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default
1000:0	Device type	Device type of the EtherCAT slave: The Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x00D21389 (13767561 _{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL2595

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters [▶ 170]	Restore default parameters	UINT8	RO	0x01 (1 _{dec})
1011:01	SubIndex 001	If this object is set to "0x64616F6C" in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0 _{dec})

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4 _{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2 _{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x0A233052 (170078290 _{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the Low Word (bit 0-15) indicates the special terminal number, the High Word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0 _{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the Low Byte (bit 0-7) of the Low Word contains the year of production, the High Byte (bit 8-15) of the Low Word contains the week of production, the High Word (bit 16-31) is 0	UINT32	RO	0x00000000 (0 _{dec})

Index 10F0 Backup parameter handling

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

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Max. Subindex	UINT8	RO	0x37 (55 _{dec})
10F3:01	Maximum Messages	Maximum number of stored messages A maximum of 50 messages can be stored	UINT8	RO	0x00 (0 _{dec})
10F3:02	Newest Message	Subindex of the latest message	UINT8	RO	0x00 (0 _{dec})
10F3:03	Newest Acknowledged Message	Subindex of the last confirmed message	UINT8	RW	0x00 (0 _{dec})
10F3:04	New Messages Available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0 _{dec})
10F3:05	Flags	not used	UINT16	RW	0x0000 (0 _{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET STRING[28]	RO	{0}
...	OCTET STRING[28]	RO	{0}
10F3:37	Diagnosis Message 050	Message 50	OCTET STRING[28]	RO	{0}

Index 10F8 Actual Time Stamp

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

Index 1600 DOX RxPDO-Map Control

Index (hex)	Name	Meaning	Data type	Flags	Default
1600:0	DOX RxPDO-Map Control	PDO Mapping RxPDO 1	UINT8	RO	0x06 (6 _{dec})
1600:01	SubIndex 001	1. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 1
1600:02	SubIndex 002	2. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x02 (Output))	UINT32	RO	0x7000:02, 1
1600:03	SubIndex 003	3. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 1
1600:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x08 (Reset))	UINT32	RO	0x7000:04, 1
1600:05	SubIndex 005	5. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 3
1600:06	SubIndex 006	6. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x08 (Reset))	UINT32	RO	0x7000:08, 1

Index 1601 DOX RxPDO-Map Current

Index (hex)	Name	Meaning	Data type	Flags	Default
1601:0	DOX RxPDO-Map Current	PDO Mapping RxPDO 2	UINT8	RO	0x01 (1 _{dec})
1601:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x11 (Output current))	UINT32	RO	0x7000:11, 16

Index 1602 DOX RxPDO-Map Impulse length

Index (hex)	Name	Meaning	Data type	Flags	Default
1602:0	DOX RxPDO-Map Impulse length	PDO Mapping RxPDO 3	UINT8	RO	0x01 (1 _{dec})
1602:01	SubIndex 001	1. PDO Mapping entry (object 0xF700 (DC Outputs), entry 0x01 (Activate))	UINT32	RO	0x7000:12, 32

Index 1603 DOX RxPDO-Map Trigger delay

Index (hex)	Name	Meaning	Data type	Flags	Default
1603:0	DOX RxPDO-Map Trigger delay	PDO Mapping RxPDO 4	UINT8	RO	0x01 (1 _{dec})
1603:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (DOX Outputs Ch.1), entry 0x13 (Trigger delay))	UINT32	RO	0x7000:13, 32

Index 1604 DC RxPDO-Map Sync

Index (hex)	Name	Meaning	Data type	Flags	Default
1604:0	DC RxPDO-Map Sync	PDO Mapping RxPDO 5	UINT8	RO	0x03 (3 _{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0xF700 (DC Outputs), entry 0x01 (Activate))	UINT32	RO	0xF700:01, 8
1604:02	SubIndex 002	2. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1604:03	SubIndex 003	3. PDO Mapping entry (object 0xF700 (DC Outputs), entry 0x03 (Start time))	UINT32	RO	0xF700:03, 64

Index 1A00 DOX TxPDO-Map Status

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	DOX TxPDO-Map Status	PDO Mapping TxPDO 1	UINT8	RO	0x09 (9 _{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x01 (Ready to activate))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x02 (Output active))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x07 (Warning))	UINT32	RO	0x6000:07, 1
1A00:05	SubIndex 005	5. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x08 (Error))	UINT32	RO	0x6000:08, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 3
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x0E (Sync error))	UINT32	RO	0x6000:0C, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (1 bits align)	UINT32	RO	0x0000:00, 3
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x6000:10, 1

Index 1A01 DOX TxPDO-Map Synchron info data

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	DOX TxPDO-Map Synchron info data	PDO Mapping TxPDO 2	UINT8	RO	0x02 (2 _{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x11 (Info data 1))	UINT32	RO	0x6000:11, 16
1A01:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (DOX Inputs Ch.1), entry 0x12 (Info data 2))	UINT32	RO	0x6000:12, 16

Index 1A02 DC TxPDO-Map Feedback

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	DC TxPDO-Map Feedback	PDO Mapping TxPDO 3	UINT8	RO	0x02 (2 _{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0xF600 (DC Inputs), entry 0x01 (Feedback))	UINT32	RO	0xF600:01, 8
1A02:02	SubIndex 002	2. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x01 (1 _{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632 _{dec})
1C12:02	Subindex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0 _{dec})
1C12:03	Subindex 003	3. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0 _{dec})
1C12:04	Subindex 004	4. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0 _{dec})
1C12:05	Subindex 005	5. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x0000 (0 _{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x01 (1 _{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656 _{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0 _{dec})
1C13:03	Subindex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0 _{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x20 (32 _{dec})
1C32:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> 0: Free Run 1: Synchron with SM 2 Event 2: DC-Mode - Synchron with SYNC0 Event 3: DC-Mode - Synchron with SYNC1 Event 	UINT16	RW	0x0001 (1 _{dec})
1C32:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> Free Run: Cycle time of the local timer Synchron with SM 2 Event: Master cycle time DC mode: SYNC0/SYNC1 Cycle Time 	UINT32	RW	0x000F4240 (1000000 _{dec})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 _{dec})
1C32:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> Bit 0 = 1: free run is supported Bit 1 = 1: Synchron with SM 2 Event is supported Bit 2-3 = 01: DC mode is supported Bit 4-5 = 10: Output Shift with SYNC1 event (only DC mode) Bit 14 = 1: dynamic times (measurement through writing of 1C32:08) 	UINT16	RO	0x0003 (3 _{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x000186A0 (100000 _{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0 _{dec})
1C32:07	Minimum delay time		UINT32	RO	0x00000000 (0 _{dec})
1C32:08	Command	<ul style="list-style-type: none"> 0: Measurement of the local cycle time is stopped 1: Measurement of the local cycle time is started <p>The entries 0x1C32:03 [▶ 140], 0x1C32:05 [▶ 140], 0x1C32:06 [▶ 140], 0x1C32:09 [▶ 140], 0x1C33:03 [▶ 141], 0x1C33:06 [▶ 140], 0x1C33:09 [▶ 141] are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p>	UINT16	RW	0x0000 (0 _{dec})
1C32:09	Maximum delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0 _{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 _{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0 _{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0 _{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0 _{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32 _{dec})
1C33:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available) 	UINT16	RW	0x0022 (34 _{dec})
1C33:02	Cycle time	as 0x1C32:02 [▶ 140]	UINT32	RW	0x000F4240 (100000 _{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> • Bit 0: free run is supported • Bit 1: Synchron with SM 2 Event is supported (outputs available) • Bit 1: Synchron with SM 3 Event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: Input Shift through local event (outputs available) • Bit 4-5 = 10: Input Shift with SYNC1 Event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▶ 140] or 0x1C33:08 [▶ 141]) 	UINT16	RO	0x0003 (3 _{dec})
1C33:05	Minimum cycle time	as 0x1C32:05 [▶ 140]	UINT32	RO	0x000186A0 (100000 _{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000000 (0 _{dec})
1C33:08	Command	as 0x1C32:08 [▶ 140]	UINT16	RW	0x0000 (0 _{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:0B	SM event missed counter	as 0x1C32:11 [▶ 140]	UINT16	RO	0x0000 (0 _{dec})
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▶ 140]	UINT16	RO	0x0000 (0 _{dec})
1C33:0D	Shift too short counter	as 0x1C32:13 [▶ 140]	UINT16	RO	0x0000 (0 _{dec})
1C33:20	Sync error	as 0x1C32:32 [▶ 140]	BOOLEAN	RO	0x00 (0 _{dec})

5.9 CoE data

Configuration of the terminal via the CoE

The terminal must be configured via the CoE before the actuators can be put into operation.

Index	Name	Flags	Wert	Einheit
8000:0	DOX Settings Ch.1	RW	> 20 <	
8000:02	Hold current	RW	0x01F4 (500)	mA
8000:03	Supply voltage	RW	0x0960 (2400)	0,01 V
8000:04	Output voltage	RW	0x03E8 (1000)	0,01 V
8000:09	Pulse on time (max.)	RW	0x00000000 (0)	µs
8000:0A	Pulse off time (min.)	RW	0x00000000 (0)	µs
8000:0B	Trigger delay (switch on)	RW	0x00000000 (0)	µs
8000:11	Warning level (supply voltage)	RW	0x05 (5)	%
8000:12	Error level (supply voltage)	RW	0x14 (20)	%
8000:13	Warning level (output voltage)	RW	0x05 (5)	%
8000:14	Error level (output voltage)	RW	0x14 (20)	%
8001:0	DOX Controller Settings Ch.1	RW	> 3 <	
8001:01	Kp factor (curr.)	RW	0x03E8 (1000)	
8001:02	Ki factor (curr.)	RW	0x000A (10)	
8001:03	Kd factor (curr.)	RW	0x0000 (0)	
8002:0	DOX Features Ch.1	RW	> 50 <	
8002:04	Enable emergency flash	RW	FALSE	
8002:05	Enable automatic reset on error	RW	TRUE	
8002:06	Enable automatic saving of counter	RW	TRUE	
8002:11	Select info data 1	RW	Output voltage (7)	
8002:19	Select info data 2	RW	Output current (8)	
8002:30	Invert digital input	RW	FALSE	
8002:32	Function for input	RW	Normal Input (0)	
8003:0	DOX Flash timing Ch.1	RW	> 12 <	
8003:01	ON-Time 1	RW	0x0000 (0)	ms
8003:02	OFF-Time 1	RW	0x0000 (0)	ms
8003:03	ON-Time 2	RW	0x0000 (0)	ms
8003:04	OFF-Time 2	RW	0x0000 (0)	ms
8003:05	ON-Time 3	RW	0x0000 (0)	ms
8003:06	OFF-Time 3	RW	0x0000 (0)	ms
8003:07	ON-Time 4	RW	0x0000 (0)	ms
8003:08	OFF-Time 4	RW	0x0000 (0)	ms
8003:09	ON-Time 5	RW	0x0000 (0)	ms
8003:0A	OFF-Time 5	RW	0x0000 (0)	ms
8003:0B	ON-Time 6	RW	0x0000 (0)	ms
8003:0C	OFF-Time 6	RW	0x0000 (0)	ms

Fig. 153: CoE configuration data

DOX Settings

For a well-working and fast current controller, the expected **"Output voltage"** (Index 0x8000:04 |► 1321) must be precisely adjusted. This value is used as the initial value of the control. That means: the better the output voltage is adjusted, the less the current controller has to readjust and the smaller the voltage increase with the first output pulses. This initial value is used whenever the controller switches from INIT to OP, or if the terminal switches from an error state to the normal state.

The driver stage has the possibility to operate in Buck or Boost mode. Buck mode is active if the "Output voltage" is smaller than the **"Supply voltage"**. In the other case the Boost mode is used. There is a hysteresis area between these operation modes. This means that the controller does not switch to the other operation mode within the range of approx. +/- 1 V around the supply voltage. This prevents pulsing of the output voltage.

The consequence of this is that the operating point of the connected load must be observed when designing the supply voltage. The user must specify which nominal voltage of the load is required at the desired operating current – **"Hold current"**.

Example: Hold current = 500 mA, output voltage = 27 V -> the supply voltage may not lie within the range of 26 V to 28 V

"Pulse on time" and **"Pulse off time"** are intended for the protection of the connected load. Here, on the one hand, "Pulse on time" sets the maximum time of the output pulse for which the load may be operated with the specified "Hold current" without being destroyed. On the other, "Pulse off time" configures the minimum switch-off time for adjusting an average power loss in overload mode so that the load is not thermally destroyed.

The delay time of the digital input is parameterized with **"Trigger delay"**.

The **"Warning level"** and **"Error level"** offer the possibility to monitor the supply voltage and output voltage. The thresholds set here are proportionally subtracted from or added to the configured values.

Example: output voltage = 10 V, warning level = +/- 5%, error level = +/- 20% -> warning at $U_o < 9.5 \text{ V}$, $U_o > 10.5 \text{ V}$; error at $U_o < 8 \text{ V}$, $U_o > 12 \text{ V}$

● Setting of the threshold values when commissioning

i If the output is frequently deactivated during start-up due to an overvoltage or undervoltage error, the cause may be the incorrect configuration of the output voltage. In this case the error threshold can be set a little higher and the compensated voltage in Index [0x9000:08](#) [[▶ 134](#)] can be read out and Index [0x8000:04](#) [[▶ 132](#)] adapted if necessary.

DOX Controller Settings

The current controller is configured with this object.

● Control factors

i The control factors (Kp, Ki, Kd: Index [0x8001:01](#) [[▶ 132](#)]; [0x8001:02](#) [[▶ 132](#)]; [0x8001:03](#) [[▶ 132](#)]) should not normally be changed, since the default setting is adequate for most applications.

DOX Features

"Enable emergency flash" activates the output of a signal sequence in the case that the watchdog (loss of communication) of the terminal is triggered. The times are parameterized with the "DOX Flash timing" object.

If an error occurs it is not absolutely necessary to manually acknowledge it with the "Reset" bit in the PDO input data. By the activation of **"Enable automatic reset on error"** an existing error is automatically cleared on resetting "Output".

● Automatic clearing of the error

i Before resetting "Output" a diagnosis must take place and the indices [0x9000](#), [0xA000](#) and/or [0xF900](#) must be read out. The information is no longer available after the output has been reset!

If **"Enable automatic saving of counter"** is activated, the objects "Operating hour counter" and "Switching operation counter" are backed up to the internal memory at intervals of 15 minutes. However, this feature can also be deactivated and carried out manually at a defined time with the command object (Index [0xFB00](#)).

The selection of the values represented in the PDO "DOX InfoData" is carried out with **"Select info data n"**.

An inversion of the digital input is possible with **"Invert digital input"**.

Different functions of the digital input can be configured via **"Function for input"**. The following selections are possible:

- Normal input (0_{dec})
-> normal digital input without additional functions (the status is shown in the corresponding PDO)
- Hardware enable (1_{dec})
-> in addition to the "Enable emergency flash" bit, the input is used as an additional **static** hardware enable, i.e. the digital input must be active so that the emergency flash is switched
- Trigger input (2_{dec})
-> in addition to the "Output" bit, the input is used as an additional trigger input, i.e. the output is switched on a rising edge on the input (in the watchdog case the input trigger has no function)
- Trigger input (Watchdog disabled) (3_{dec})
-> same function as before but with the difference that the trigger input is still functional with the watchdog activated

DOX Flash timing

With Emergency flash activated the times of the signal sequence are specified here. With a resolution of 1 ms the user can define a total of 6 pulse outputs. If all 6 pulses are not required, the no longer used "ON-Time n" must be written to 0. If the desired pulse sequence has been processed, the terminal starts again at ON-Time 1.

Terminal diagnostics

The terminal offers the user various values in the CoE for diagnostics.

Index	Name	Flags	Wert	Einheit
9000:0	DOX Info data Ch.1	RO	> 18 <	
9000:01	Status word	RO	0x0004 (4)	
9000:08	Output voltage	RO	0x03E7 (999)	0,01 V
9000:09	Output current	RO	0x0000 (0)	mA
9000:11	Operating hour counter	RO	0x00000000 (0)	min
9000:12	Switching operation counter	RO	0x00000006 (6)	
A000:0	DOX Diag data Ch.1	RO	> 11 <	
A000:01	Saturated	RO	FALSE	
A000:02	Over temperature	RO	FALSE	
A000:04	Under voltage (Supply)	RO	FALSE	
A000:05	Over voltage (Supply)	RO	FALSE	
A000:06	Short circuit	RO	FALSE	
A000:08	No control power	RO	FALSE	
A000:09	Misc error	RO	FALSE	
A000:0A	Under voltage (Output)	RO	FALSE	
A000:0B	Over voltage (Output)	RO	FALSE	
F000:0	Modular device profile	RO	> 2 <	
F008	Code word	RW	0x00000000 (0)	
F010:0	Module list	RW	> 1 <	
F600:0	DC Inputs	RO	> 1 <	
F700:0	DC Outputs	RO	> 3 <	
F80F:0	DOX Vendor data	RW	> 7 <	
F900:0	DOX Info data	RO	> 6 <	
F900:01	Software version (driver)	RO	01	
F900:02	Internal temperature	RO	31	°C
F900:05	Supply voltage	RO	0x09B5 (2485)	0,01 V
F900:06	Supply current	RO	0x0139 (313)	mA
FB00:0	DOX Command	RO	> 3 <	

Fig. 154: CoE - diagnostic data

DOX Info data, DOX Diag data

[Index 0x9000 \[▶ 134\]](#) offers various channel-related measured values, such as output voltage or current. If a warning or error occurs, the user can obtain precise information about the cause of the fault from [Index 0xA000 \[▶ 134\]](#).

Terminal-specific information can be found in [Index 0xF900 \[▶ 135\]](#).

All 0x9000-series and 0xF900-series objects can be displayed in the process image via the "Synchron info data" and are to be configured via [index 0x8002:11 or 0x8002:19 \[▶ 133\]](#) respectively.

Meaning of the bits in "Diag data":

- Saturated: current controller output of 100% duty cycle
- Over temperature: the internal temperature of the terminal is higher than 80 °C (this value is configurable); if it exceeds 100 °C then Misc error is also set and the driver stage is deactivated.
- Under voltage (Supply): the measured supply voltage is x% lower than the configured supply voltage (see [index 0x8000:03 \[▶ 132\]](#) and [Index 0x8000:11 \[▶ 132\]](#) or [Index 0x8000:12 \[▶ 132\]](#) respectively)
- Over voltage (Supply): the measured supply voltage is x% higher than the configured supply voltage
- Short circuit: Short circuit / overcurrent in the driver stage
- No control power: no control voltage / supply voltage applied
- Misc error: general error display
- Under voltage (Output): the measured output voltage is x% lower than the configured output voltage (see [Index 0x8000:04 \[▶ 132\]](#) and [Index 0x8000:13 \[▶ 132\]](#) or [Index 0x8000:14 \[▶ 132\]](#) respectively)
- Over voltage (Output): the measured output voltage is x% higher than the configured output voltage

Command interface

The terminal offers the user a command interface for different services.

Index	Name	Flags	Wert	Einheit
[-] FB00:0	DOX Command	RO	> 3 <	
FB00:01	Request	RW	00 00	
FB00:02	Status	RO	0x00 (0)	
FB00:03	Response	RO	00 00 00 00	

Fig. 155: CoE command object

DOX Command

In addition to the general commands described [here \[▶ 136\]](#), there are three more specific commands for the EL2595 that are designed to aid the user with the operation of the terminal.

0x0001 (manual saving of the operating hour and switching cycle counter):

The two counters in [Index 0x9000:11 \[▶ 134\]](#) and [Index 0x9000:12 \[▶ 134\]](#) are saved acyclically with this command.

0x0002 (manual deletion of the operating hour and switching cycle counter):

The two counters in [Index 0x9000:11 \[▶ 134\]](#) and [Index 0x9000:12 \[▶ 134\]](#) are deleted with this command.

0x0103 (calibration of the CoE value of the output voltage):

In order to ensure optimum switching characteristics of the load, the configured output voltage should lie as close as possible to the real output voltage. Therefore the measured value from [Index 0x9000:08 \[▶ 134\]](#) should be transferred into the object [Index 0x8000:04 \[▶ 132\]](#) during commissioning. If the user issues this command to the terminal, the terminal carries out this process automatically.

i **Order of the data in the Command object**

The data in the Request and in the Response subindex are created as byte arrays. For the user this means that he must first carry out a byte swap with the commands/requests that he writes.

Example: Command: 0x0103 -> Request: 03 01

6 Diagnostics

6.1 Diagnostics – basic principles of diag messages

DiagMessages designates a system for the transmission of messages from the EtherCAT Slave to the EtherCAT Master/TwinCAT. The messages are stored by the device in its own CoE under 0x10F3 and can be read by the application or the System Manager. An error message referenced via a code is output for each event stored in the device (warning, error, status change).

Definition

The *DiagMessages* system is defined in the ETG (EtherCAT Technology Group) in the guideline ETG.1020, chapter 13 “Diagnosis handling”. It is used so that pre-defined or flexible diagnostic messages can be conveyed from the EtherCAT Slave to the Master. In accordance with the ETG, the process can therefore be implemented supplier-independently. Support is optional. The firmware can store up to 250 *DiagMessages* in its own CoE.

Each *DiagMessage* consists of

- Diag Code (4-byte)
- Flags (2-byte; info, warning or error)
- Text ID (2-byte; reference to explanatory text from the ESI/XML)
- Timestamp (8-byte, local slave time or 64-bit Distributed Clock time, if available)
- Dynamic parameters added by the firmware

The *DiagMessages* are explained in text form in the ESI/XML file belonging to the EtherCAT device: on the basis of the Text ID contained in the *DiagMessage*, the corresponding plain text message can be found in the languages contained in the ESI/XML. In the case of Beckhoff products these are usually German and English.

Via the entry *NewMessagesAvailable* the user receives information that new messages are available.

DiagMessages can be confirmed in the device: the last/latest unconfirmed message can be confirmed by the user.

In the CoE both the control entries and the history itself can be found in the CoE object 0x10F3:

Index	Name	Flags	Value
1018:0	Identity	RO	> 4 <
10F0:0	Backup parameter handling	RO	> 1 <
10F3:0	Diagnosis History	RO	> 55 <
10F3:01	Maximum Messages	RO	0x32 (50)
10F3:02	Newest Message	RO	0x15 (21)
10F3:03	Newest Acknowledged Message	R/W	0x14 (20)
10F3:04	New Messages Available	RO	FALSE
10F3:05	Flags	R/W	0x0000 (0)
10F3:06	Diagnosis Message 001	RO	00 E0 A4 08 10 00 03 00 60 1F 0D 00 00 00 00 00 06 00 00 00 06 00 00 06 00 00 06 00 FF 00
10F3:07	Diagnosis Message 002	RO	00 E0 A4 08 10 00 02 00 00 6A 18 00 00 00 00 00 06 00 00 00 06 00 00 06 00 00 06 00 00 00
10F3:08	Diagnosis Message 003	RO	00 E0 A4 08 10 00 03 00 40 D8 67 02 00 00 00 00 06 00 00 00 06 00 00 03 00 06 00 00 00
10F3:09	Diagnosis Message 004	RO	00 E0 A4 08 12 00 00 81 E0 89 47 03 00 00 00 00 06 00 04 44 06 00 00 00 06 00 00 00

Fig. 156: *DiagMessages* in the CoE

The subindex of the latest *DiagMessage* can be read under 0x10F3:02.

Support for commissioning

The DiagMessages system is to be used above all during the commissioning of the plant. The diagnostic values e.g. in the StatusWord of the device (if available) are helpful for online diagnosis during the subsequent continuous operation.

TwinCAT System Manager implementation

From TwinCAT 2.11 DiagMessages, if available, are displayed in the device's own interface. Operation (collection, confirmation) also takes place via this interface.

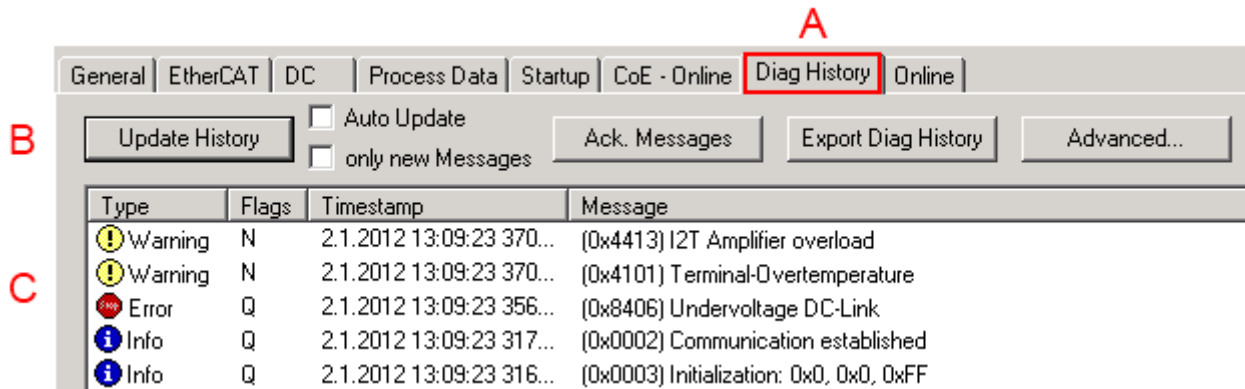


Fig. 157: Implementation of the DiagMessage system in the TwinCAT System Manager

The operating buttons (B) and the history read out (C) can be seen on the Diag History tab (A). The components of the message:

- Info/Warning/Error
- Acknowledge flag (N = unconfirmed, Q = confirmed)
- Time stamp
- Text ID
- Plain text message according to ESI/XML data

The meanings of the buttons are self-explanatory.

DiagMessages within the ADS Logger/Eventlogger

Since TwinCAT 3.1 build 4022 DiagMessages send by the terminal are shown by the TwinCAT ADS Logger. Given that DiagMessages are represented IO- comprehensive at one place, commissioning will be simplified. In addition, the logger output could be stored into a data file – hence DiagMessages are available long-term for analysis.

DiagMessages are actually only available locally in CoE 0x10F3 in the terminal and can be read out manually if required, e.g. via the DiagHistory mentioned above.

In the latest developments, the EtherCAT Terminals are set by default to report the presence of a DiagMessage as emergency via EtherCAT; the event logger can then retrieve the DiagMessage. The function is activated in the terminal via 0x10F3:05, so such terminals have the following entry in the StartUp list by default:

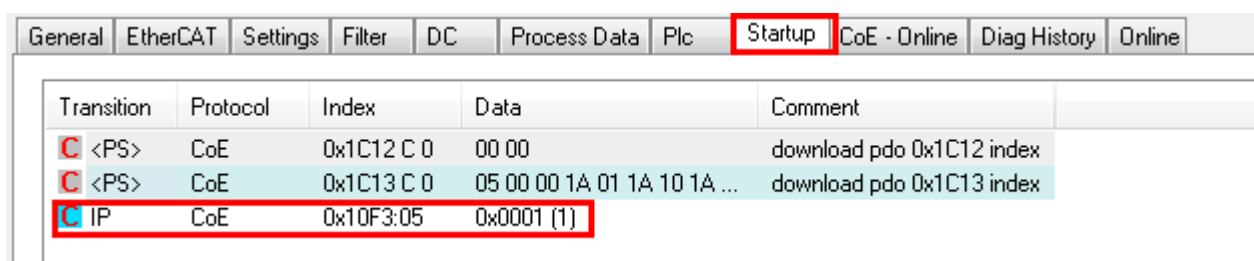


Fig. 158: Startup List

If the function is to be deactivated because, for example, many messages come in or the EventLogger is not used, the StartUp entry can be deleted or set to 0.

Reading messages into the PLC

- In preparation -

Interpretation

Time stamp

The time stamp is obtained from the local clock of the terminal at the time of the event. The time is usually the distributed clock time (DC) from register x910.

Please note: When EtherCAT is started, the DC time in the reference clock is set to the same time as the local IPC/TwinCAT time. From this moment the DC time may differ from the IPC time, since the IPC time is not adjusted. Significant time differences may develop after several weeks of operation without a EtherCAT restart. As a remedy, external synchronization of the DC time can be used, or a manual correction calculation can be applied, as required: The current DC time can be determined via the EtherCAT master or from register x901 of the DC slave.

Structure of the Text ID

The structure of the MessageID is not subject to any standardization and can be supplier-specifically defined. In the case of Beckhoff EtherCAT devices (EL, EP) it usually reads according to **xyzz**:

x	y	zz
0: Systeminfo 2: reserved 1: Info 4: Warning 8: Error	0: System 1: General 2: Communication 3: Encoder 4: Drive 5: Inputs 6: I/O general 7: reserved	Error number

Example: Message 0x4413 --> Drive Warning Number 0x13

Overview of text IDs

Specific text IDs are listed in the device documentation.

Text ID	Type	Place	Text Message	Additional comment
0x0001	Information	System	No error	No error
0x0002	Information	System	Communication established	Connection established
0x0003	Information	System	Initialization: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1000	Information	System	Information: 0x%X, 0x%X, 0x%X	General information; parameters depend on event. See device documentation for interpretation.
0x1012	Information	System	EtherCAT state change Init - PreOp	
0x1021	Information	System	EtherCAT state change PreOp - Init	
0x1024	Information	System	EtherCAT state change PreOp - Safe-Op	
0x1042	Information	System	EtherCAT state change SafeOp - PreOp	
0x1048	Information	System	EtherCAT state change SafeOp - Op	
0x1084	Information	System	EtherCAT state change Op - SafeOp	
0x1100	Information	General	Detection of operation mode completed: 0x%X, %d	Detection of the mode of operation ended
0x1135	Information	General	Cycle time o.k.: %d	Cycle time OK
0x1157	Information	General	Data manually saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved manually
0x1158	Information	General	Data automatically saved (Idx: 0x%X, SubIdx: 0x%X)	Data saved automatically
0x1159	Information	General	Data deleted (Idx: 0x%X, SubIdx: 0x%X)	Data deleted
0x117F	Information	General	Information: 0x%X, 0x%X, 0x%X	Information
0x1201	Information	Communication	Communication re-established	Communication to the field side restored This message appears, for example, if the voltage was removed from the power contacts and re-applied during operation.
0x1300	Information	Encoder	Position set: %d, %d	Position set - StartInputhandler
0x1303	Information	Encoder	Encoder Supply ok	Encoder power supply unit OK
0x1304	Information	Encoder	Encoder initialization successfully, channel: %X	Encoder initialization successfully completed
0x1305	Information	Encoder	Sent command encoder reset, channel: %X	Send encoder reset command
0x1400	Information	Drive	Drive is calibrated: %d, %d	Drive is calibrated
0x1401	Information	Drive	Actual drive state: 0x%X, %d	Current drive status
0x1705	Information		CPU usage returns in normal range (< 85%%)	Processor load is back in the normal range
0x1706	Information		Channel is not in saturation anymore	Channel is no longer in saturation
0x1707	Information		Channel is not in overload anymore	Channel is no longer overloaded
0x170A	Information		No channel range error anymore	A measuring range error is no longer active
0x170C	Information		Calibration data saved	Calibration data were saved
0x170D	Information		Calibration data will be applied and saved after sending the command "0x5AFE"	Calibration data are not applied and saved until the command "0x5AFE" is sent.

Text ID	Type	Place	Text Message	Additional comment
0x2000	Information	System	%s: %s	
0x2001	Information	System	%s: Network link lost	Network connection lost
0x2002	Information	System	%s: Network link detected	Network connection found
0x2003	Information	System	%s: no valid IP Configuration - Dhcp client started	Invalid IP configuration
0x2004	Information	System	%s: valid IP Configuration (IP: %d.%d.%d.%d) assigned by Dhcp server %d.%d.%d.%d	Valid IP configuration, assigned by the DHCP server
0x2005	Information	System	%s: Dhcp client timed out	DHCP client timeout
0x2006	Information	System	%s: Duplicate IP Address detected (%d.%d.%d.%d)	Duplicate IP address found
0x2007	Information	System	%s: UDP handler initialized	UDP handler initialized
0x2008	Information	System	%s: TCP handler initialized	TCP handler initialized
0x2009	Information	System	%s: No more free TCP sockets available	No free TCP sockets available.

Text ID	Type	Place	Text Message	Additional comment
0x4000	Warning		Warning: 0x%X, 0x%X, 0x%X	General warning; parameters depend on event. See device documentation for interpretation.
0x4001	Warning	System	Warning: 0x%X, 0x%X, 0x%X	
0x4002	Warning	System	%s: %s Connection Open (IN:%d OUT:%d API:%dms) from %d. %d.%d.%d successful	
0x4003	Warning	System	%s: %s Connection Close (IN:%d OUT:%d) from %d.%d.%d.%d successful	
0x4004	Warning	System	%s: %s Connection (IN:%d OUT:%d) with %d.%d.%d.%d timed out	
0x4005	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Error: %u)	
0x4006	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Input Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4007	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Output Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4008	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (RPI:%dms not supported -> API:%dms)	
0x4101	Warning	General	Terminal-Overtemperature	Overtemperature. The internal temperature of the terminal exceeds the parameterized warning threshold.
0x4102	Warning	General	Discrepancy in the PDO-Configuration	The selected PDOs do not match the set operating mode. Sample: Drive operates in velocity mode, but the velocity PDO is but not mapped in the PDOs.
0x417F	Warning	General	Warning: 0x%X, 0x%X, 0x%X	
0x428D	Warning	General	Challenge is not Random	
0x4300	Warning	Encoder	Subincrements deactivated: %d, %d	Sub-increments deactivated (despite activated configuration)
0x4301	Warning	Encoder	Encoder-Warning	General encoder error
0x4302	Warning	Encoder	Maximum frequency of the input signal is nearly reached (channel %d)	
0x4303	Warning	Encoder	Limit counter value was reduced because of the PDO configuration (channel %d)	
0x4304	Warning	Encoder	Reset counter value was reduced because of the PDO configuration (channel %d)	
0x4400	Warning	Drive	Drive is not calibrated: %d, %d	Drive is not calibrated
0x4401	Warning	Drive	Starttype not supported: 0x%X, %d	Start type is not supported
0x4402	Warning	Drive	Command rejected: %d, %d	Command rejected
0x4405	Warning	Drive	Invalid modulo subtype: %d, %d	Modulo sub-type invalid
0x4410	Warning	Drive	Target overrun: %d, %d	Target position exceeded
0x4411	Warning	Drive	DC-Link undervoltage (Warning)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x4412	Warning	Drive	DC-Link overvoltage (Warning)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x4413	Warning	Drive	I2T-Model Amplifier overload (Warning)	<ul style="list-style-type: none"> The amplifier is being operated outside the specification. The I2T-model of the amplifier is incorrectly parameterized.
0x4414	Warning	Drive	I2T-Model Motor overload (Warning)	<ul style="list-style-type: none"> The motor is being operated outside the parameterized rated values. The I2T-model of the motor is incorrectly parameterized.

Text ID	Type	Place	Text Message	Additional comment
0x4415	Warning	Drive	Speed limitation active	The maximum speed is limited by the parameterized objects (e.g. velocity limitation, motor speed limitation). This warning is output if the set velocity is higher than one of the parameterized limits.
0x4416	Warning	Drive	Step lost detected at position: 0x %X%X	Step loss detected
0x4417	Warning	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized warning threshold
0x4418	Warning	Drive	Limit: Current	Limit: current is limited
0x4419	Warning	Drive	Limit: Amplifier I2T-model exceeds 100%%	The threshold values for the maximum current were exceeded.
0x441A	Warning	Drive	Limit: Motor I2T-model exceeds 100%%	Limit: Motor I2T-model exceeds 100%
0x441B	Warning	Drive	Limit: Velocity limitation	The threshold values for the maximum speed were exceeded.
0x441C	Warning	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x4600	Warning	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x4610	Warning	General IO	Wrong output voltage range	Output voltage not in the correct range
0x4705	Warning		Processor usage at %d %%	Processor load at %d %%
0x470A	Warning		EtherCAT Frame missed (change Settings or DC Operation Mode or Sync0 Shift Time)	EtherCAT frame missed (change DC Operation Mode or Sync0 Shift Time under Settings)

Text ID	Type	Place	Text Message	Additional comment
0x8000	Error	System	%s: %s	
0x8001	Error	System	Error: 0x%X, 0x%X, 0x%X	General error; parameters depend on event. See device documentation for interpretation.
0x8002	Error	System	Communication aborted	Communication aborted
0x8003	Error	System	Configuration error: 0x%X, 0x%X, 0x%X	General; parameters depend on event. See device documentation for interpretation.
0x8004	Error	System	%s: Unsuccessful FwdOpen-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8005	Error	System	%s: FwdClose-Request sent to %d.%d.%d.%d (%s)	
0x8006	Error	System	%s: Unsuccessful FwdClose-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8007	Error	System	%s: Connection with %d.%d.%d.%d (%s) closed	
0x8100	Error	General	Status word set: 0x%X, %d	Error bit set in the status word
0x8101	Error	General	Operation mode incompatible to PDO interface: 0x%X, %d	Mode of operation incompatible with the PDO interface
0x8102	Error	General	Invalid combination of Inputs and Outputs PDOs	Invalid combination of input and output PDOs
0x8103	Error	General	No variable linkage	No variables linked
0x8104	Error	General	Terminal-Overtemperature	The internal temperature of the terminal exceeds the parameterized error threshold. Activation of the terminal is prevented
0x8105	Error	General	PD-Watchdog	Communication between the fieldbus and the output stage is secured by a Watchdog. The axis is stopped automatically if the fieldbus communication is interrupted. <ul style="list-style-type: none"> The EtherCAT connection was interrupted during operation. The Master was switched to Config mode during operation.
0x8135	Error	General	Cycle time has to be a multiple of 125 µs	The IO or NC cycle time divided by 125 µs does not produce a whole number.
0x8136	Error	General	Configuration error: invalid sampling rate	Configuration error: Invalid sampling rate
0x8137	Error	General	Electronic type plate: CRC error	Content of the external name plate memory invalid.
0x8140	Error	General	Sync Error	Real-time violation
0x8141	Error	General	Sync%X Interrupt lost	Sync%X Interrupt lost
0x8142	Error	General	Sync Interrupt asynchronous	Sync Interrupt asynchronous
0x8143	Error	General	Jitter too big	Jitter limit violation
0x817F	Error	General	Error: 0x%X, 0x%X, 0x%X	
0x8200	Error	Communication	Write access error: %d, %d	Error while writing
0x8201	Error	Communication	No communication to field-side (Auxiliary voltage missing)	<ul style="list-style-type: none"> There is no voltage applied to the power contacts. A firmware update has failed.
0x8281	Error	Communication	Ownership failed: %X	
0x8282	Error	Communication	To many Keys founded	
0x8283	Error	Communication	Key Creation failed: %X	
0x8284	Error	Communication	Key loading failed	
0x8285	Error	Communication	Reading Public Key failed: %X	
0x8286	Error	Communication	Reading Public EK failed: %X	
0x8287	Error	Communication	Reading PCR Value failed: %X	
0x8288	Error	Communication	Reading Certificate EK failed: %X	
0x8289	Error	Communication	Challenge could not be hashed: %X	
0x828A	Error	Communication	Tickstamp Process failed	
0x828B	Error	Communication	PCR Process failed: %X	
0x828C	Error	Communication	Quote Process failed: %X	
0x82FF	Error	Communication	Bootmode not activated	Boot mode not activated
0x8300	Error	Encoder	Set position error: 0x%X, %d	Error while setting the position

Text ID	Type	Place	Text Message	Additional comment
0x8301	Error	Encoder	Encoder increments not configured: 0x%X, %d	Encoder increments not configured
0x8302	Error	Encoder	Encoder error	The amplitude of the resolver is too small
0x8303	Error	Encoder	Encoder power missing (channel %d)	
0x8304	Error	Encoder	Encoder communication error, channel: %X	Encoder communication error
0x8305	Error	Encoder	EnDat2.2 is not supported, channel: %X	EnDat2.2 is not supported
0x8306	Error	Encoder	Delay time, tolerance limit exceeded, 0x%X, channel: %X	Runtime measurement, tolerance exceeded
0x8307	Error	Encoder	Delay time, maximum value exceeded, 0x%X, channel: %X	Runtime measurement, maximum value exceeded
0x8308	Error	Encoder	Unsupported ordering designation, 0x%X, channel: %X (only 02 and 22 is supported)	Wrong EnDat order ID
0x8309	Error	Encoder	Encoder CRC error, channel: %X	Encoder CRC error
0x830A	Error	Encoder	Temperature %X could not be read, channel: %X	Temperature cannot be read
0x830C	Error	Encoder	Encoder Single-Cycle-Data Error, channel: %X	CRC error detected. Check the transmission path and the CRC polynomial
0x830D	Error	Encoder	Encoder Watchdog Error, channel: %X	The sensor has not responded within a predefined time period
0x8310	Error	Encoder	Initialisation error	
0x8311	Error	Encoder	Maximum frequency of the input signal is exceeded (channel %d)	
0x8312	Error	Encoder	Encoder plausibility error (channel %d)	
0x8313	Error	Encoder	Configuration error (channel %d)	
0x8314	Error	Encoder	Synchronisation error	
0x8315	Error	Encoder	Error status input (channel %d)	
0x8400	Error	Drive	Incorrect drive configuration: 0x %X, %d	Drive incorrectly configured
0x8401	Error	Drive	Limiting of calibration velocity: %d, %d	Limitation of the calibration velocity
0x8402	Error	Drive	Emergency stop activated: 0x%X, %d	Emergency stop activated
0x8403	Error	Drive	ADC Error	Error during current measurement in the ADC
0x8404	Error	Drive	Overcurrent	Overcurrent in phase U, V or W
0x8405	Error	Drive	Invalid modulo position: %d	Modulo position invalid
0x8406	Error	Drive	DC-Link undervoltage (Error)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x8407	Error	Drive	DC-Link overvoltage (Error)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x8408	Error	Drive	I2T-Model Amplifier overload (Error)	<ul style="list-style-type: none"> The amplifier is being operated outside the specification. The I2T-model of the amplifier is incorrectly parameterized.
0x8409	Error	Drive	I2T-Model motor overload (Error)	<ul style="list-style-type: none"> The motor is being operated outside the parameterized rated values. The I2T-model of the motor is incorrectly parameterized.
0x840A	Error	Drive	Overall current threshold exceeded	Total current exceeded
0x8415	Error	Drive	Invalid modulo factor: %d	Modulo factor invalid
0x8416	Error	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized error threshold. The motor stops immediately. Activation of the output stage is prevented.
0x8417	Error	Drive	Maximum rotating field velocity exceeded	Rotary field speed exceeds the value specified for dual use (EU 1382/2014).
0x841C	Error	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x8550	Error	Inputs	Zero crossing phase %X missing	Zero crossing phase %X missing

Text ID	Type	Place	Text Message	Additional comment
0x8551	Error	Inputs	Phase sequence Error	Wrong direction of rotation
0x8552	Error	Inputs	Overcurrent phase %X	Overcurrent phase %X
0x8553	Error	Inputs	Overcurrent neutral wire	Overcurrent neutral wire
0x8581	Error	Inputs	Wire broken Ch %D	Wire broken Ch %d
0x8600	Error	General IO	Wrong supply voltage range	Supply voltage not in the correct range
0x8601	Error	General IO	Supply voltage to low	Supply voltage too low
0x8602	Error	General IO	Supply voltage to high	Supply voltage too high
0x8603	Error	General IO	Over current of supply voltage	Overcurrent of supply voltage
0x8610	Error	General IO	Wrong output voltage range	Output voltage not in the correct range
0x8611	Error	General IO	Output voltage to low	Output voltage too low
0x8612	Error	General IO	Output voltage to high	Output voltage too high
0x8613	Error	General IO	Over current of output voltage	Overcurrent of output voltage
0x8700	Error		Channel/Interface not calibrated	Channel/interface not synchronized
0x8701	Error		Operating time was manipulated	Operating time was manipulated
0x8702	Error		Oversampling setting is not possible	Oversampling setting not possible
0x8703	Error		No slave controller found	No slave controller found
0x8704	Error		Slave controller is not in Bootstrap	Slave controller is not in bootstrap
0x8705	Error		Processor usage to high (>= 100%%)	Processor load too high (>= 100%%)
0x8706	Error		Channel in saturation	Channel in saturation
0x8707	Error		Channel overload	Channel overload
0x8708	Error		Overloadtime was manipulated	Overload time was manipulated
0x8709	Error		Saturationtime was manipulated	Saturation time was manipulated
0x870A	Error		Channel range error	Measuring range error for the channel
0x870B	Error		no ADC clock	No ADC clock available
0xFFFF	Information		Debug: 0x%X, 0x%X, 0x%X	Debug: 0x%X, 0x%X, 0x%X

6.2 Diagnostics - specific diag messages - EL2595

Interpretation

Structure of the Text ID

The structure of the MessageID is not subject to any standardization and can be supplier-specifically defined. In the case of Beckhoff EtherCAT devices (EL, EP) it usually reads according to **xyzz**:

x	y	zz
0: Systeminfo 1: Info 4: Warning 8: Error	1: General 2: Communication 3: Encoder 4: Drive 5: Inputs 6: IO	Error number

Example: Message 0x4413 --> Drive Warning Number 0x13

Specific text IDs - EL2595

Text ID	Type	Location	Text (English only)	Cause
0x0002	Information	System	Communication established	Communication established
0x0003	Information	System	Initialization: 0x%X, 0x%X, 0x%X	Initialization: 0x%X, 0x%X, 0x%X
0x1157	Information	General	Data manually saved (Idx: 0x%X, SubIdx: 0x%X)	Data manually saved (Idx: 0x%X, SubIdx: 0x%X)
0x1158	Information	General	Data automatically saved (Idx: 0x%X, SubIdx: 0x%X)	Data automatically saved (Idx: 0x%X, SubIdx: 0x%X)
0x1159	Information	General	Data deleted (Idx: 0x%X, SubIdx: 0x%X)	Data deleted (Idx: 0x%X, SubIdx: 0x%X)
0x1201	Information	Communication	Communication re-established	Communication to the field side restored This message appears, for example, if the voltage was removed from the power contacts and re-applied during operation
0x4600	Warning	IO	Wrong supply voltage range	Supply voltage not in the correct range
0x4610	Warning	IO	Wrong output voltage range	Output voltage not in the correct range
0x8002	Error	System	Communication aborted	Communication aborted
0x8100	Error	General	Error bit set in status word 0x%X, %X	Error bit set in status word 0x%X, %X
0x8200	Error	Communication	Write access error: %d, %d	Error while writing: %d, %d
0x8201	Error	Communication	No communication to field-side (Auxiliary voltage missing)	- There is no voltage applied to the power contacts - A firmware update has failed
0x8601	Error	IO	Supply voltage too low	Supply voltage too low
0x8602	Error	IO	Supply voltage too high	Supply voltage too high
0x8612	Error	IO	Output voltage too high	Output voltage too high
0x8613	Error	IO	Over current of output voltage	Overcurrent of output voltage

7 Appendix

7.1 Firmware compatibility

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

Note

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

NOTE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the [separate page \[▶ 158\]](#).

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!

EL2595			
Hardware (HW)	Firmware (FW)	Revision no.	Date of release
00	01 (only prototypes)	EL2595-0000-0017	2013/07
	02	EL2595-0000-0018	2013/07
	03		2013/11
01 - 03*	04	EL2595-0000-0019	2014/11
	05		2015/11
	06		2020/03
	07		2020/03
	08*		2020/03

*) 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.

Note regarding EL2595

NOTE

Secondary voltage necessary for firmware updates

In order to change the firmware of the EL2595, the permissible operating voltages must be applied both to the fieldbus side (E-bus) and to the load side (power contacts), otherwise the terminal remains in INIT_ERROR or cannot be switched to Bootstrap mode.

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

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

NOTE**Only use TwinCAT 3 software!**

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

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

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

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

Storage locations

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

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

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxx-xxx_REV0016_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

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

NOTE**Risk of damage to the device!**

- ✓ Note the following when downloading new device files
 - a) Firmware downloads to an EtherCAT device must not be interrupted
 - b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.
 - c) The power supply must adequately dimensioned. The signal level must meet the specification.
- ⇒ In the event of malfunctions during the update process the EtherCAT device may become unusable and require re-commissioning by the manufacturer.

7.2.1 Device description ESI file/XML**NOTE****Attention regarding update of the ESI description/EEPROM**

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

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

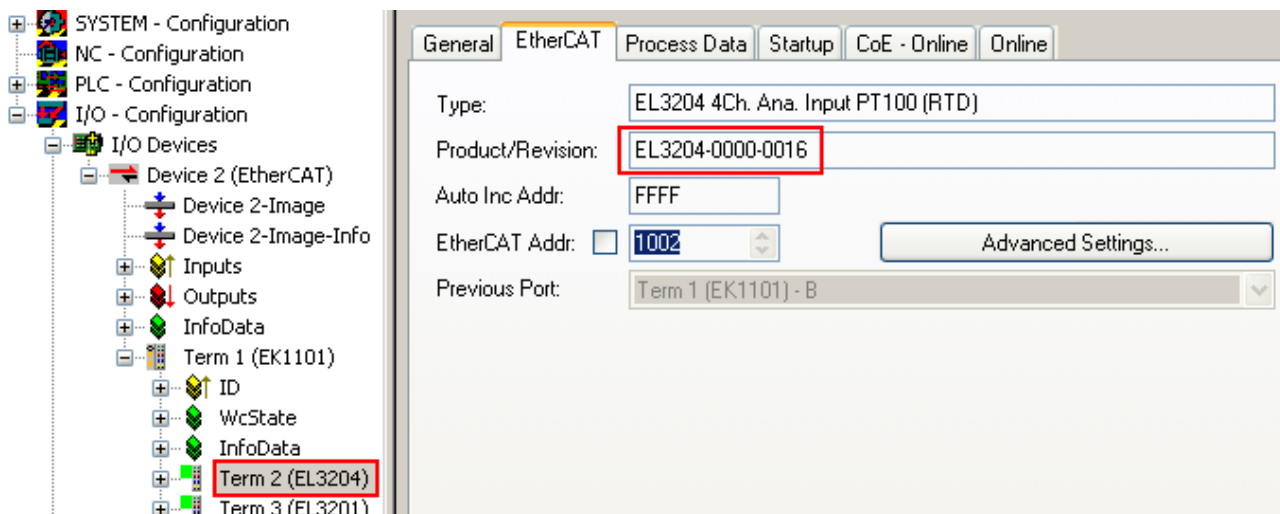


Fig. 159: Device identifier consisting of name EL3204-0000 and revision -0016

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

i 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:

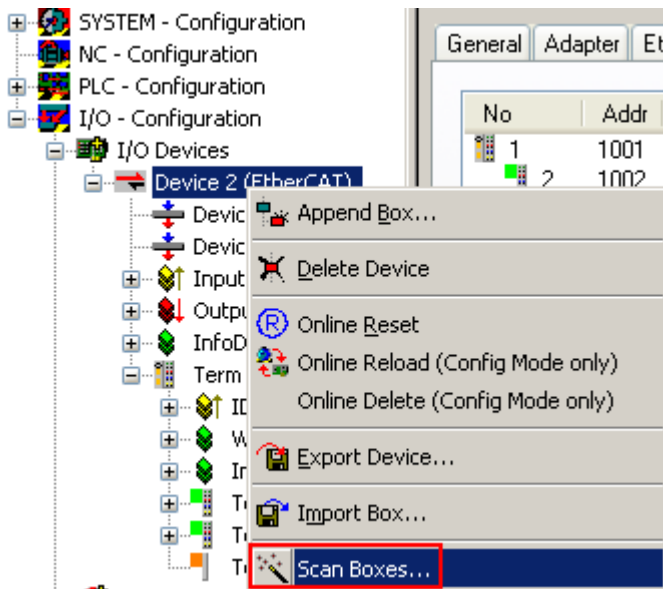


Fig. 160: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 161: Configuration is identical

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

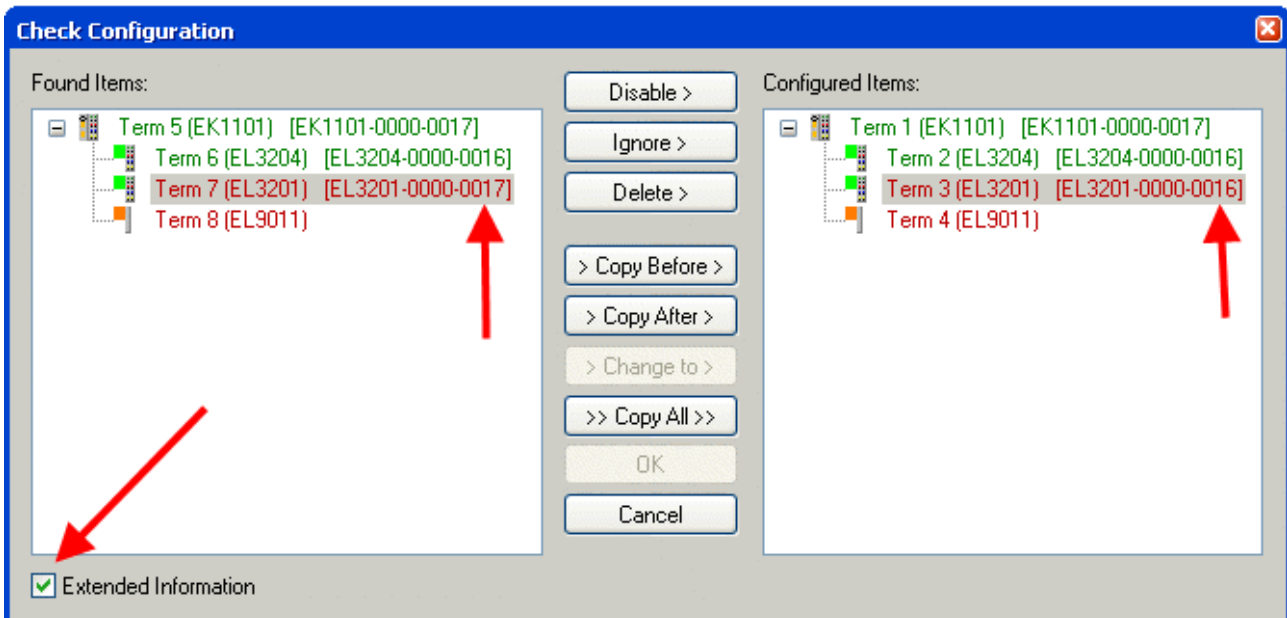


Fig. 162: Change dialog

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*

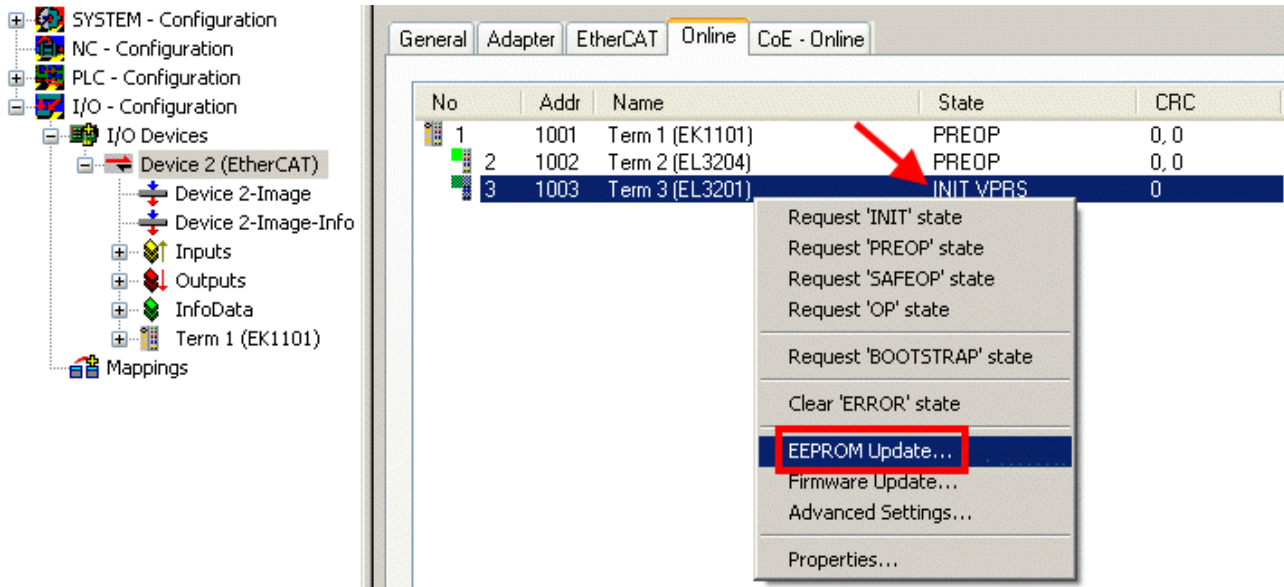


Fig. 163: EEPROM Update

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.

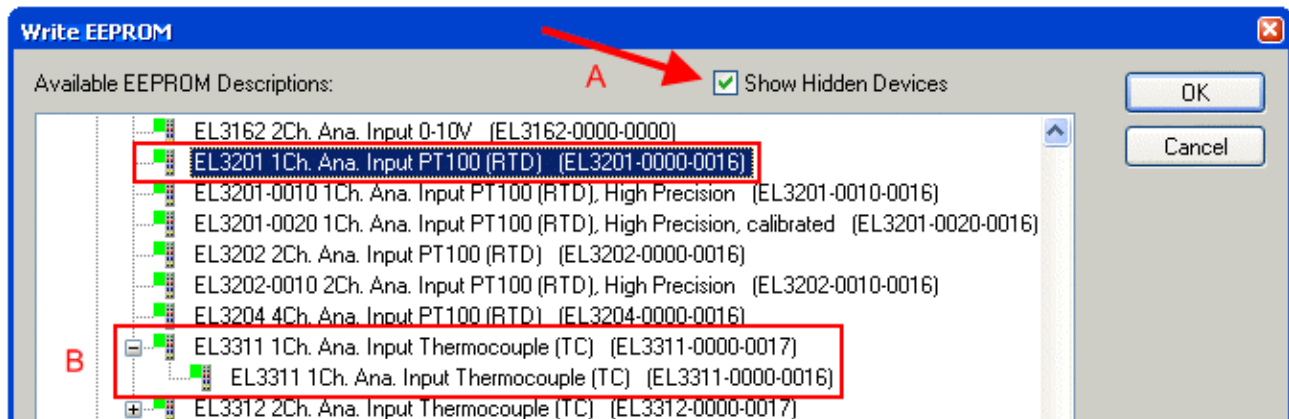


Fig. 164: Selecting the new ESI

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

i 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.

7.2.2 Firmware explanation

Determining the firmware version

Determining the version on laser inscription

Beckhoff EtherCAT slaves feature serial numbers applied by laser. The serial number has the following structure: **KK YY FF HH**

- KK - week of production (CW, calendar week)
- YY - year of production
- FF - firmware version
- HH - hardware version

Example with ser. no.: 12 10 03 02:

- 12 - week of production 12
- 10 - year of production 2010
- 03 - firmware version 03
- 02 - hardware version 02

Determining the version via the System Manager

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

● CoE Online and Offline CoE

i

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.

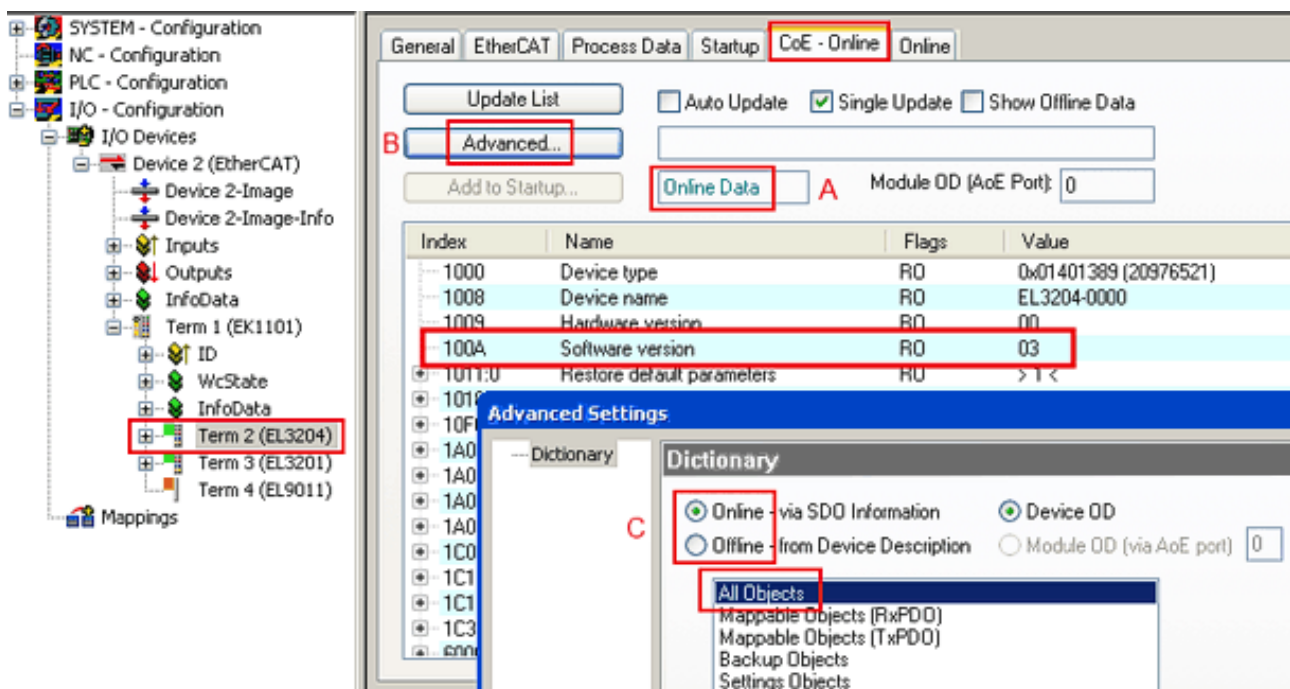


Fig. 165: Display of EL3204 firmware version

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*.

7.2.3 Updating controller firmware *.efw

● CoE directory

i 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*.

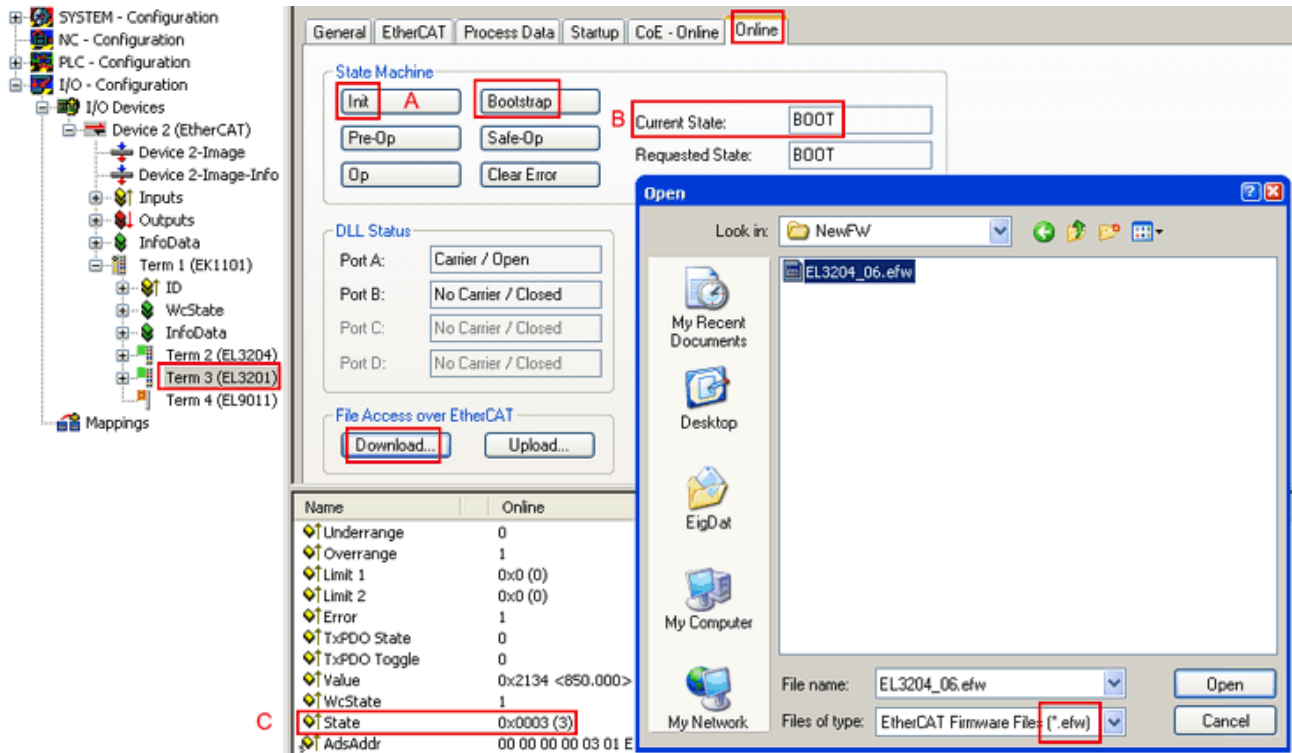
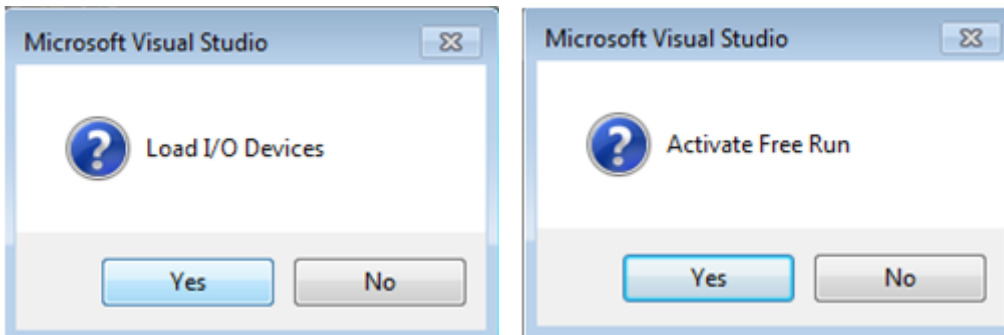


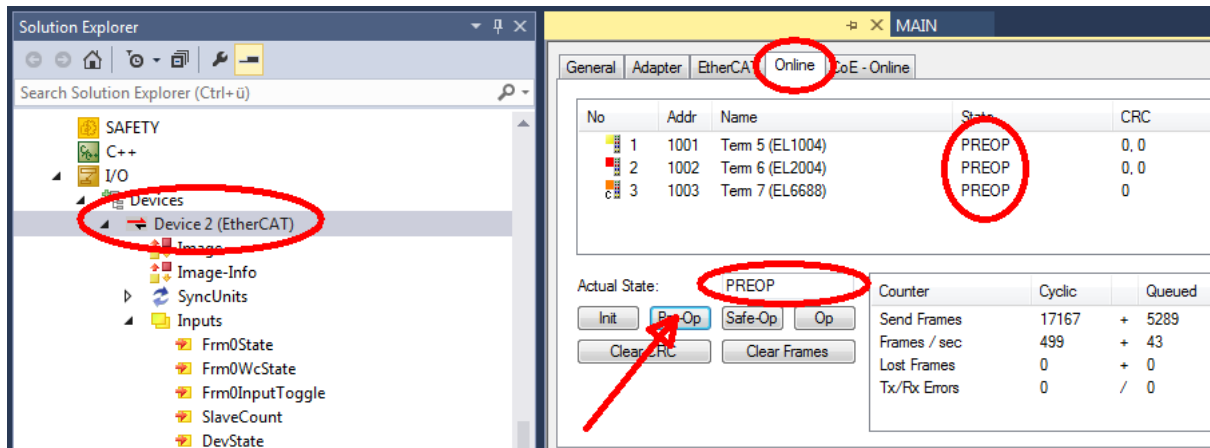
Fig. 166: Firmware Update

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

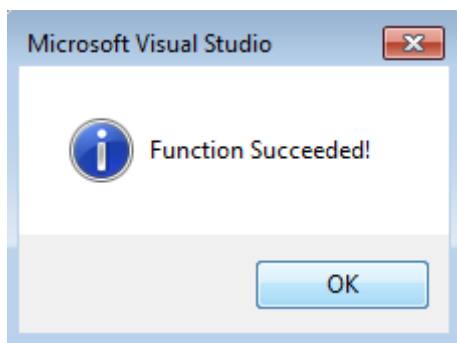
- Switch TwinCAT system to ConfigMode/FreeRun with cycle time ≥ 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.



- 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 pass word 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.

7.2.4 FPGA firmware *.rbf

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

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

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

Determining the version via the System Manager

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

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

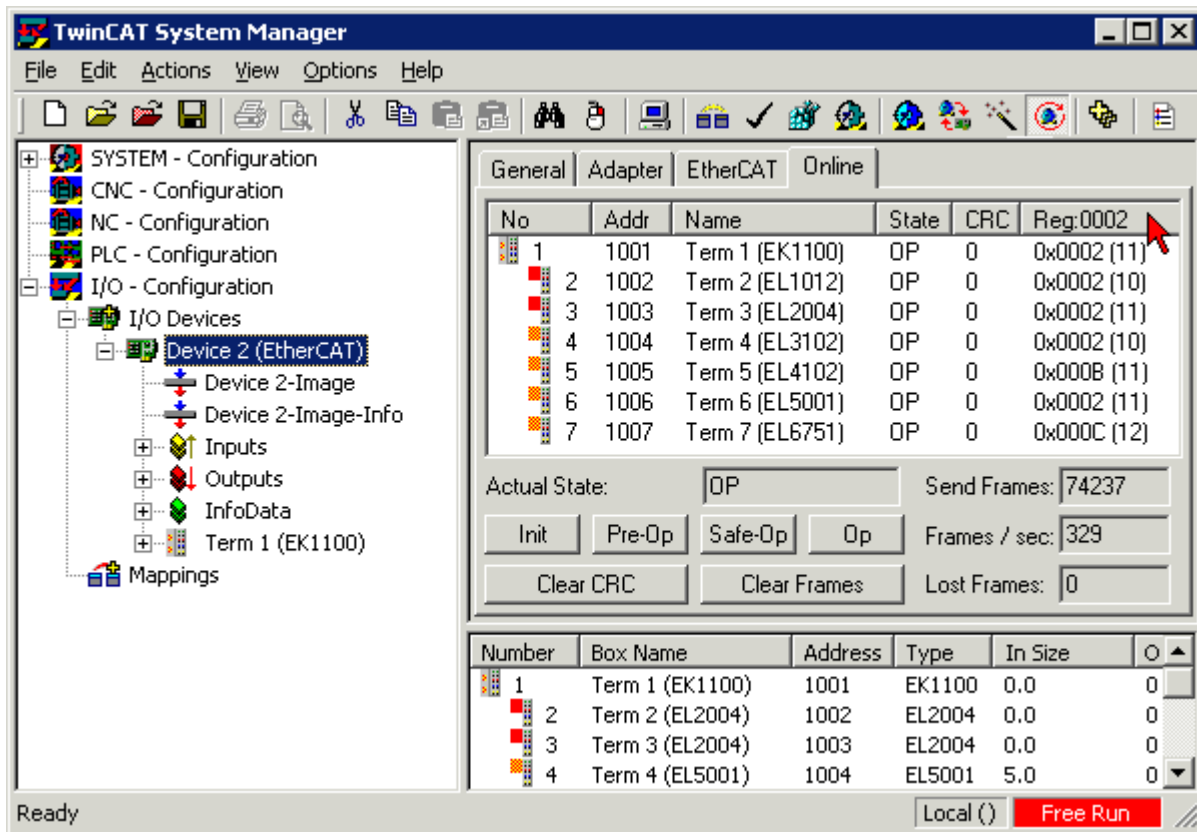
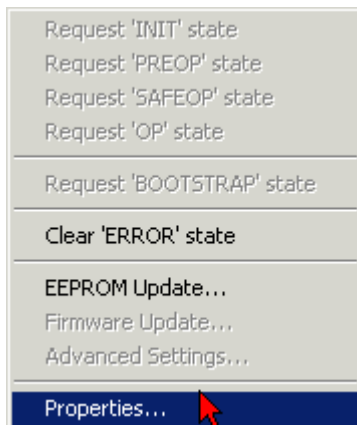


Fig. 167: FPGA firmware version definition

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

Fig. 168: 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.

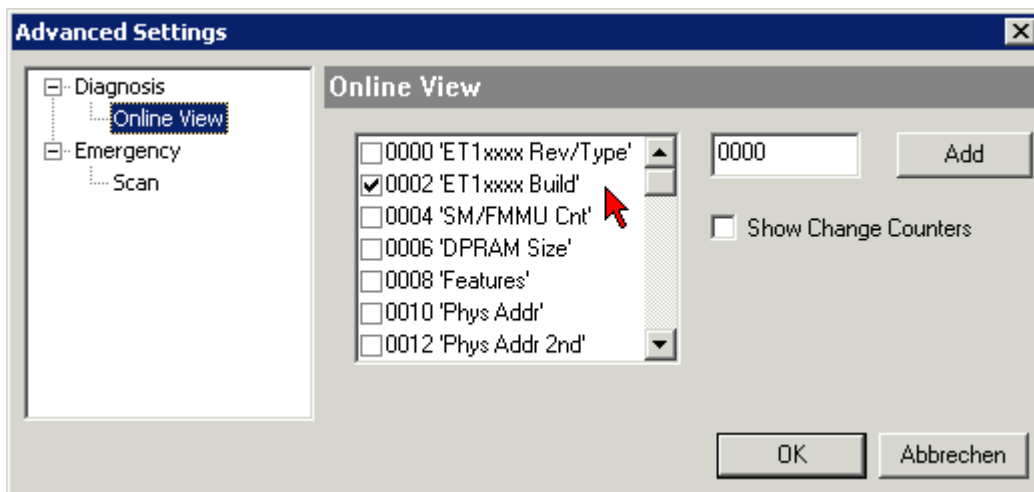


Fig. 169: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

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

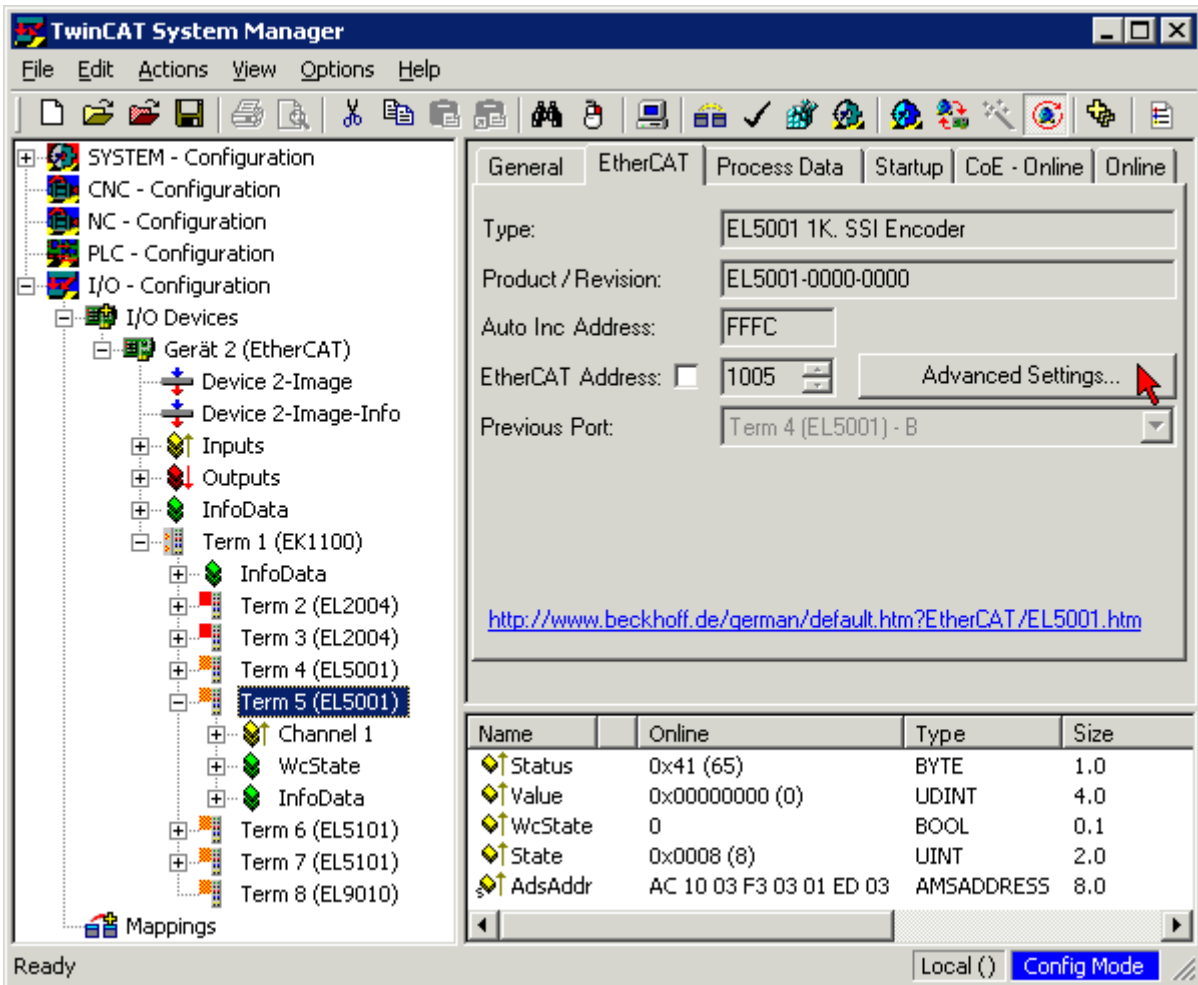
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

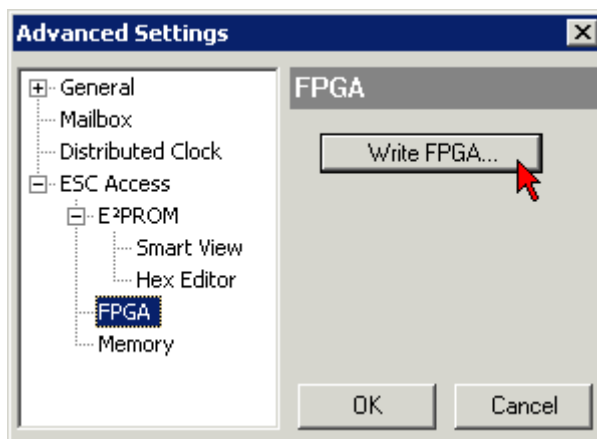
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

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

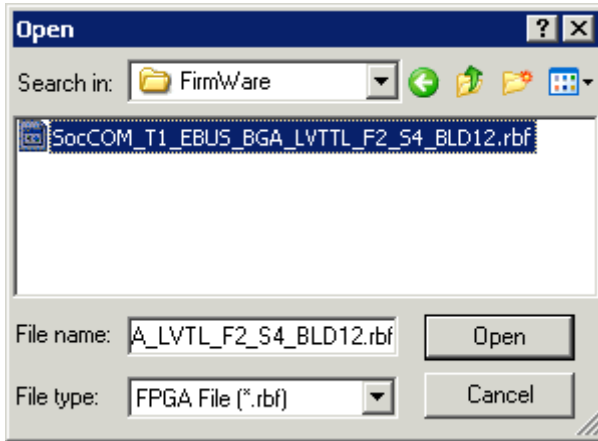
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



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



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



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

NOTE

Risk of damage to the device!

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

7.2.5 Simultaneous updating of several EtherCAT devices

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

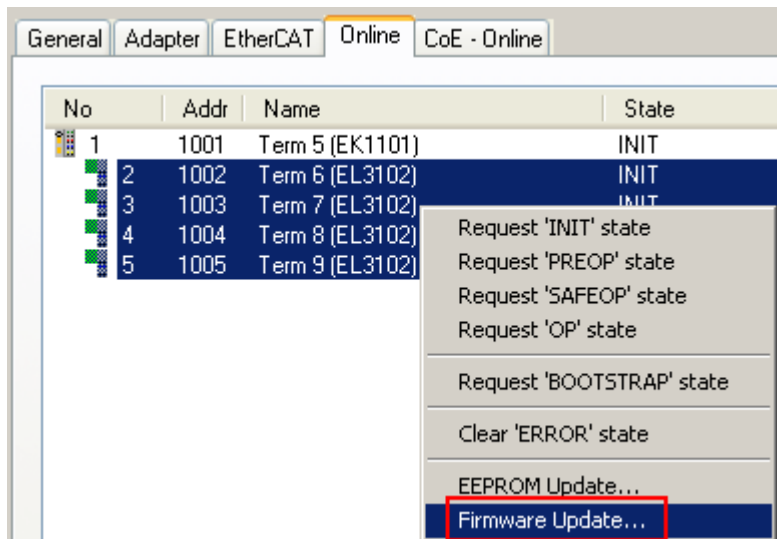


Fig. 170: Multiple selection and firmware update

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

7.3 Restoring the delivery state

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

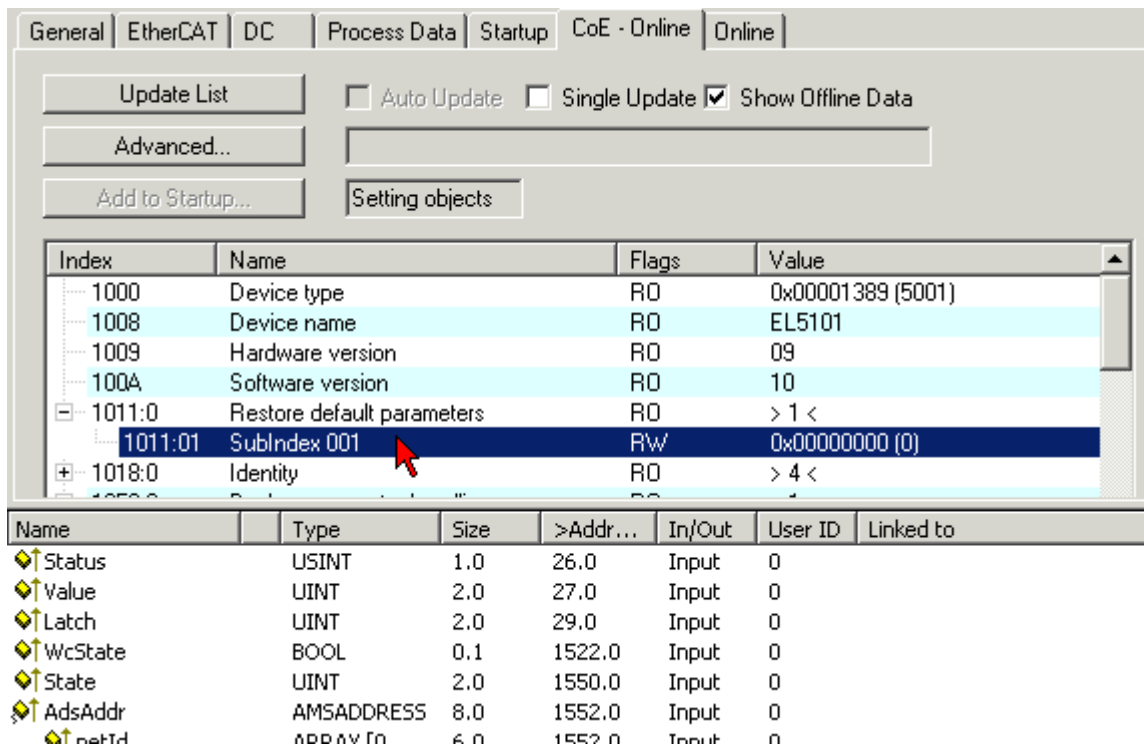


Fig. 171: Selecting the *Restore default parameters* PDO

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

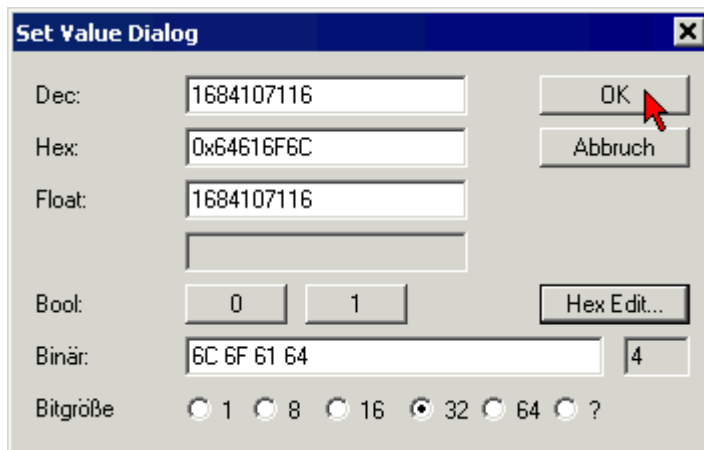


Fig. 172: Entering a restore value in the Set Value dialog

● Alternative restore value

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

7.4 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

Hotline: +49 5246 963 157
Fax: +49 5246 963 9157
e-mail: support@beckhoff.com

Beckhoff Service

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

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

Hotline: +49 5246 963 460
Fax: +49 5246 963 479
e-mail: service@beckhoff.com

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone: +49 5246 963 0
Fax: +49 5246 963 198
e-mail: info@beckhoff.com
web: <https://www.beckhoff.com>

List of figures

Fig. 1	EL5021 EL terminal, standard IP20 IO device with serial/ batch number and revision ID (since 2014/01).....	8
Fig. 2	EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number.....	9
Fig. 3	EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418	9
Fig. 4	BIC as data matrix code (DMC, code scheme ECC200).....	10
Fig. 5	EL2595	12
Fig. 6	System manager current calculation	15
Fig. 7	EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog	16
Fig. 8	States of the EtherCAT State Machine.....	18
Fig. 9	“CoE Online” tab	20
Fig. 10	Startup list in the TwinCAT System Manager	21
Fig. 11	Offline list.....	22
Fig. 12	Online list	22
Fig. 13	Attaching on mounting rail	25
Fig. 14	Disassembling of terminal.....	26
Fig. 15	Power contact on left side.....	27
Fig. 16	Standard wiring.....	28
Fig. 17	Pluggable wiring	28
Fig. 18	High Density Terminals.....	28
Fig. 19	Mounting a cable on a terminal connection	29
Fig. 20	Correct positioning.....	30
Fig. 21	Incorrect positioning.....	31
Fig. 22	Recommended minimum distances for standard installation position	32
Fig. 23	EL2595 LEDs.....	34
Fig. 24	EL2595 pin assignment	35
Fig. 25	Relationship between user side (commissioning) and installation.....	37
Fig. 26	Control configuration with Embedded PC, input (EL1004) and output (EL2008)	38
Fig. 27	Initial TwinCAT 2 user interface.....	39
Fig. 28	Selection of the target system	40
Fig. 29	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	40
Fig. 30	Select “Scan Devices...”	41
Fig. 31	Automatic detection of I/O devices: selection the devices to be integrated.....	41
Fig. 32	Mapping of the configuration in the TwinCAT 2 System Manager.....	42
Fig. 33	Reading of individual terminals connected to a device.....	42
Fig. 34	TwinCAT PLC Control after startup	43
Fig. 35	Sample program with variables after a compile process (without variable integration)	44
Fig. 36	Appending the TwinCAT PLC Control project	44
Fig. 37	PLC project integrated in the PLC configuration of the System Manager	45
Fig. 38	Creating the links between PLC variables and process objects.....	45
Fig. 39	Selecting PDO of type BOOL	46
Fig. 40	Selecting several PDOs simultaneously: activate “Continuous” and “All types”.....	46
Fig. 41	Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample	47
Fig. 42	Choose target system (remote)	48
Fig. 43	PLC Control logged in, ready for program startup	49

Fig. 44	Initial TwinCAT 3 user interface.....	50
Fig. 45	Create new TwinCAT project.....	50
Fig. 46	New TwinCAT3 project in the project folder explorer	51
Fig. 47	Selection dialog: Choose the target system	51
Fig. 48	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	52
Fig. 49	Select "Scan"	52
Fig. 50	Automatic detection of I/O devices: selection the devices to be integrated.....	53
Fig. 51	Mapping of the configuration in VS shell of the TwinCAT3 environment.....	53
Fig. 52	Reading of individual terminals connected to a device.....	54
Fig. 53	Adding the programming environment in "PLC"	55
Fig. 54	Specifying the name and directory for the PLC programming environment	55
Fig. 55	Initial "Main" program of the standard PLC project.....	56
Fig. 56	Sample program with variables after a compile process (without variable integration).....	57
Fig. 57	Start program compilation.....	57
Fig. 58	Creating the links between PLC variables and process objects	58
Fig. 59	Selecting PDO of type BOOL	58
Fig. 60	Selecting several PDOs simultaneously: activate "Continuous" and "All types"	59
Fig. 61	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample	59
Fig. 62	Creating a PLC data type	60
Fig. 63	Instance_of_struct	60
Fig. 64	Linking the structure	61
Fig. 65	Reading a variable from the structure of the process data	61
Fig. 66	TwinCAT development environment (VS shell): logged-in, after program startup.....	62
Fig. 67	System Manager "Options" (TwinCAT 2).....	63
Fig. 68	Call up under VS Shell (TwinCAT 3)	63
Fig. 69	Overview of network interfaces	64
Fig. 70	EtherCAT device properties(TwinCAT 2): click on "Compatible Devices..." of tab "Adapte"	64
Fig. 71	Windows properties of the network interface.....	65
Fig. 72	Exemplary correct driver setting for the Ethernet port	65
Fig. 73	Incorrect driver settings for the Ethernet port	66
Fig. 74	TCP/IP setting for the Ethernet port	67
Fig. 75	Identifier structure	68
Fig. 76	OnlineDescription information window (TwinCAT 2)	69
Fig. 77	Information window OnlineDescription (TwinCAT 3)	69
Fig. 78	File OnlineDescription.xml created by the System Manager	70
Fig. 79	Indication of an online recorded ESI of EL2521 as an example	70
Fig. 80	Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3).....	70
Fig. 81	Using the ESI Updater (>= TwinCAT 2.11).....	72
Fig. 82	Using the ESI Updater (TwinCAT 3).....	72
Fig. 83	Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)	73
Fig. 84	Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3).....	73
Fig. 85	Selecting the Ethernet port	73
Fig. 86	EtherCAT device properties (TwinCAT 2)	74
Fig. 87	Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3).....	74
Fig. 88	Selection dialog for new EtherCAT device	75
Fig. 89	Display of device revision	75

Fig. 90	Display of previous revisions	76
Fig. 91	Name/revision of the terminal	76
Fig. 92	EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3).....	77
Fig. 93	Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3).....	78
Fig. 94	Scan Devices (left: TwinCAT 2; right: TwinCAT 3).....	78
Fig. 95	Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3).....	78
Fig. 96	Detected Ethernet devices	79
Fig. 97	Example default state	79
Fig. 98	Installing EtherCAT terminal with revision -1018	80
Fig. 99	Detection of EtherCAT terminal with revision -1019	80
Fig. 100	Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)	80
Fig. 101	Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3).....	81
Fig. 102	Scan progressexemplary by TwinCAT 2	81
Fig. 103	Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3).....	81
Fig. 104	Displaying of “Free Run” and “Config Mode” toggling right below in the status bar	81
Fig. 105	TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)	81
Fig. 106	Online display example	82
Fig. 107	Faulty identification	82
Fig. 108	Identical configuration (left: TwinCAT 2; right: TwinCAT 3).....	83
Fig. 109	Correction dialog	83
Fig. 110	Name/revision of the terminal	84
Fig. 111	Correction dialog with modifications	85
Fig. 112	Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3).....	85
Fig. 113	TwinCAT 2 Dialog Change to Alternative Type	85
Fig. 114	Branch element as terminal EL3751	86
Fig. 115	“General” tab.....	86
Fig. 116	“EtherCAT” tab.....	87
Fig. 117	“Process Data” tab.....	88
Fig. 118	Configuring the process data	89
Fig. 119	“Startup” tab.....	90
Fig. 120	“CoE - Online” tab.....	91
Fig. 121	Dialog “Advanced settings”	92
Fig. 122	“Online” tab	92
Fig. 123	“DC” tab (Distributed Clocks).....	93
Fig. 124	Representation of warning thresholds	101
Fig. 125	Selection of the mode of operation	102
Fig. 126	Activating PDO	103
Fig. 127	Selection of the diagnostic information of an EtherCAT Slave	104
Fig. 128	Basic EtherCAT Slave Diagnosis in the PLC.....	105
Fig. 129	EL3102, CoE directory	107
Fig. 130	Example of commissioning aid for a EL3204	108
Fig. 131	Default behaviour of the System Manager	109
Fig. 132	Default target state in the Slave	109
Fig. 133	PLC function blocks	110

Fig. 134 Illegally exceeding the E-Bus current 111

Fig. 135 Warning message for exceeding E-Bus current 111

Fig. 136 All PDOs for the synchronous operation..... 112

Fig. 137 Normal digital output..... 113

Fig. 138 Digital output with specification of the pulse length 114

Fig. 139 Digital output with external trigger and specification of pulse length and trigger delay 115

Fig. 140 "Process data" tab 117

Fig. 141 TwinCAT System Manager with the PDO selection 117

Fig. 142 Standard EL2595 process image 118

Fig. 143 Additional information data 119

Fig. 144 Additional process data in the "External trigger input" mode..... 120

Fig. 145 Additional process data for Distributed Clocks mode 121

Fig. 146 Process Data tab SM2, EL2595 (default) 122

Fig. 147 Process Data tab SM3, EL2595 (default) 123

Fig. 148 EL2595 DC schematic 125

Fig. 149 EL2595 DC settings..... 126

Fig. 150 Activation of the master distributed clock display 127

Fig. 151 Extended process image of the EtherCAT master 128

Fig. 152 Sequence of a switching operation with activated DistributedClock..... 129

Fig. 153 CoE configuration data 142

Fig. 154 CoE - diagnostic data 144

Fig. 155 CoE command object 145

Fig. 156 DiagMessages in the CoE 147

Fig. 157 Implementation of the DiagMessage system in the TwinCAT System Manager 148

Fig. 158 Startup List..... 148

Fig. 159 Device identifier consisting of name EL3204-0000 and revision -0016 160

Fig. 160 Scan the subordinate field by right-clicking on the EtherCAT device 161

Fig. 161 Configuration is identical 161

Fig. 162 Change dialog 161

Fig. 163 EEPROM Update 162

Fig. 164 Selecting the new ESI..... 162

Fig. 165 Display of EL3204 firmware version 163

Fig. 166 Firmware Update 164

Fig. 167 FPGA firmware version definition 166

Fig. 168 Context menu Properties 166

Fig. 169 Dialog Advanced Settings 167

Fig. 170 Multiple selection and firmware update 169

Fig. 171 Selecting the Restore default parameters PDO..... 170

Fig. 172 Entering a restore value in the Set Value dialog..... 170

More Information:

[www.beckhoff.com/en-us/products/i-o/ethercat-terminals/
el2xxx-digital-output/el2595.html](http://www.beckhoff.com/en-us/products/i-o/ethercat-terminals/el2xxx-digital-output/el2595.html)

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

