



Documentation

EL3783

EL3783 - Power monitoring oversampling terminal for 690 V

Version: 1.1
Date: 2019-05-10

BECKHOFF

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
1.5	Start	12
2	Product overview	13
2.1	Introduction	13
2.2	Technical data	14
2.3	Technology	15
3	Basics communication	17
3.1	EtherCAT basics.....	17
3.2	EtherCAT cabling – wire-bound.....	17
3.3	General notes for setting the watchdog.....	18
3.4	EtherCAT State Machine	20
3.5	CoE Interface.....	22
3.6	Distributed Clock	27
4	Mounting and wiring	28
4.1	Instructions for ESD protection	28
4.2	Installation on mounting rails	29
4.3	Connection system	32
4.4	Positioning of passive Terminals	36
4.5	Prescribed installation position	37
4.6	LEDs and connection	38
5	Commissioning	42
5.1	TwinACT	42
5.1.1	TwinCAT Quick Start	42
5.1.2	TwinCAT Development Environment.....	67
5.2	General Notes - EtherCAT Slave Application	99
5.3	Basic function principles	107
5.4	Oversampling terminals/boxes and TwinCAT Scope	111
5.4.1	TwinCAT 3 procedure.....	112
5.4.2	TwinCAT 2 procedure.....	120
5.5	Process data.....	129
5.6	Object description and parameterization	138
5.6.1	Restore object.....	138
5.6.2	Configuration data	139
5.6.3	Input data	140
5.6.4	Information and diagnostic data.....	140
5.6.5	Standard objects.....	141
5.7	Notices on analog specifications	147
5.7.1	Full scale value (FSV).....	147
5.7.2	Measuring error/ measurement deviation	148

5.7.3	Temperature coefficient tK [ppm/K]	148
5.7.4	Single-ended/differential typification	150
5.7.5	Common-mode voltage and reference ground (based on differential inputs).....	155
5.7.6	Dielectric strength	155
5.7.7	Temporal aspects of analog/digital conversion.....	156
6	Diagnosis	160
6.1	Diagnostics – basic principles of diag messages	160
7	Appendix	170
7.1	Firmware Update EL/ES/EM/ELM/EPxxxx	170
7.1.1	Device description ESI file/XML.....	171
7.1.2	Firmware explanation	174
7.1.3	Updating controller firmware *.efw.....	175
7.1.4	FPGA firmware *.rbf.....	176
7.1.5	Simultaneous updating of several EtherCAT devices.....	180
7.2	Firmware compatibility	181
7.3	Restoring the delivery state	181
7.4	Support and Service	182

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 P®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® 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, DE102004044764, DE102007017835 with corresponding applications or registrations in various other countries.

The TwinCAT Technology is covered, including but not limited to the following patent applications and patents: EP0851348, US6167425 with corresponding applications or registrations in various other countries.



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
1.1	<ul style="list-style-type: none"> • Update Example program • Update revision status • Update structure
1.0	<ul style="list-style-type: none"> • Addenda & corrections • 1st public issue
0.8	<ul style="list-style-type: none"> • Addenda & corrections
0.7	<ul style="list-style-type: none"> • Update object description
0.6	<ul style="list-style-type: none"> • Addenda & corrections
0.5	<ul style="list-style-type: none"> • Addenda & corrections
0.4	<ul style="list-style-type: none"> • Addenda & corrections
0.3	<ul style="list-style-type: none"> • Addenda & corrections
0.2	<ul style="list-style-type: none"> • Addenda & corrections
0.1	<ul style="list-style-type: none"> • preliminary documentation for EL3783

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 12 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

Unique serial number/ID, ID number

In addition, in some series each individual module has its own unique serial number.

See also the further documentation in the area

- IP67: [EtherCAT Box](#)
- Safety: [TwinSafe](#)
- Terminals with factory calibration certificate and other measuring terminals

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: CU2016 switch with serial/ batch number

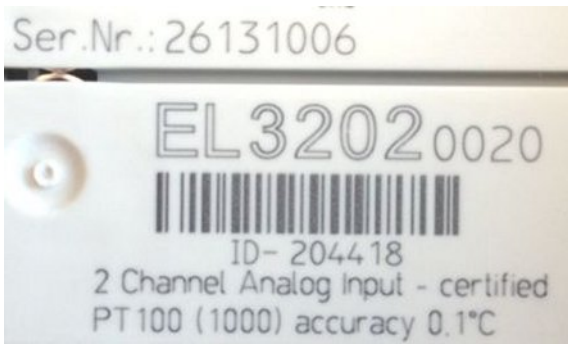


Fig. 4: EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418

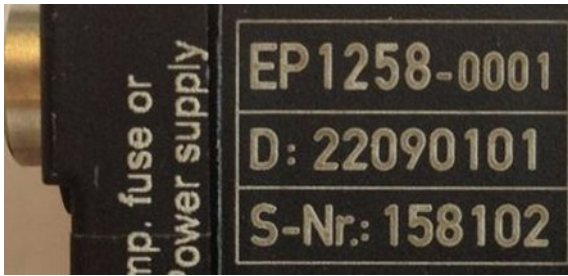


Fig. 5: EP1258-00001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102

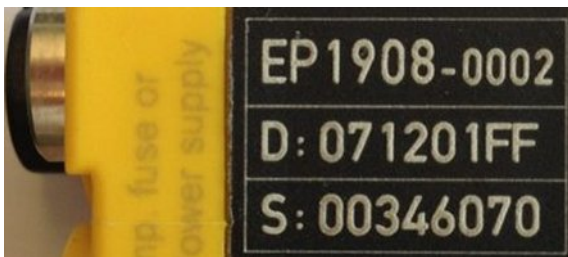


Fig. 6: EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070



Fig. 7: EL2904 IP20 safety terminal with batch number/ date code 50110302 and unique serial number 00331701



Fig. 8: ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201

1.5 Start

For commissioning:

- Install the EL3783 as described in section [Mounting and wiring \[► 28\]](#).
- configure the EL3783 in TwinCAT as described in the chapter Commissioning.

2 Product overview

2.1 Introduction

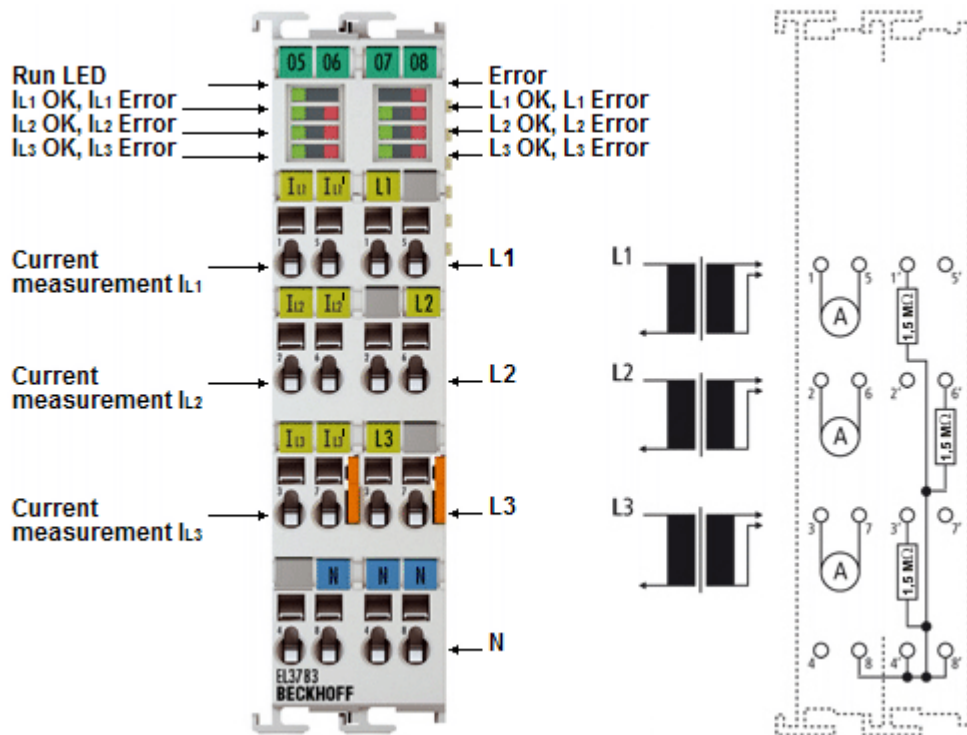


Fig. 9: EL3783

Power monitoring oversampling terminal for 690 V

The EL3783 EtherCAT Terminal is designed as a power monitoring terminal for the state monitoring of a 3-phase AC voltage system (rated mains voltage 400/690 V_{rms}). For each phase voltages up to 520 V_{rms} and currents up to 6.5 A_{rms} are sampled as instantaneous values with a resolution of 16 bit.

The six channels are measured simultaneously based on the EtherCAT oversampling principle with a temporal resolution of 50 μs and passed on to the controller. The controller has sufficient computing power for true RMS or performance calculation and complex custom algorithms based on the measured voltages and currents.

Through the oversampling principle the terminal is able to measure at significantly shorter intervals than the cycle time of the controller. AC parameters must be connected and measured with a common reference potential.

The EL3783 supports distributed clocks for measuring synchronously with other EtherCAT devices, but can also be operated without distributed clocks.

i RMS value (rms) specifications

All AC value specifications in this documentation such as RMS specifications (rms) refer to a 50/60 Hz 3-phase mains network with a sinusoidal waveform (crest factor 1.414).

Also see about this

- 📖 Technology [▶ 15]
- 📖 Basic function principles [▶ 107]

2.2 Technical data

i RMS value (rms) specifications

All AC value specifications in this documentation such as RMS specifications (rms) refer to a 50/60 Hz 3-phase mains network with a sinusoidal waveform (crest factor 1.414).

Technical data	EL3783
Number of inputs	3 x current, 3 x voltage
Oversampling factor	n = 1...100 selectable
Distributed Clocks	yes
Accuracy of Distributed Clocks	<< 1 μ s
Measuring error	< $\pm 0.2\%$ of full scale value
Conversion time	Minimum 50 μ s, all channels measured simultaneously
Mains voltage (nominal voltage range)	AC accordingly: 400 V _{rms} (UL _x -N) or 690 V _{rms} (UL _x -UL _y) (TN network: 600 V _{rms})
Technical voltage measuring range	520 V _{rms} (UL _x -N) or 897 V _{rms} (UL _x -UL _y) common reference potential N/GND max. time for voltages exceeding 500 V _{rms} (UL _x -N) or 863 V _{rms} (UL _x -UL _y): t _{max} < 10 s *
Max. permitted overvoltage	max. ± 736 V (peak value, UL _x -N, equivalent to 520 V _{rms}) or max. ± 1270 V (peak value, UL _x -UL _y , equivalent to 897 V _{rms})*
Voltage resolution (16-bit including sign)	1 digit ~ 22.5 mV
Input resistance voltage circuit	typ. 1.5 M Ω
Nominal current range	AC accordingly: 1 A _{rms} (automatic switching to 5 A _{rms}) recommended via measuring transformer x A AC/1 A AC
Technical current measuring range	1.84 A (peak value, equivalent to 1.3 A _{rms}) or 9.2 A (peak value, equivalent to 6.5 A _{rms})
Max. permitted overcurrent	max. ± 10 A peak value, equivalent to 7 A _{rms} *, surge overload capacity 60 A (sinusoidal) for 1 sec., the upstream use of current-limiting current transformers is recommended
Largest short-term deviation during a specified electrical interference test	< $\pm 0.5\%$ of full scale value for the current measurement
Current resolution (16-bit including sign)	1 digit ~ 56.2 μ A (~281 μ A if 5 A range is active)
Input resistance current circuit	typ. 3 m Ω
Signal type	variable
Electrical isolation	4,500 V (test voltage connecting terminal/E-bus), production test 2 sec.
Current consumption power contacts	-
Current consumption via E-bus	typ. 260 mA
Special features	Technical measuring range for current and voltage 130% of the respective measuring range, Anti-aliasing filter (see Basic functional principles), adjustable automatic current range switching, electrically isolated current inputs
Configuration	via TwinCAT System Manager
Weight	approx. 100 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. 27 mm x 100 mm x 70 mm (width aligned: 24 mm)
Mounting [► 29]	on 35 mm mounting rail conforms 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	see notice [► 37] !
Approval	CE

* operation for longer periods above the rated range can impair the function and/or lead to a shortening of the service life

2.3 Technology

Measurement of AC variables with the EL3783

Normal three-phase mains supply

The normal 3-phase low-voltage grids in Central Europe are characterized by the following interrelationship: rated voltage is usually the effective voltage U_{RMS} , e.g. $230 V_{RMS}$, as a star voltage between one of the three live conductors (L1, L2 or L3) and neutral N.

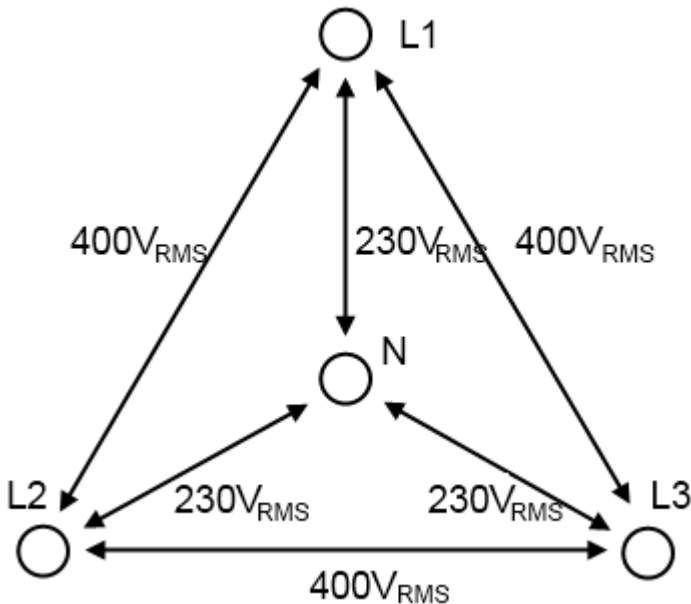


Fig. 10: Voltages in three-phase mains supply of Central Europe

With a pure sinusoidal oscillation (unloaded grid) a maximum peak voltage (max. amplitude) of approx.

$\pm 325V_{peak}$ to N is calculated from the then valid crest factor from $\sqrt{2}$ in the star voltage.

A phase-to-phase voltage of $230V \cdot \sqrt{3} \approx 400V$ (RMS) can be measured between the live conductors.

RMS value specifications

● RMS value ($_{rms}$) specifications

i All AC value specifications in this documentation such as RMS specifications ($_{rms}$) refer to a 50/60 Hz 3-phase mains network with a sinusoidal waveform.

The EL3783 EtherCAT Terminal is a power monitoring terminal for state monitoring of a 3-phase AC voltage system. The following properties are characteristic of the EL3783:

- 3 channels measure $-737...+737 V$ through analog-to-digital converters in 16-bit resolution as amplitude value
- 3 channels measure $-9.2...+9.2 A$ through analog-to-digital converters in 16-bit resolution as amplitude value
- all 6 analog input channels are measured simultaneously
- the EL3783 is an oversampling terminal and can therefore record not just 1, but up to 100 samples per channel in each PLC/EtherCAT cycle. These are sent as a data packet to the controller via the cyclic process data.
The minimum sampling time is $50 \mu s$, corresponding to 20,000 samples/second.

- The voltage and current curve can have any form; the EL3783 is thus suitable only for AC measurements
- the EL3783 can be synchronized with other EtherCAT device over Distributed Clocks, but can also be operated without Distributed Clocks with oversampling
- no pre-evaluations or calculations of the amplitude values take place in the EL3783

Hence, the EL3783 is suitable for very different applications, for example

- 3-phase monitoring: the voltage and current to N are measured for a 3-phase load
- 3-load monitoring: the voltage and current to N can be measured for 3 loads even if connected to the same phase
- each channel can measure as desired, provided the measured is referred to N
- Measurement of non-sinusoidal amplitude curves, including rectangular

Evaluations and calculations of the raw data sent to the controller, such as active power (P), cumulative power consumption (W) or power factor ($\cos \varphi$) must take place in the controller. The controller has sufficient computing power for true RMS or performance calculation and complex custom algorithms based on the measured voltages and currents.

● **Design of voltage measuring range**

i Low-voltage power supply systems are defined, for example, in IEC 60038. Since 2003 the specification here is $230/400 V_{RMS} \pm 10\%$, corresponding to $\pm 357 V_{peak}$. So that substantial spikes can also be measured, the EL3783 supports a measuring range of $520 V_{RMS} AC$, corresponding to $\pm 737 V_{peak}$.

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.

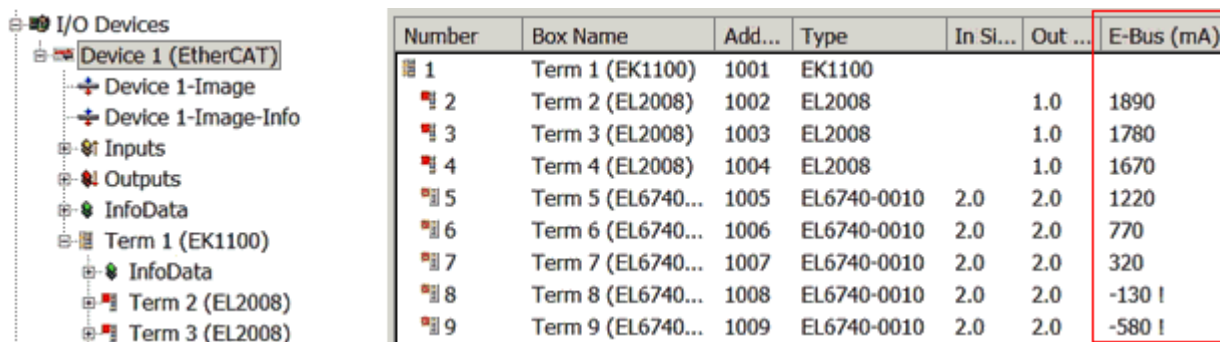
● Recommended cables

i 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.



Number	Box Name	Add...	Type	In Si...	Out ...	E-Bus (mA)
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL2008)	1002	EL2008		1.0	1890
3	Term 3 (EL2008)	1003	EL2008		1.0	1780
4	Term 4 (EL2008)	1004	EL2008		1.0	1670
5	Term 5 (EL6740...)	1005	EL6740-0010	2.0	2.0	1220
6	Term 6 (EL6740...)	1006	EL6740-0010	2.0	2.0	770
7	Term 7 (EL6740...)	1007	EL6740-0010	2.0	2.0	320
8	Term 8 (EL6740...)	1008	EL6740-0010	2.0	2.0	-130 I
9	Term 9 (EL6740...)	1009	EL6740-0010	2.0	2.0	-580 I

Fig. 11: 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 2 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.

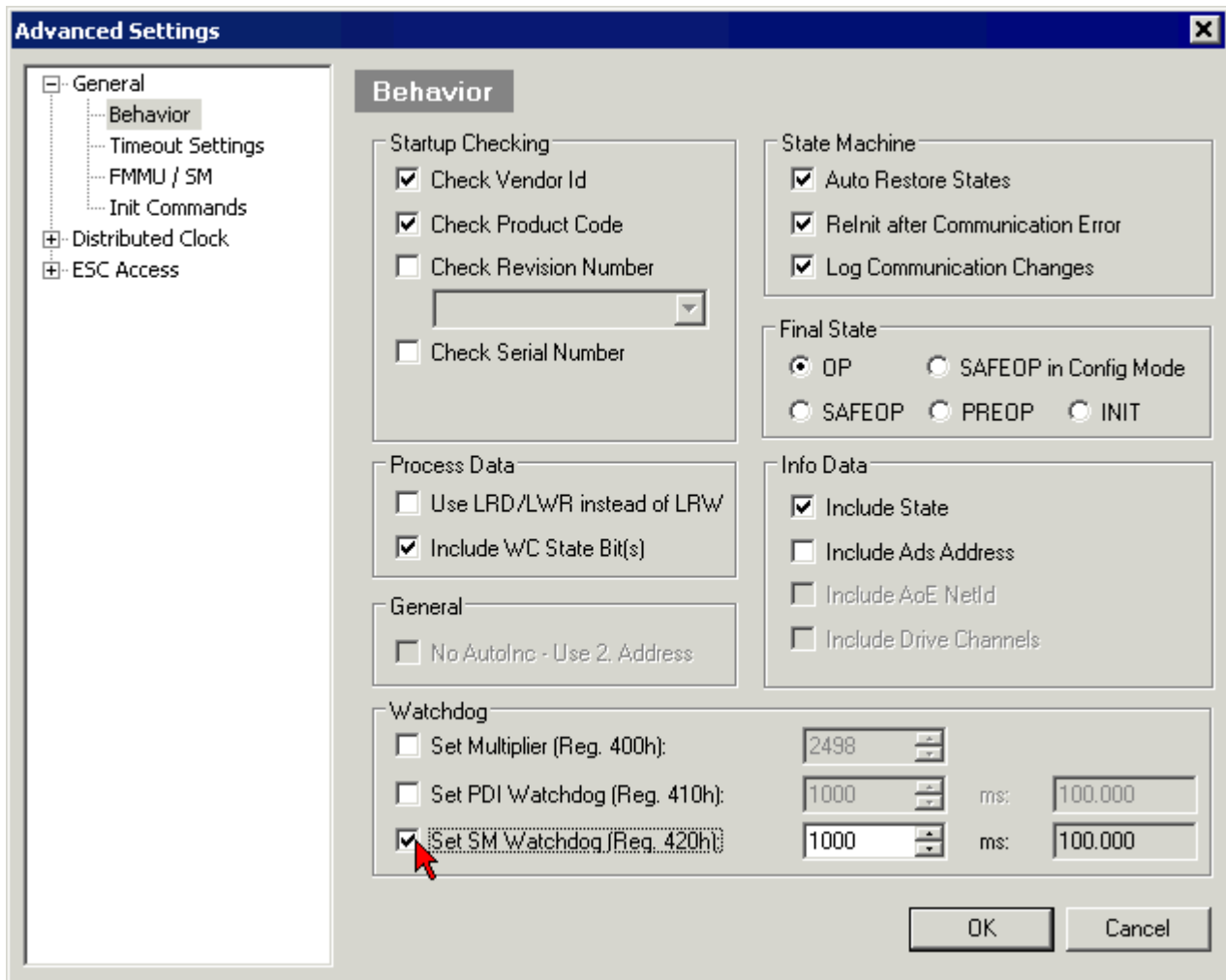


Fig. 12: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

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.

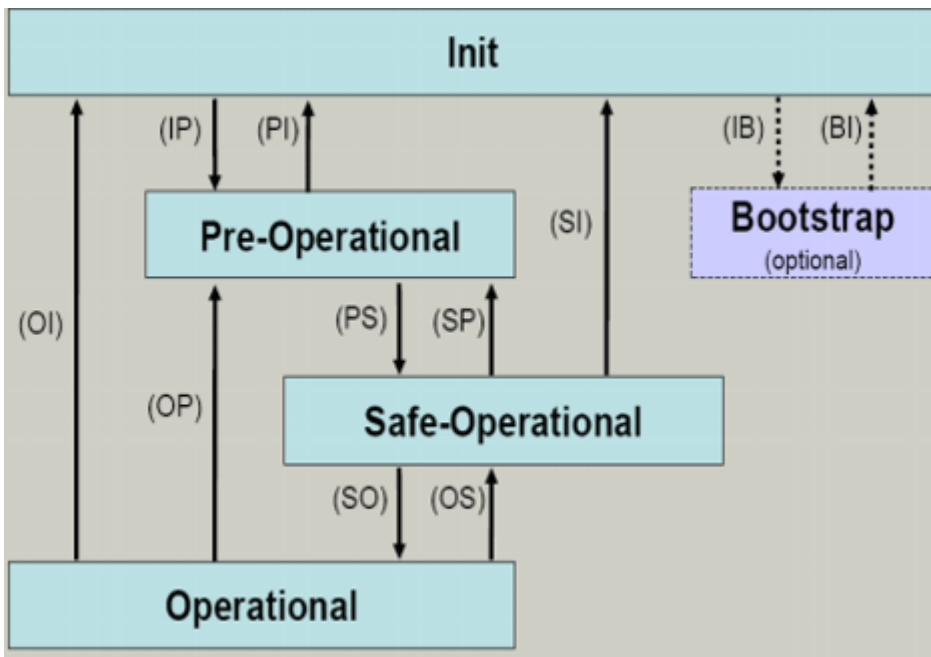


Fig. 13: States of the EtherCAT State Machine

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` [▶ 18] 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 (CANopen 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 2 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 "x" 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: In some EtherCAT devices the channel parameters are stored here (as an alternative to the 0x8000 range).
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

● Availability



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

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

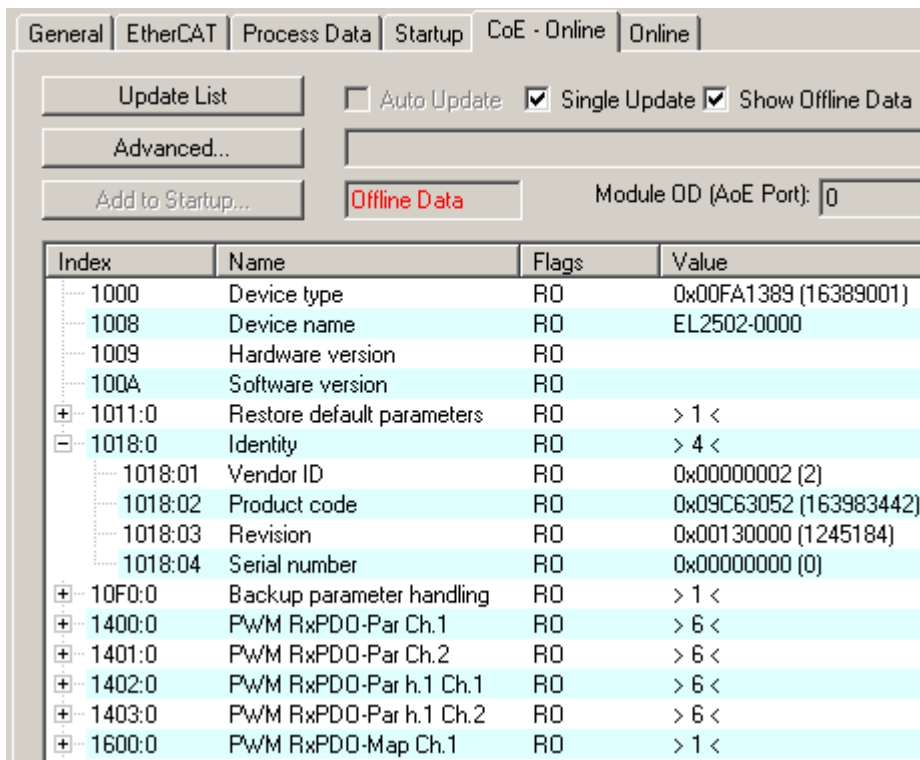


Fig. 14: "CoE Online " tab

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 parameterised 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.

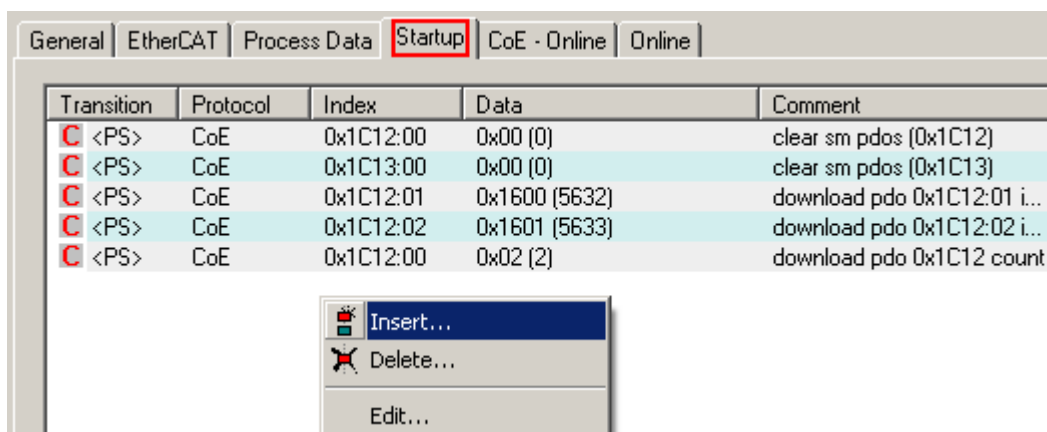


Fig. 15: Startup list in the TwinCAT System Manager

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.

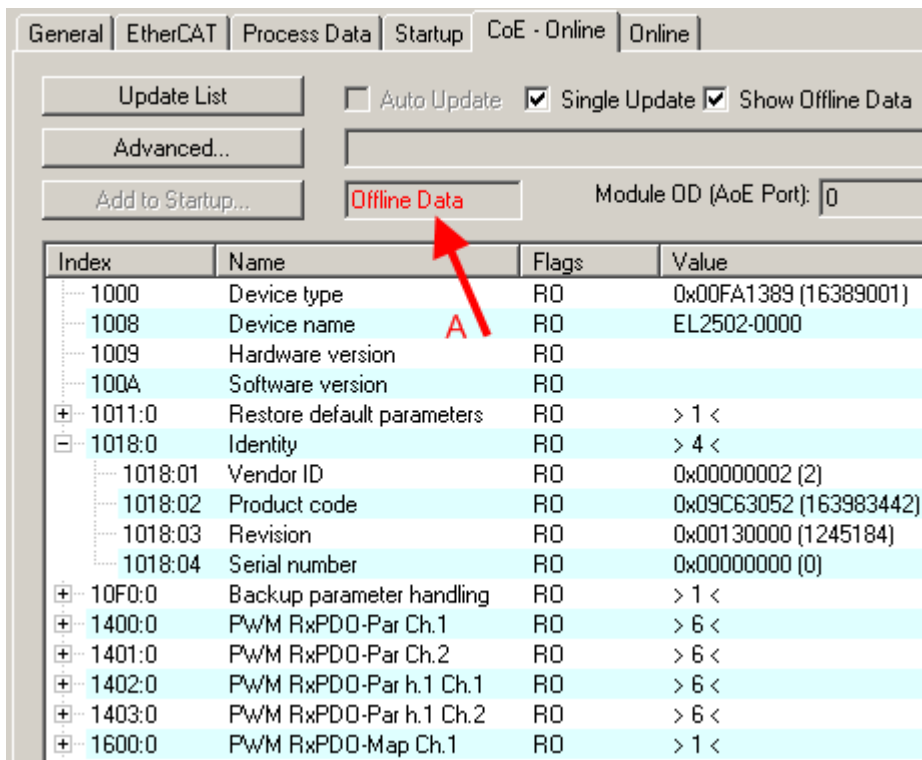


Fig. 16: Offline list

- 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.

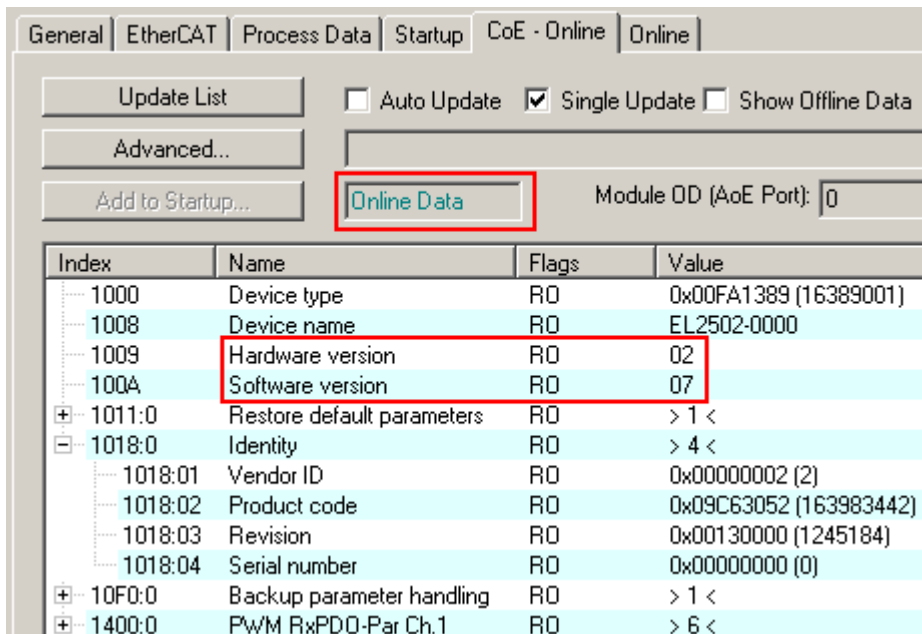


Fig. 17: Online list

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 4 logical channels and therefore 4 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 Instructions for ESD protection

NOTE

Destruction of the devices by electrostatic discharge possible!

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

- Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly.
- Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.).
- Surroundings (working place, packaging and personnel) should be grounded probably, when handling with the devices.
- Each assembly must be terminated at the right hand end with an [EL9011](#) or [EL9012](#) bus end cap, to ensure the protection class and ESD protection.

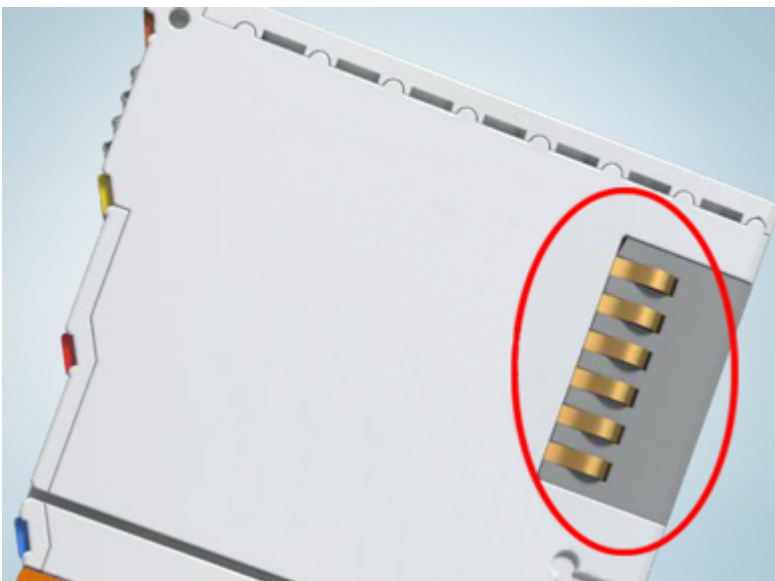


Fig. 18: Spring contacts of the Beckhoff I/O components

4.2 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

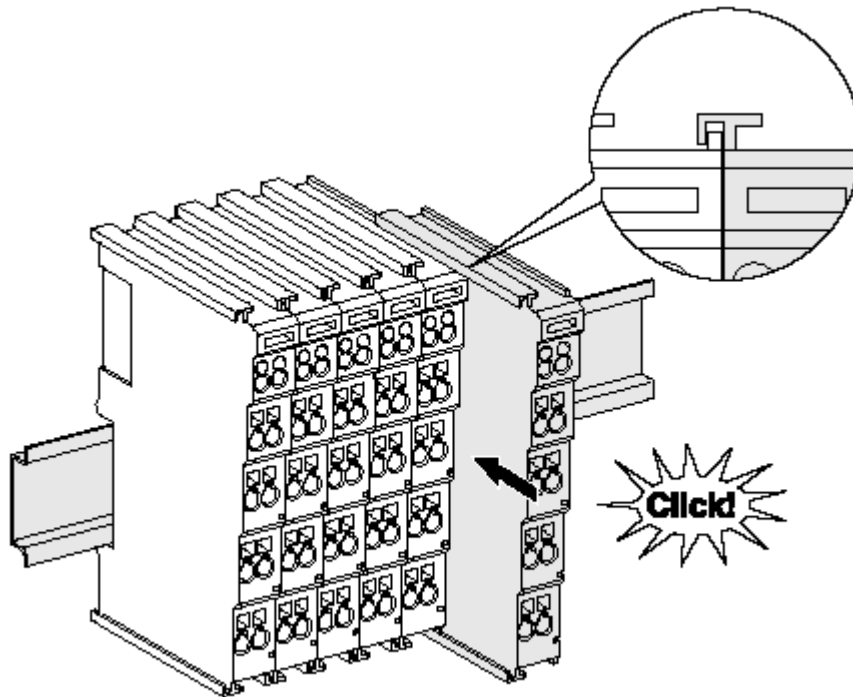


Fig. 19: Attaching on mounting rail

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

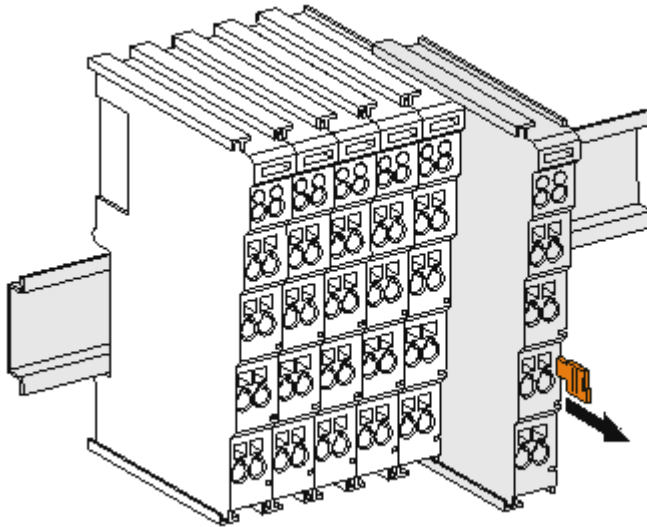


Fig. 20: Disassembling of terminal

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.

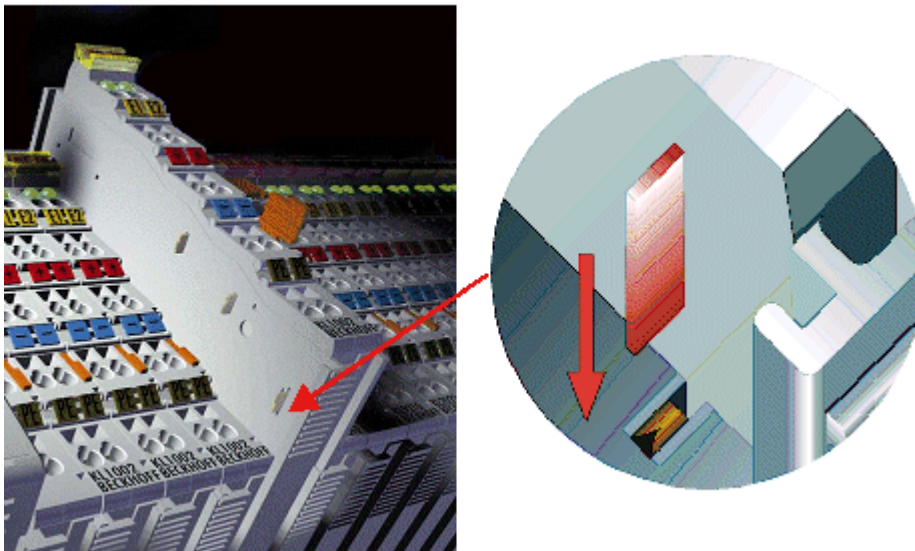


Fig. 21: Power contact on left side

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!

4.3 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. 22: 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. 23: 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. 24: 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 (HD) 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 \[► 34\]](#) below!

Wiring

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

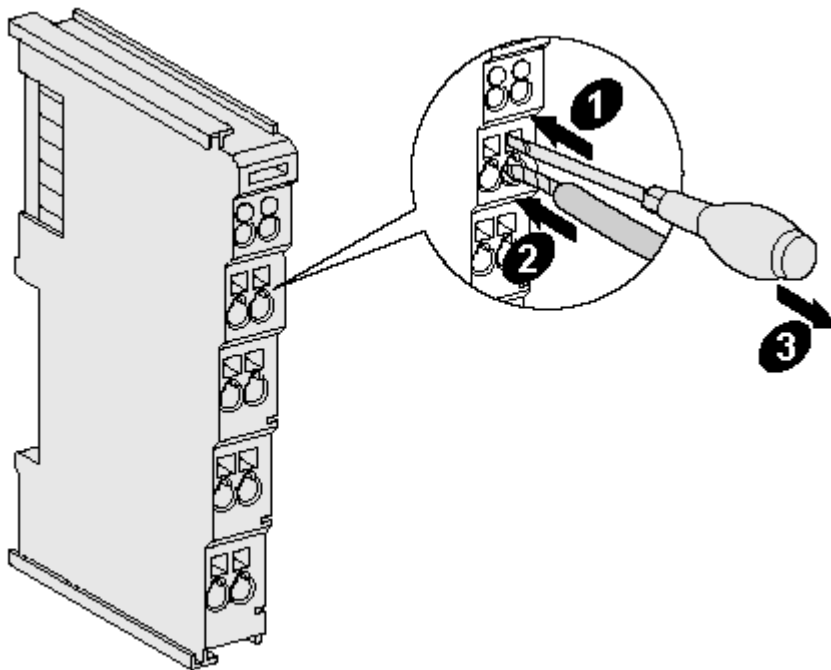


Fig. 25: Mounting a cable on a terminal connection

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 [►_33!])
Wire stripping length	8 ... 9 mm

Shielding**Shielding**

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

4.4 Positioning of passive Terminals

Hint for positioning of passive terminals in the bus terminal block



EtherCAT Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called passive terminals. The passive terminals have no current consumption out of the E-Bus.

To ensure an optimal data transfer, you must not directly string together more than 2 passive terminals!

Examples for positioning of passive terminals (highlighted)

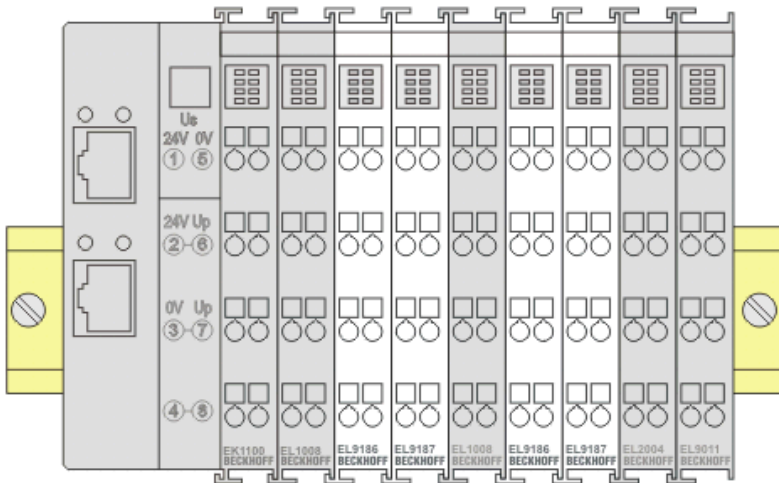


Fig. 26: Correct positioning

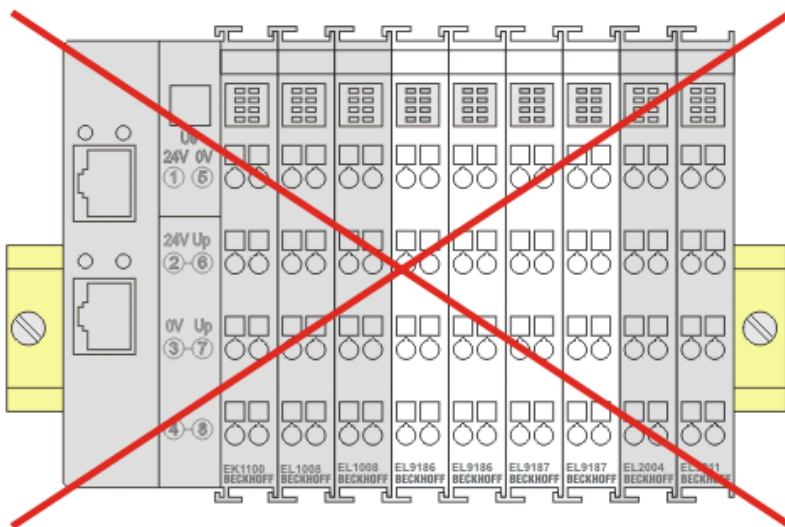


Fig. 27: Incorrect positioning

4.5 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.

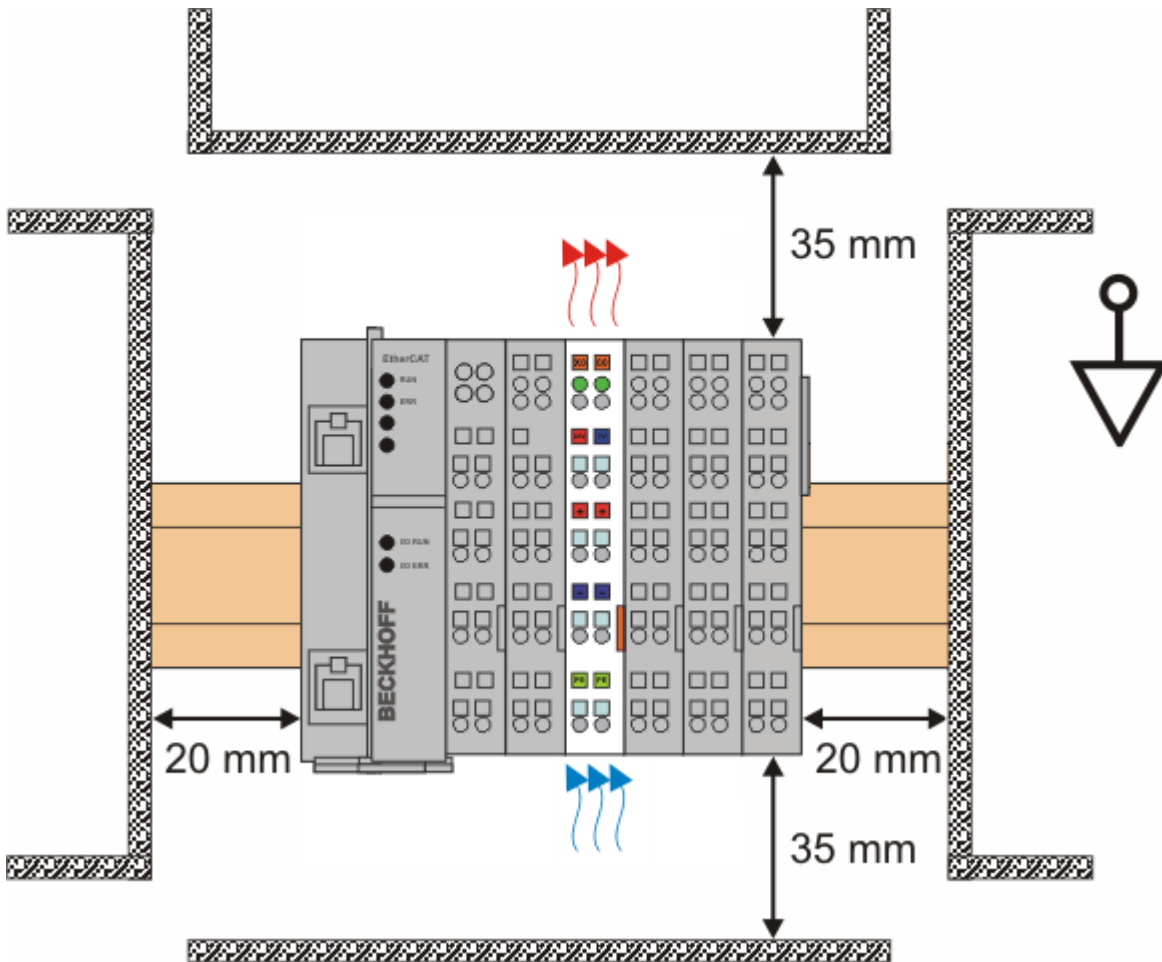


Fig. 28: Recommended minimum distances for standard installation position

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

4.6 LEDs and connection

⚠ WARNING

Do not operate current transformers in no-load mode!

Please note that many manufacturers do not permit their current transformers to be operated in no-load mode! Connect the EL3783 to the secondary windings of the current transformers before using the current transformer!

⚠ WARNING

Earthing of the terminal point N when measuring current!

If you do not connect the terminal point N with the neutral conductor of your mains supply (e.g. if the EL3783 is used purely for current measurements), terminal point N should be earthed, in order to avoid dangerous overvoltages in the event of a current transformer fault!

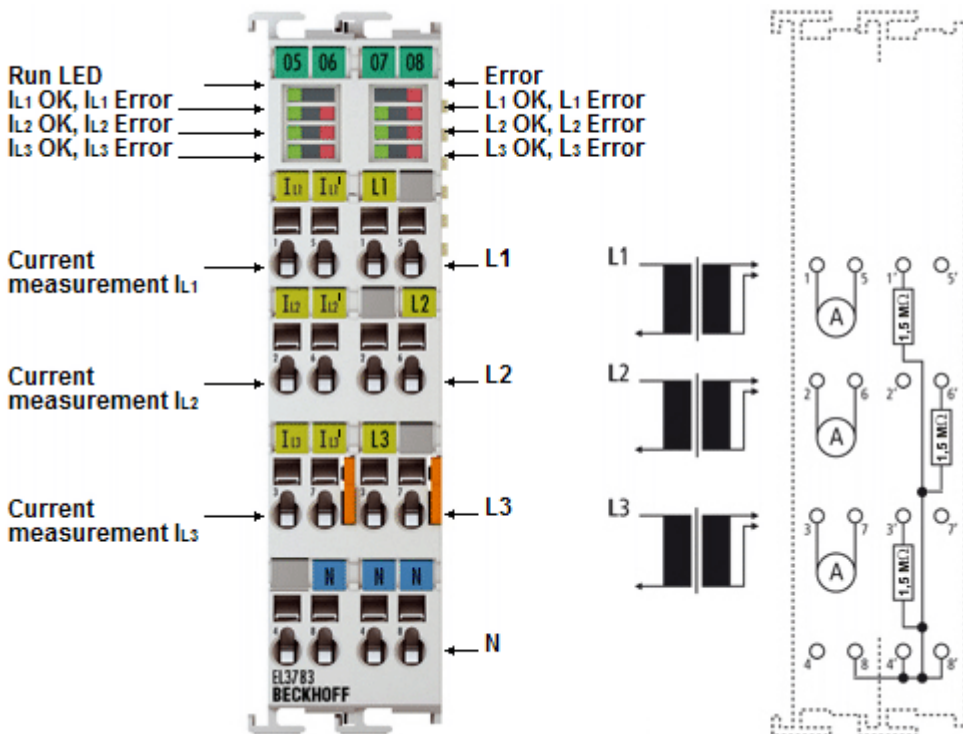


Fig. 29: EL3783

Terminal point		Description
Name	No.	
IL1	1	Phase L1 current measurement input
IL2	2	Phase L2 current measurement input
IL3	3	Phase L3 current measurement input
	4	n.c.
IL1'	5	Phase L1 current measurement output
IL2'	6	Phase L2 current measurement output
IL3'	7	Phase L3 current measurement output
N	8	Neutral conductor

Terminal point		Description
Name	No.	
L1	1'	Phase L1 voltage measurement input
	2'	n.c.
L3	3'	Phase L3 voltage measurement input
	4'	Neutral conductor
N	5'	n.c.
	6'	Phase L2 voltage measurement input
L2	7'	n.c.
	8'	Neutral conductor

LEDs

LED	Color	Meaning	
RUN	green		This LED indicates the terminal's operating state:
		off	State of the EtherCAT State Machine [▶ 20]: INIT = initialization of the terminal or BOOTSTRAP = function for <u>firmware updates</u> [▶ 170] of the terminal
		flashing	State of the EtherCAT State Machine: PREOP = function for mailbox communication and different standard-settings set
		Single flash	State of the EtherCAT State Machine: SAFEOP = verification of the <u>Sync Manager</u> [▶ 91] 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
IL1 OK	green	on	Current IL1 ok
IL1 Error	red	on	Overcurrent on L1. Current > 1.8 A (with 1 A measuring range), equivalent to 1.3 Arms Current > 9.2 A (with 5 A measuring range), equivalent to 6.5 Arms
IL2 OK	green	on	Current IL2 ok
IL2 Error	red	on	Overcurrent on L2. Current > 1.8 A (with 1 A measuring range), equivalent to 1.3 Arms Current > 9.2 A (with 5 A measuring range), equivalent to 6.5 Arms
IL3 OK	green	on	Current IL3 ok
IL3 Error	red	on	Overcurrent on L3. Current > 1.8 A (with 1 A measuring range), equivalent to 1.3 Arms Current > 9.2 A (with 5 A measuring range), equivalent to 6.5 Arms
L1 OK	green	on	Voltage on L1 OK
L1 Error	red	on	Overvoltage on L1. Voltage > 737 V (L1-N)
L2 OK	green	on	Voltage on L2 OK
L2 error	red	on	Overvoltage on L2. Voltage > 737 V (L2-N)
L3 OK	green	on	Voltage on L3 OK
L3 error	red	on	Overvoltage on L3. Voltage > 737 V (L3-N)
Error	red	on	Other errors (logical OR conjunction)

Wiring and measurement methods

Examples of wiring for measurements in the three-phase grid

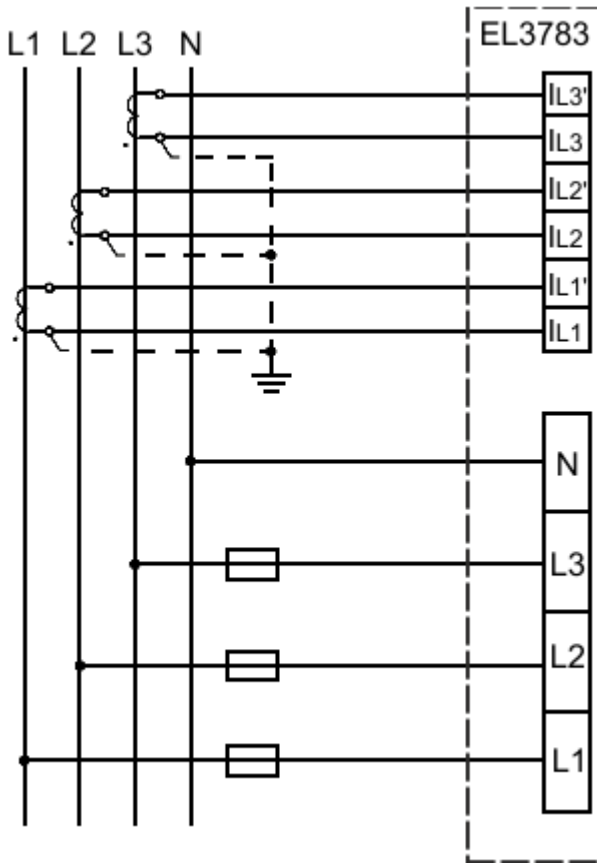


Fig. 30: Measurement in a 3-phase, 4-conductor grid, asymmetrical load

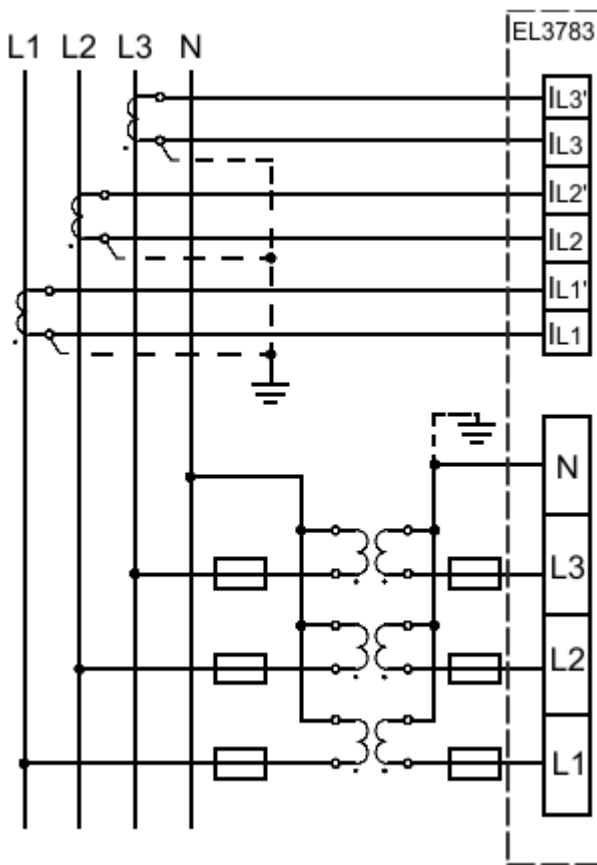


Fig. 31: Measurement in a 3-phase, 4-conductor grid with 3 voltage transformers, asymmetrical load

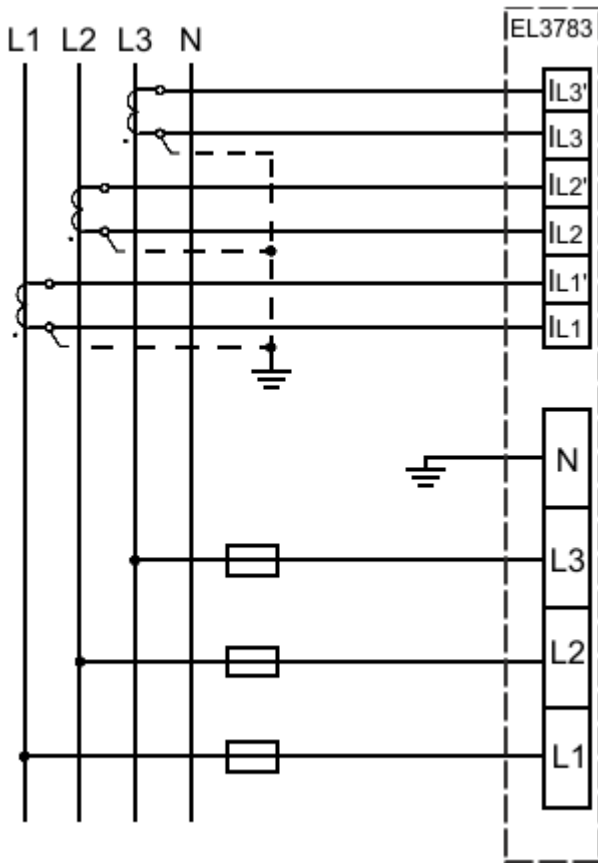


Fig. 32: Measurement in a 3-phase, 3-conductor grid, asymmetrical load

⚠ CAUTION

Risk of device damage!

The user must ensure proper fusing of the voltage measurement path in accordance with the measured variable as well as adequate transient protection!

5 Commissioning

5.1 TwinACT

5.1.1 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:

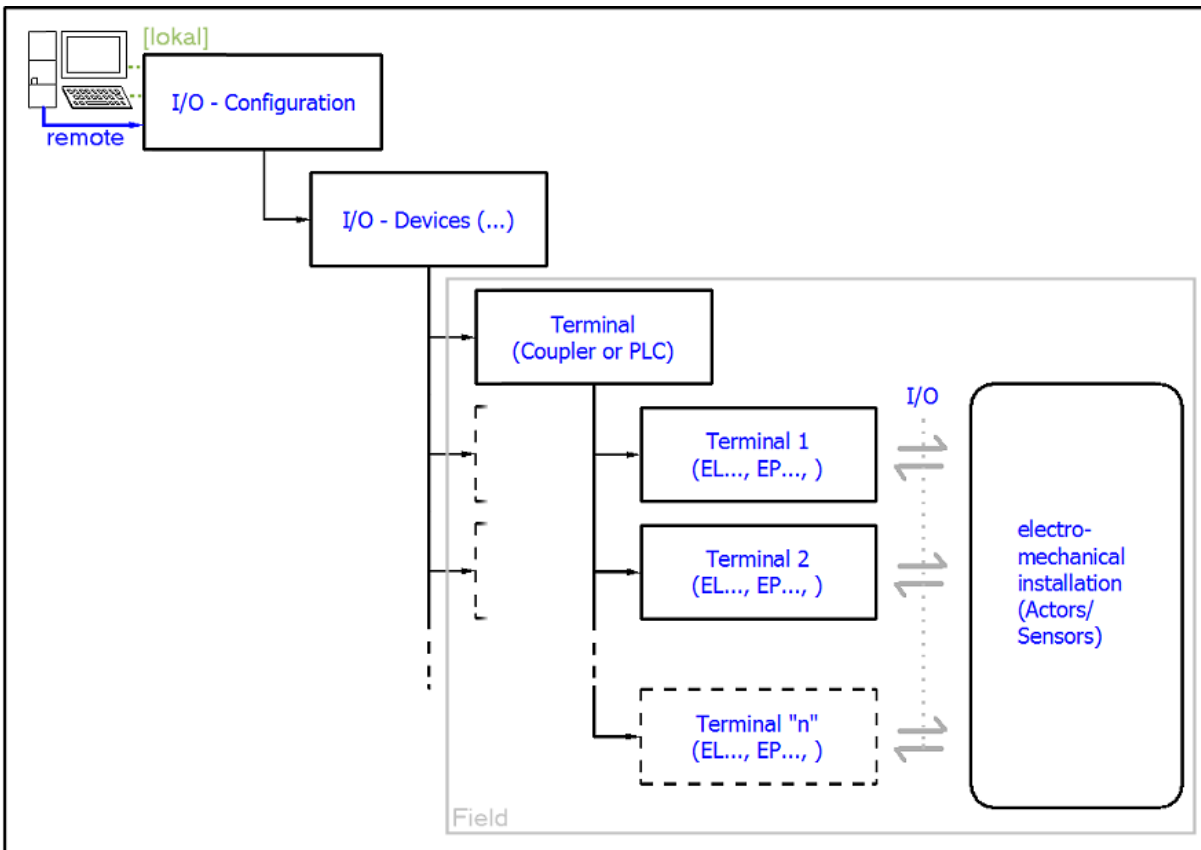


Fig. 33: Relationship between user side (commissioning) and installation

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 analog input terminal -10...+10 V)
- 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)

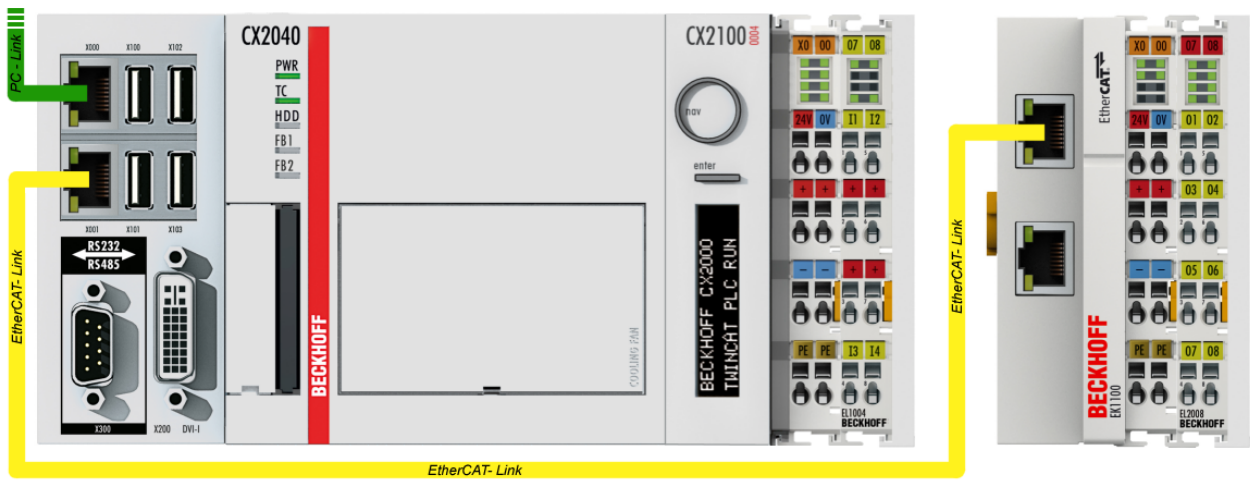


Fig. 34: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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.

5.1.1.1 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:

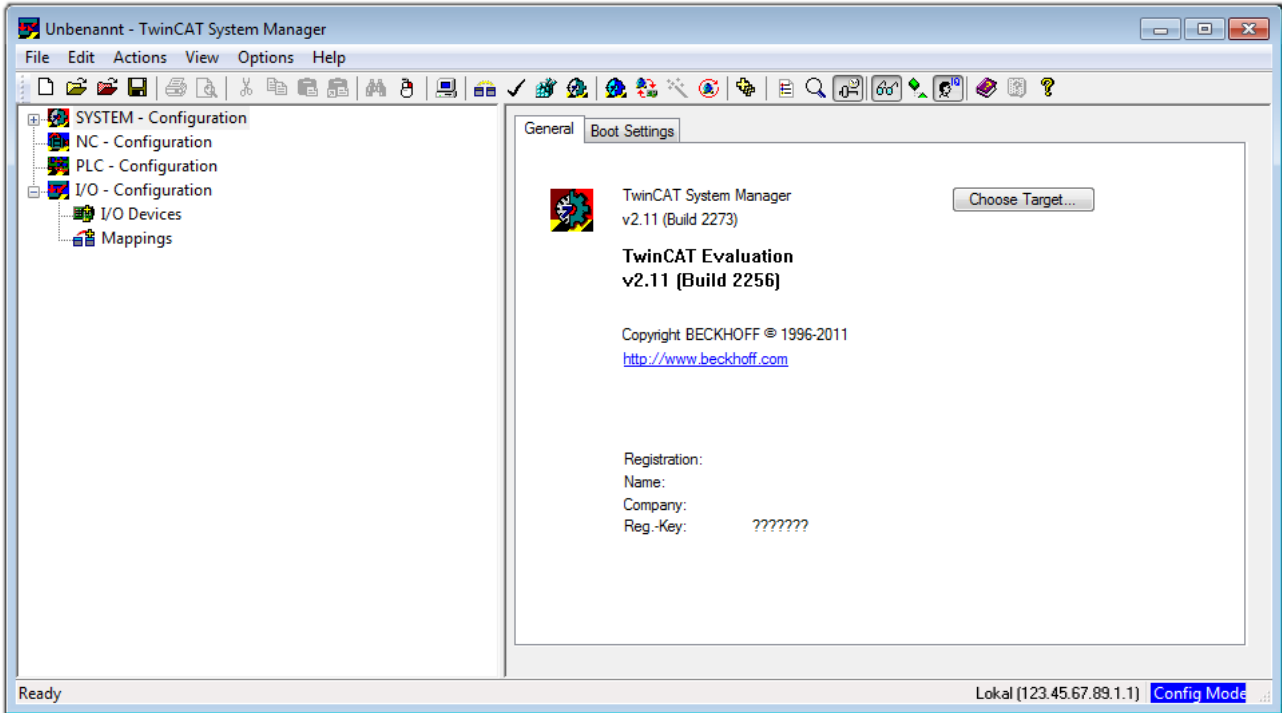



Fig. 35: Initial TwinCAT 2 user interface

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 [▶ 47]".

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:

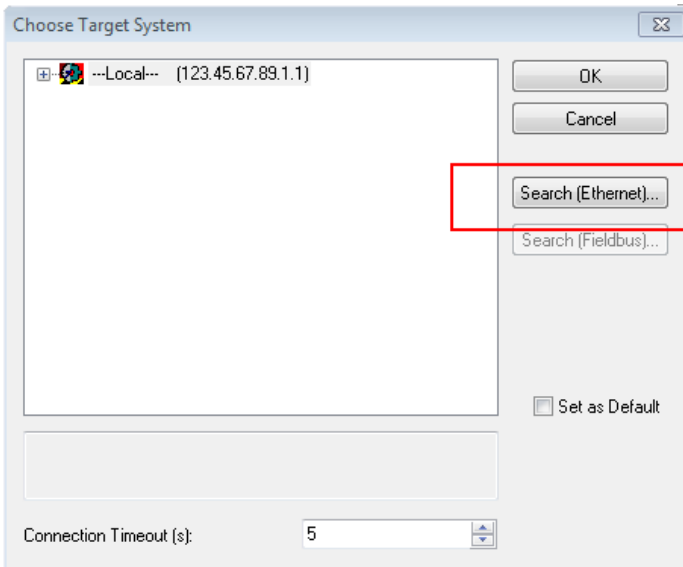


Fig. 36: Selection of the target system

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.

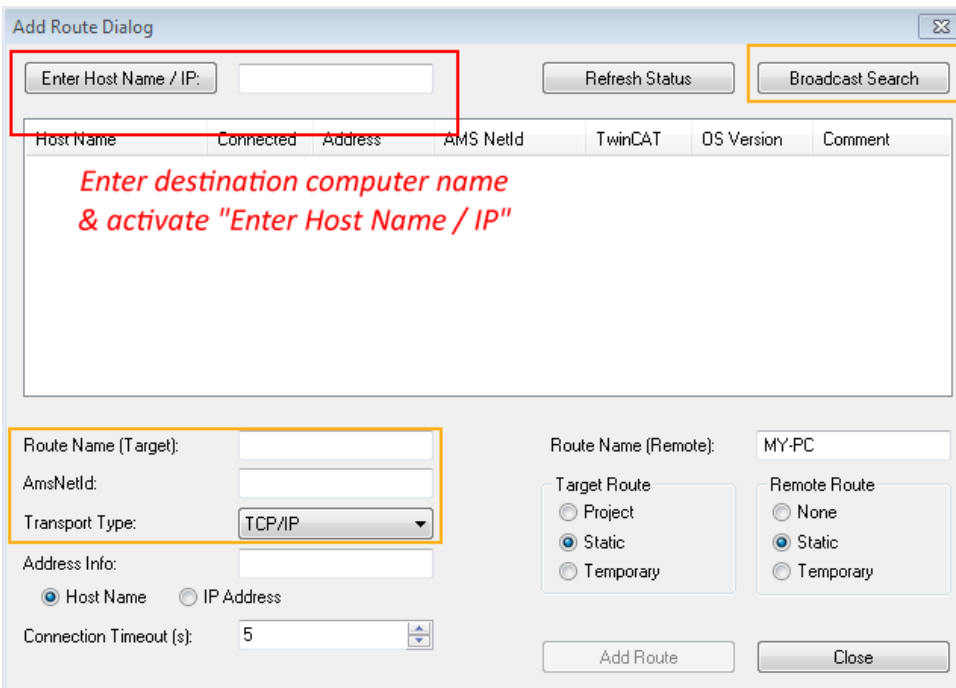
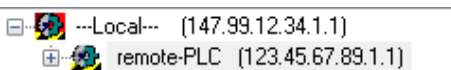


Fig. 37: Specify the PLC for access by the TwinCAT System Manager: selection of the target system



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).

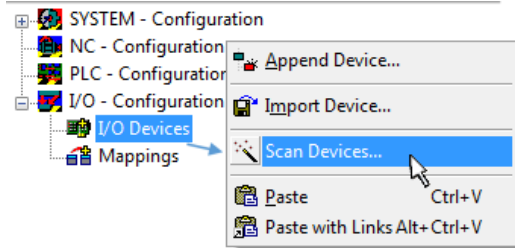


Fig. 38: Select "Scan Devices..."

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

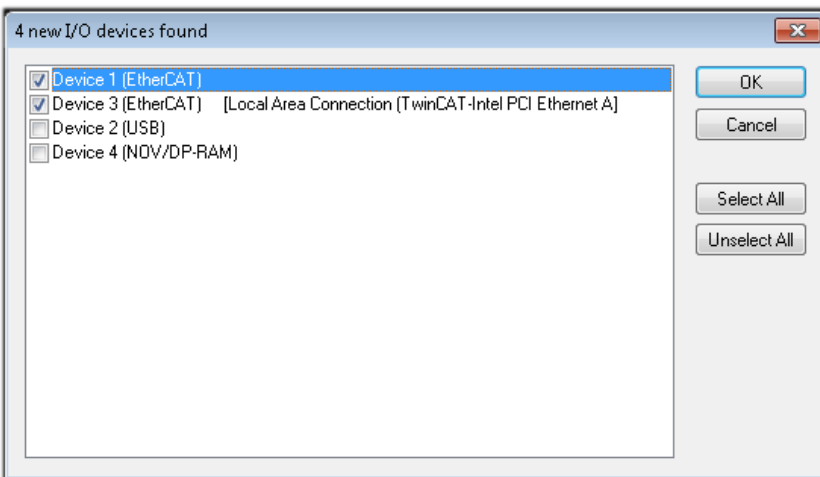


Fig. 39: Automatic detection of I/O devices: selection the devices to be integrated

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 \[▶ 43\]](#) described at the beginning of this section, the result is as follows:

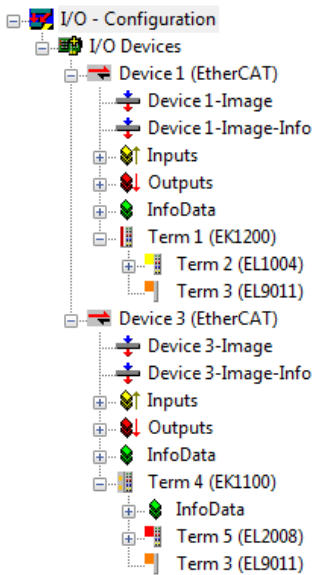


Fig. 40: Mapping of the configuration in the TwinCAT 2 System Manager

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:

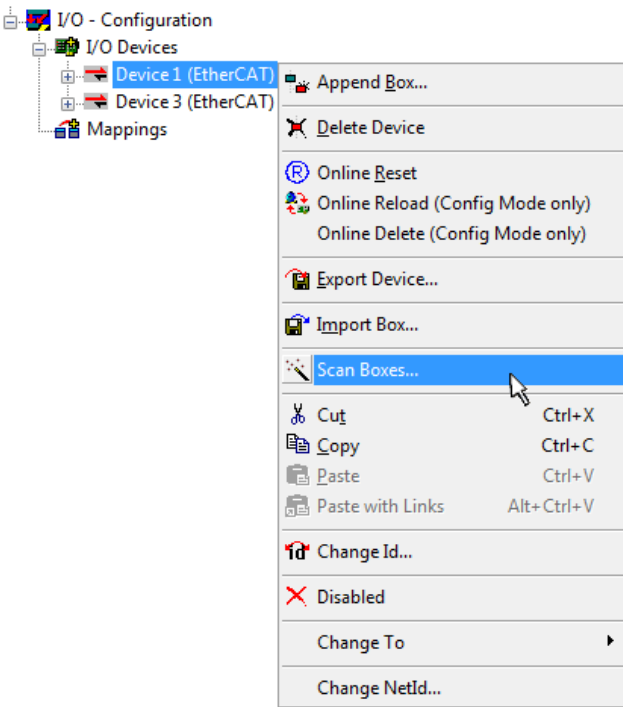


Fig. 41: Reading of individual terminals connected to a device

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:

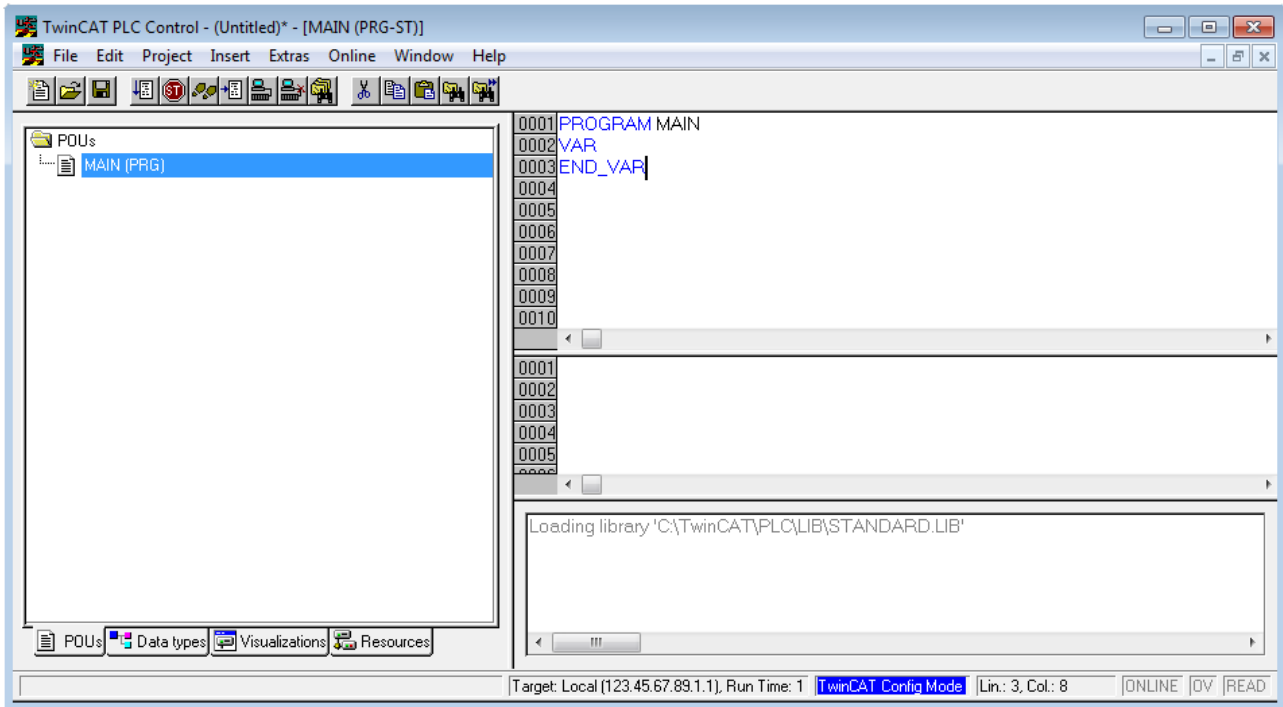


Fig. 42: TwinCAT PLC Control after startup

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

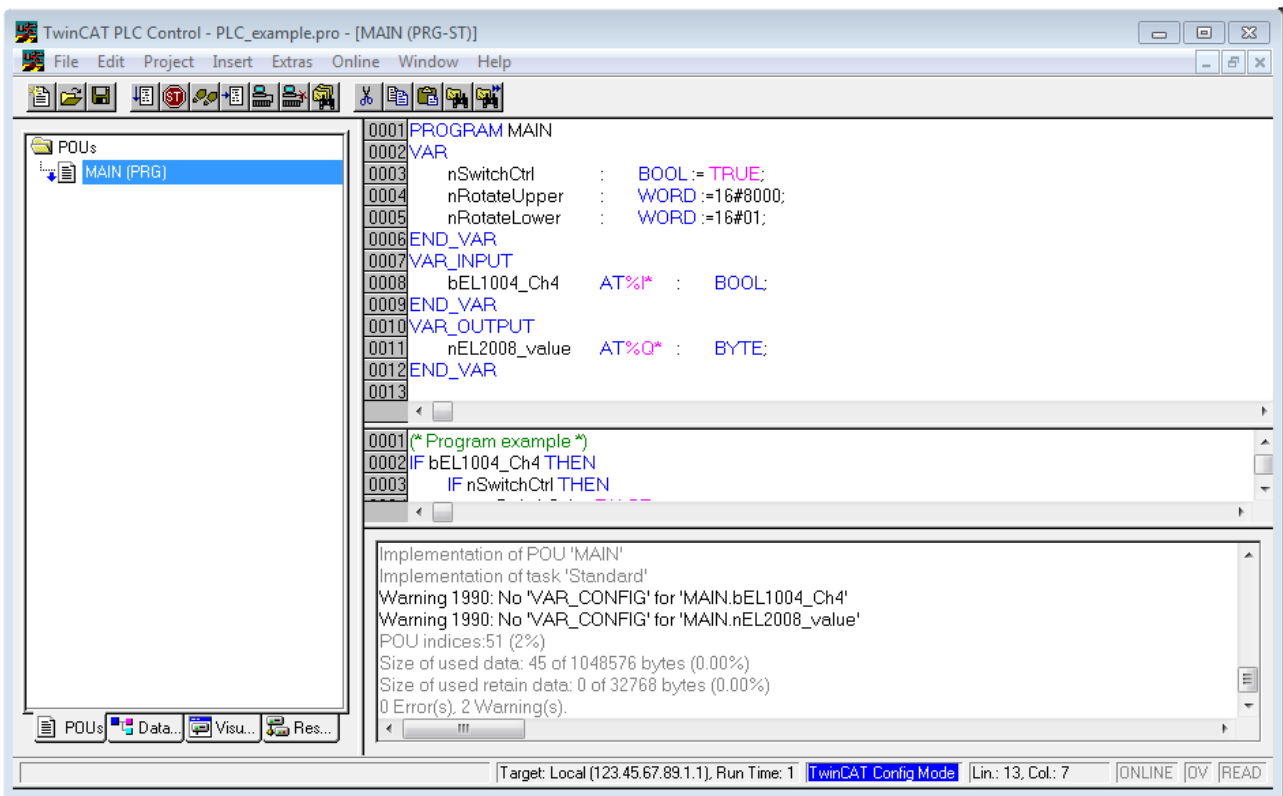


Fig. 43: Sample program with variables after a compile process (without variable integration)

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...":

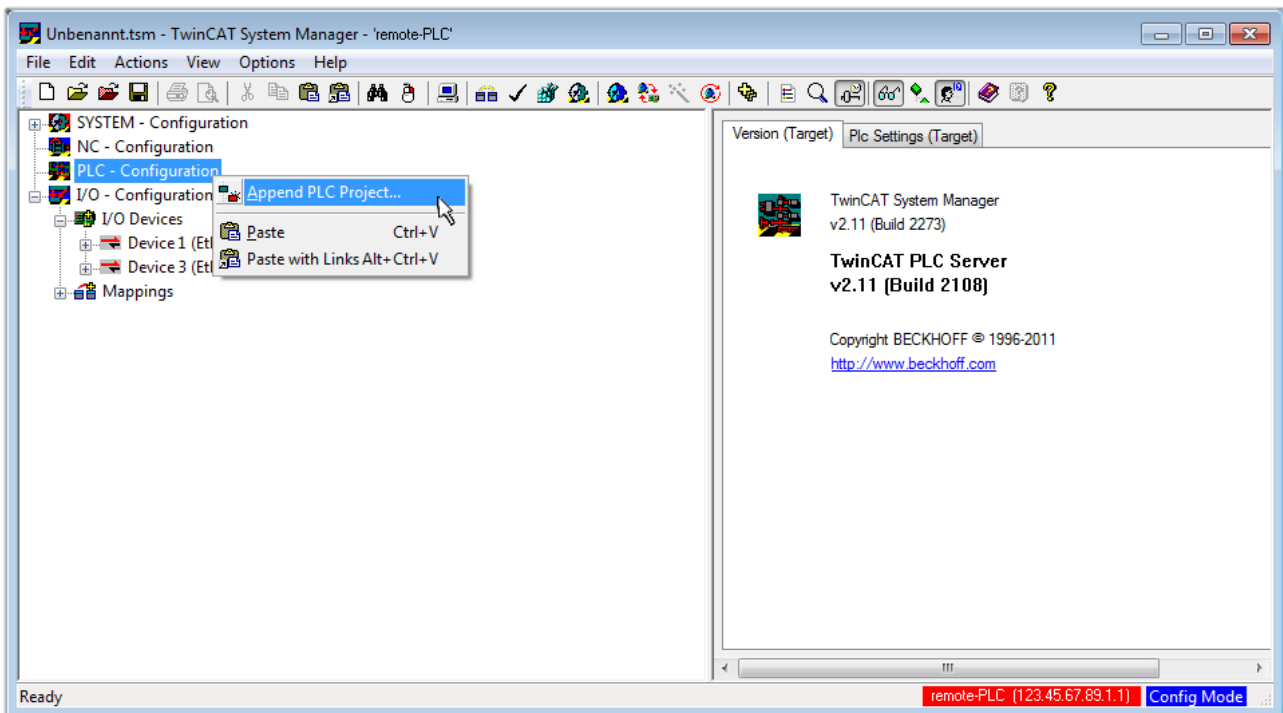


Fig. 44: Appending the TwinCAT PLC Control project

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:

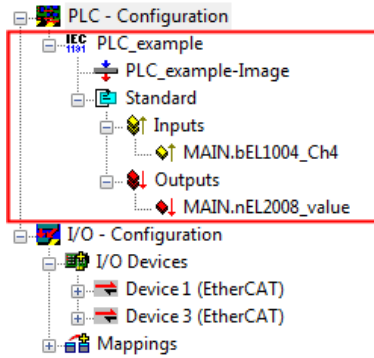


Fig. 45: PLC project integrated in the PLC configuration of the System Manager

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":

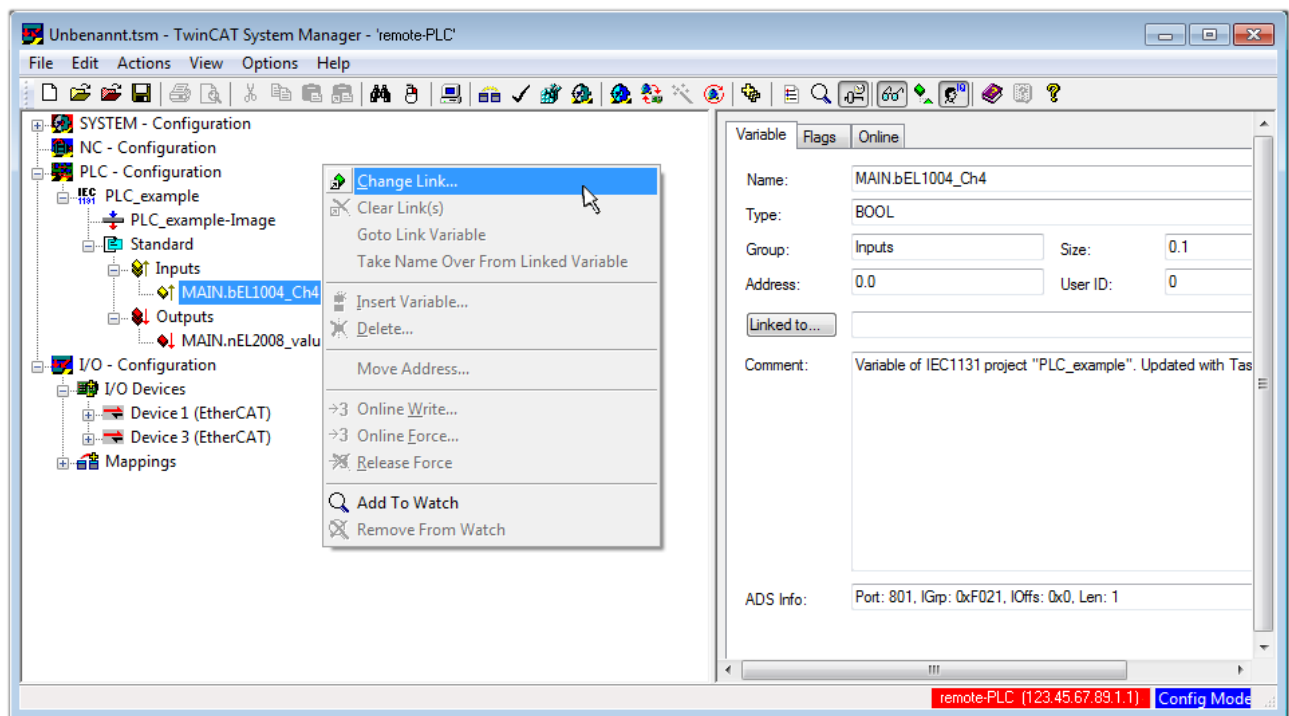


Fig. 46: Creating the links between PLC variables and process objects

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

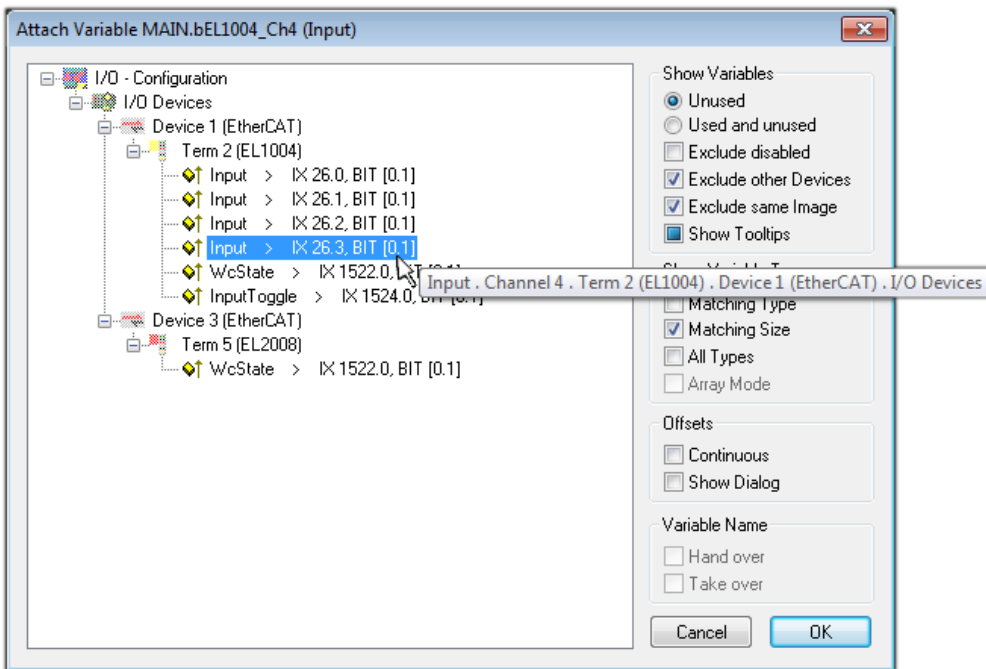


Fig. 47: Selecting PDO of type BOOL

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:

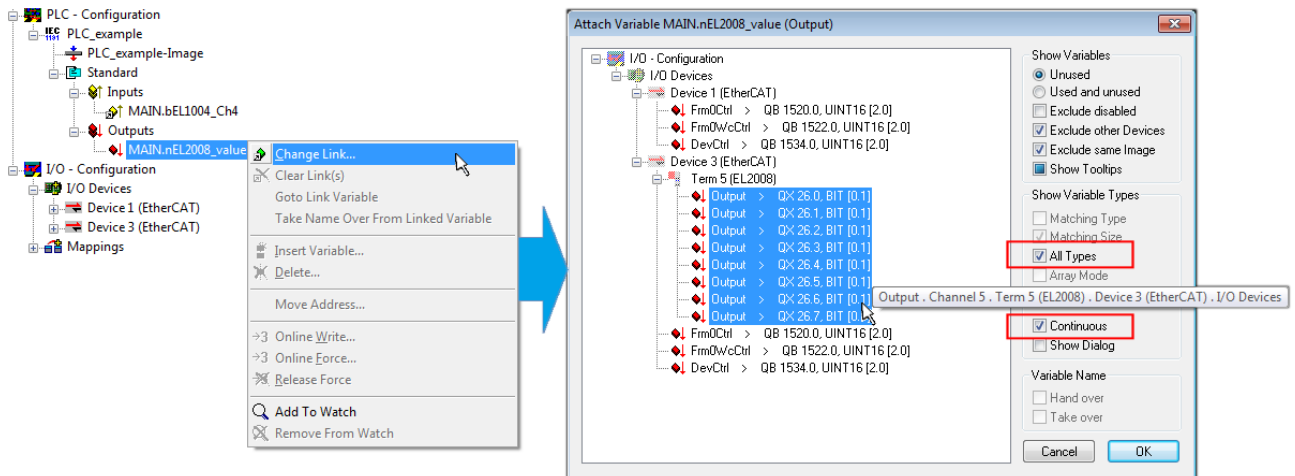



Fig. 48: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

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:

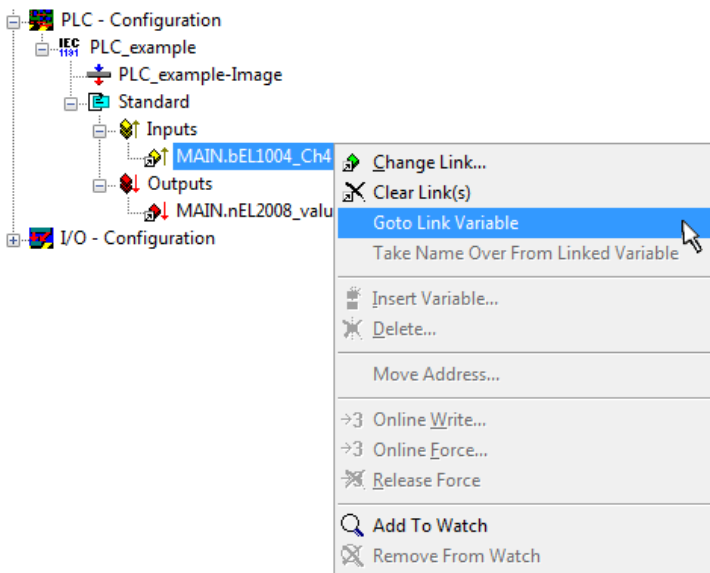
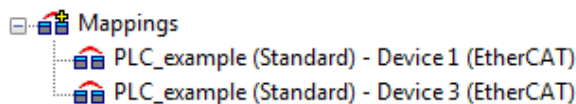


Fig. 49: Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample

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-standardised 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...":

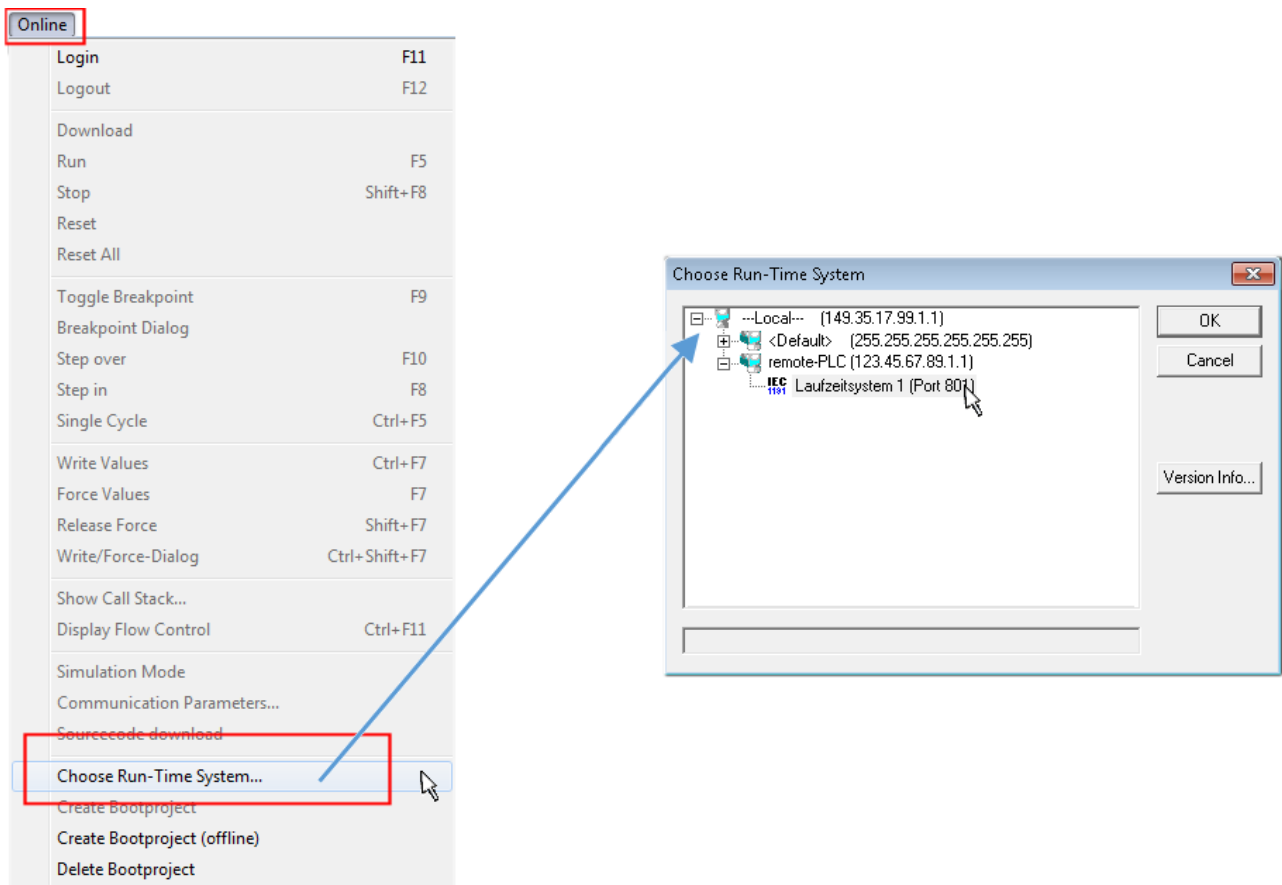



Fig. 50: Choose target system (remote)

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:

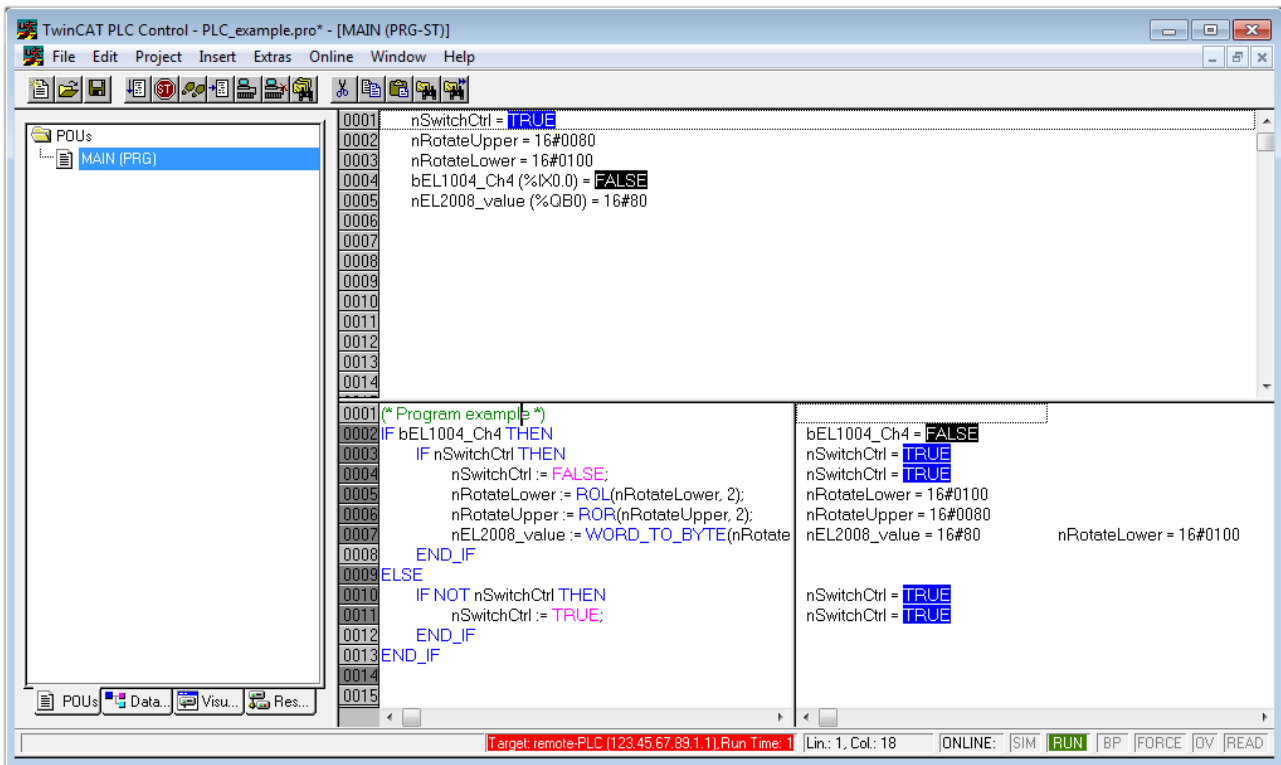


Fig. 51: PLC Control logged in, ready for program startup

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

5.1.1.2 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:

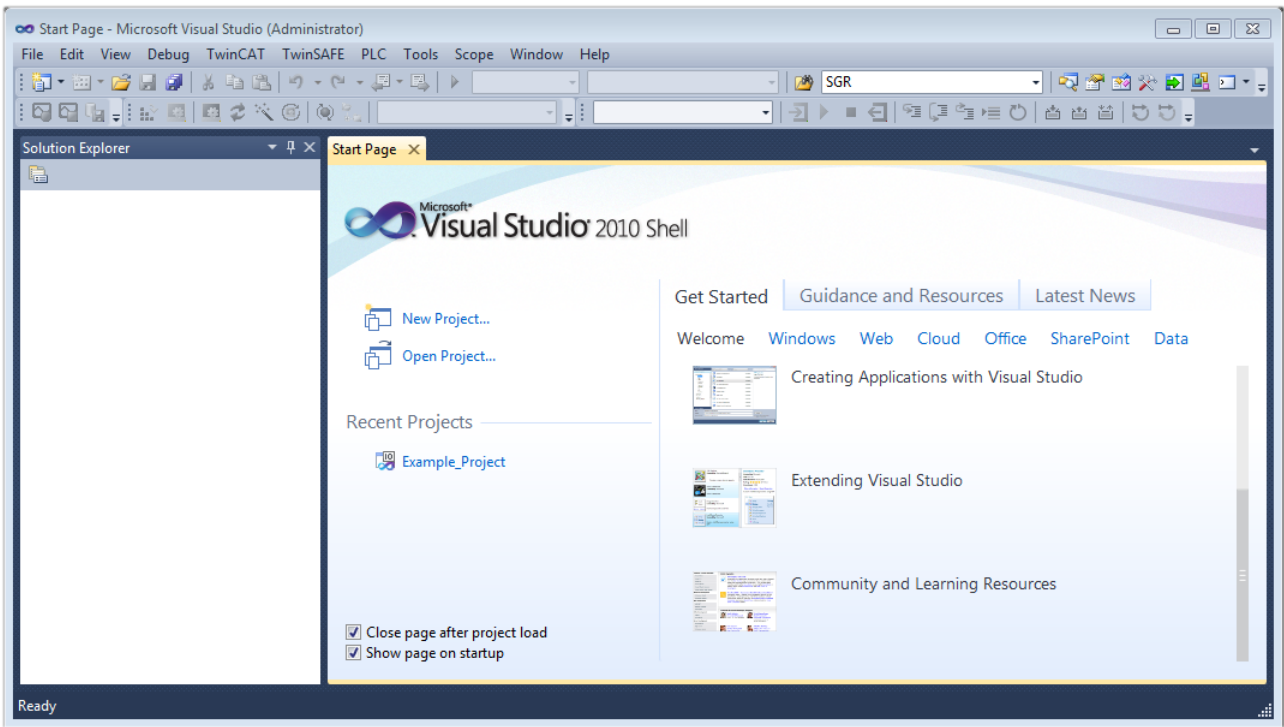



Fig. 52: Initial TwinCAT 3 user interface

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):

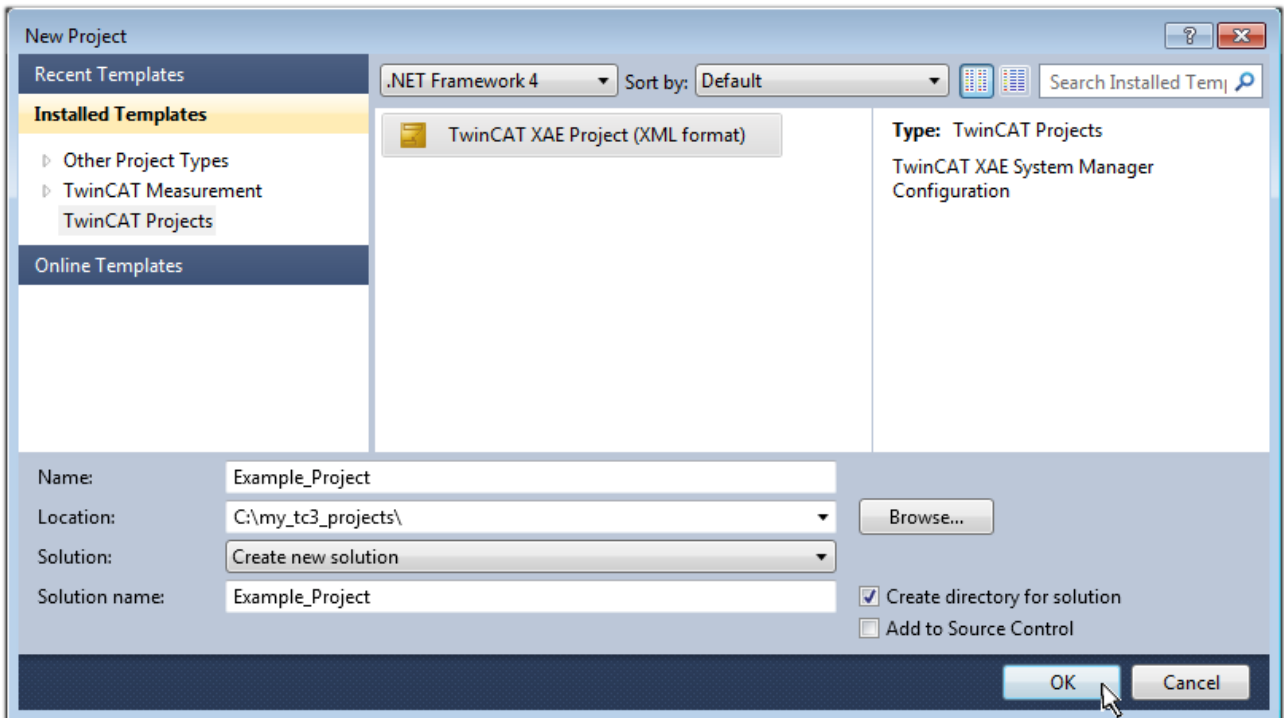


Fig. 53: Create new TwinCAT project

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

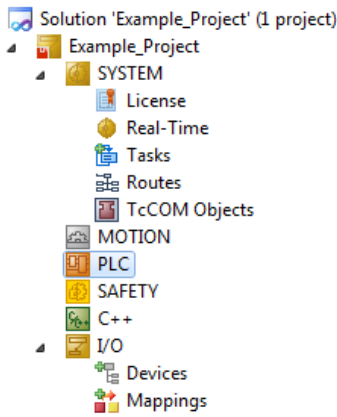
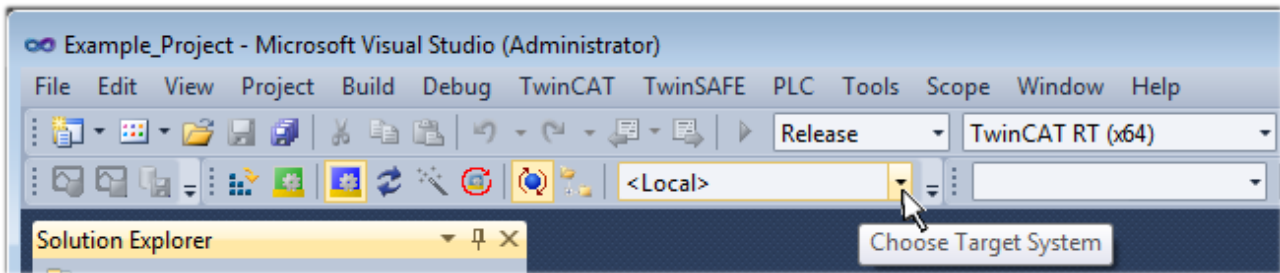


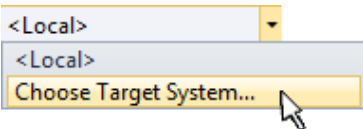
Fig. 54: New TwinCAT3 project in the project folder explorer

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 [▶ 58]".

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:

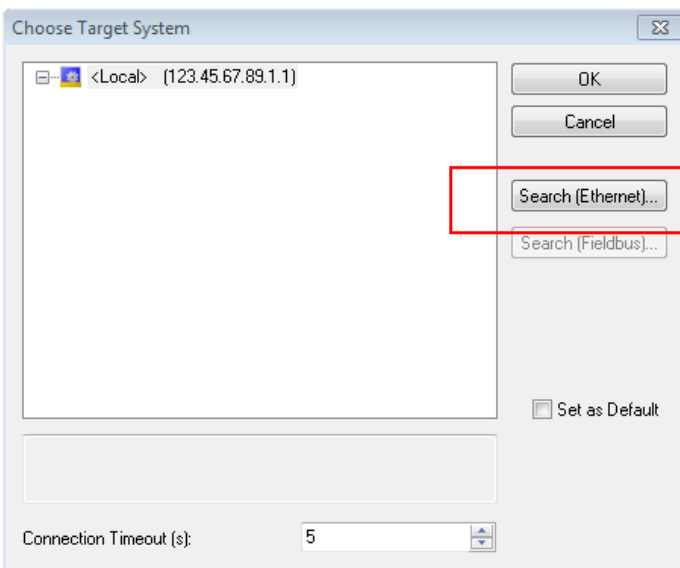


Fig. 55: Selection dialog: Choose the target system

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.

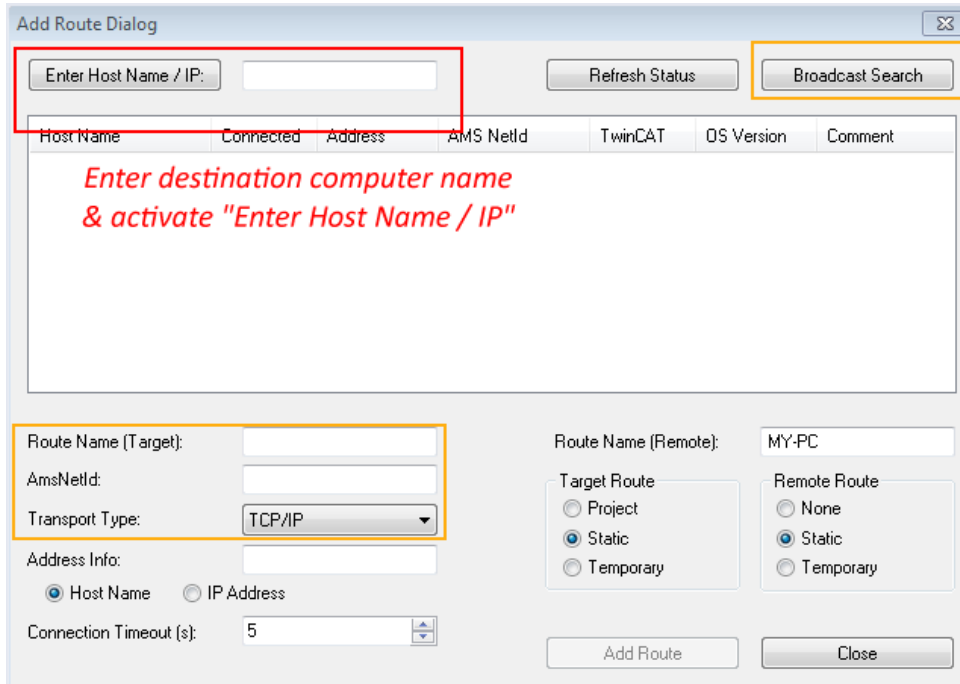
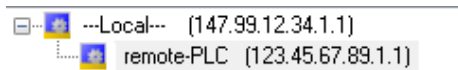


Fig. 56: Specify the PLC for access by the TwinCAT System Manager: selection of the target system


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)".

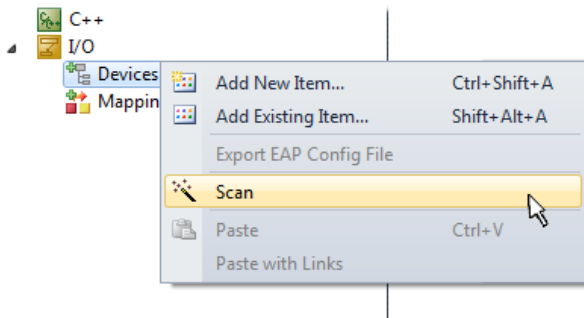


Fig. 57: Select "Scan"

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

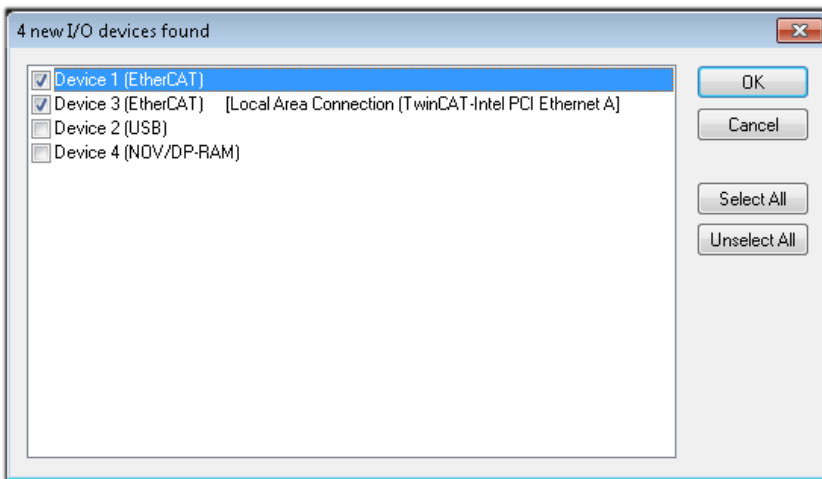


Fig. 58: Automatic detection of I/O devices: selection the devices to be integrated

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 \[▶ 43\]](#) described at the beginning of this section, the result is as follows:

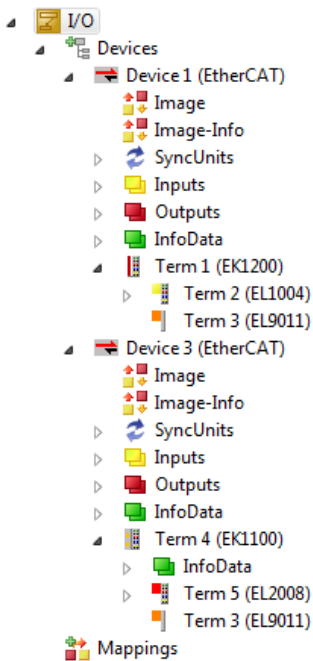


Fig. 59: Mapping of the configuration in VS shell of the TwinCAT3 environment

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:

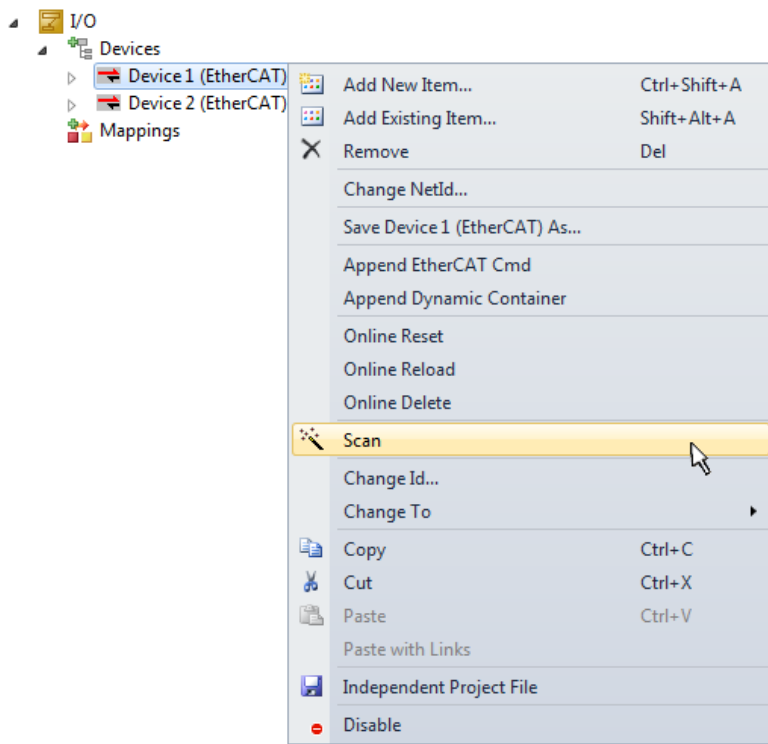


Fig. 60: Reading of individual terminals connected to a device

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....":

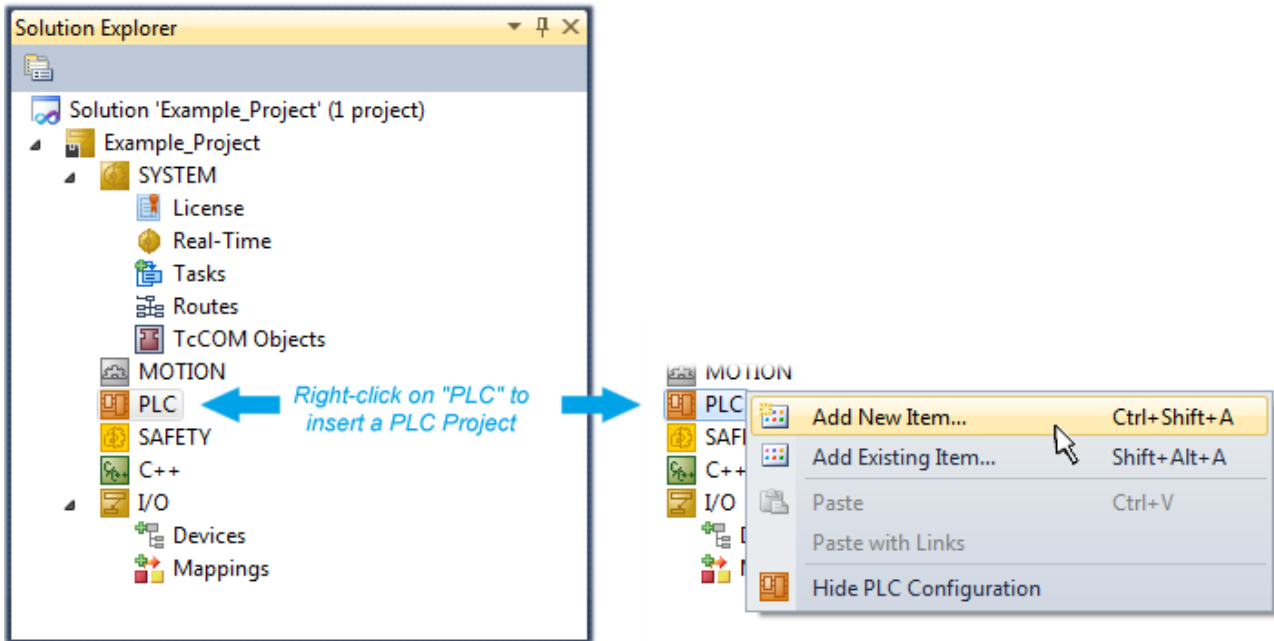


Fig. 61: Adding the programming environment in "PLC"

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

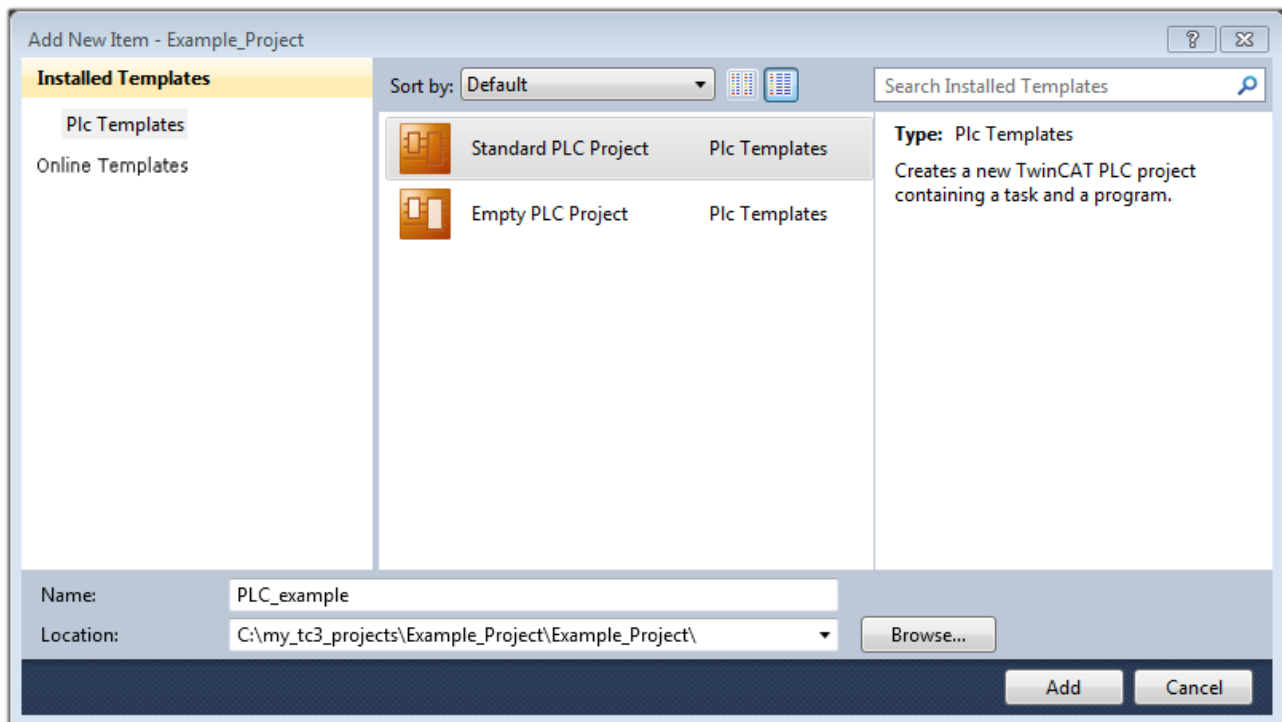


Fig. 62: Specifying the name and directory for the PLC programming environment

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:

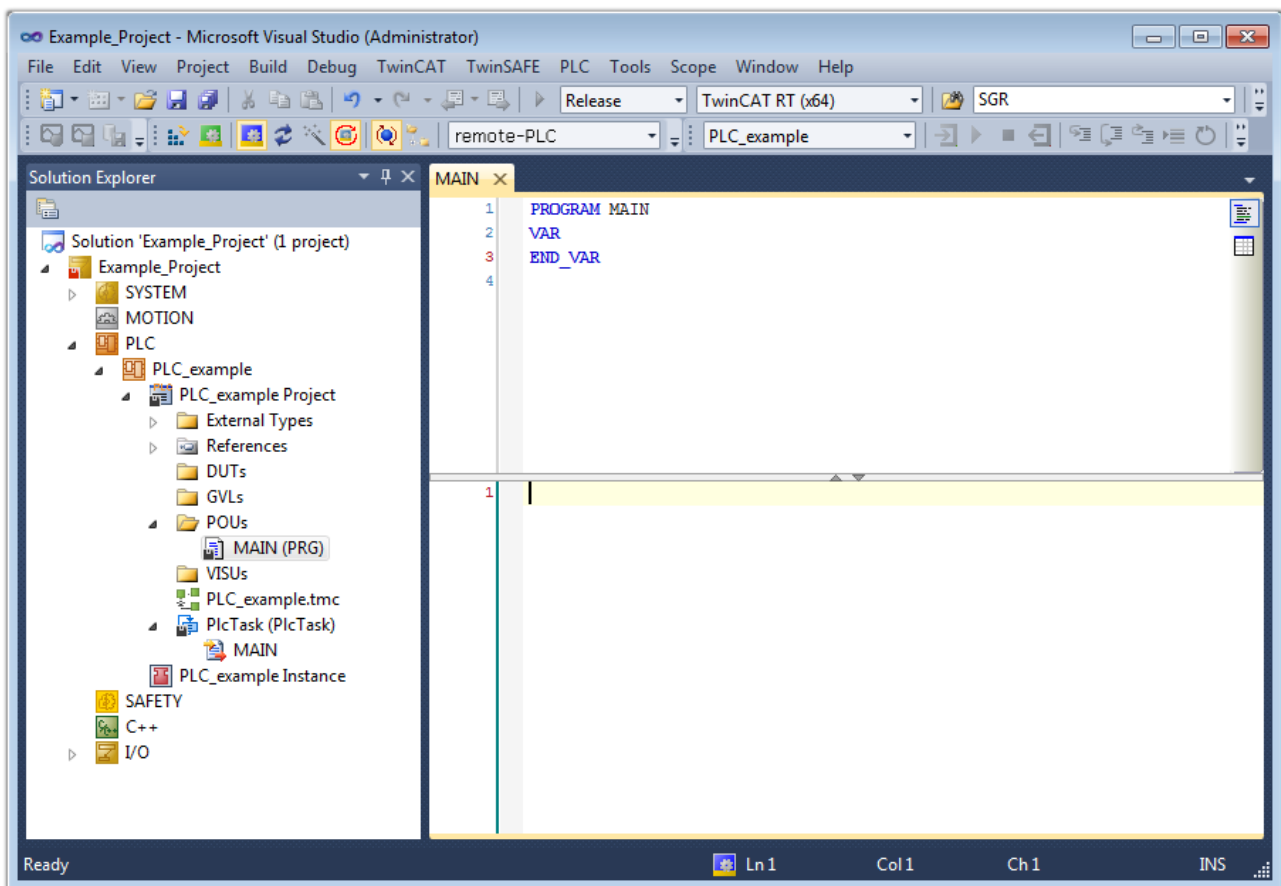


Fig. 63: Initial "Main" program of the standard PLC project

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

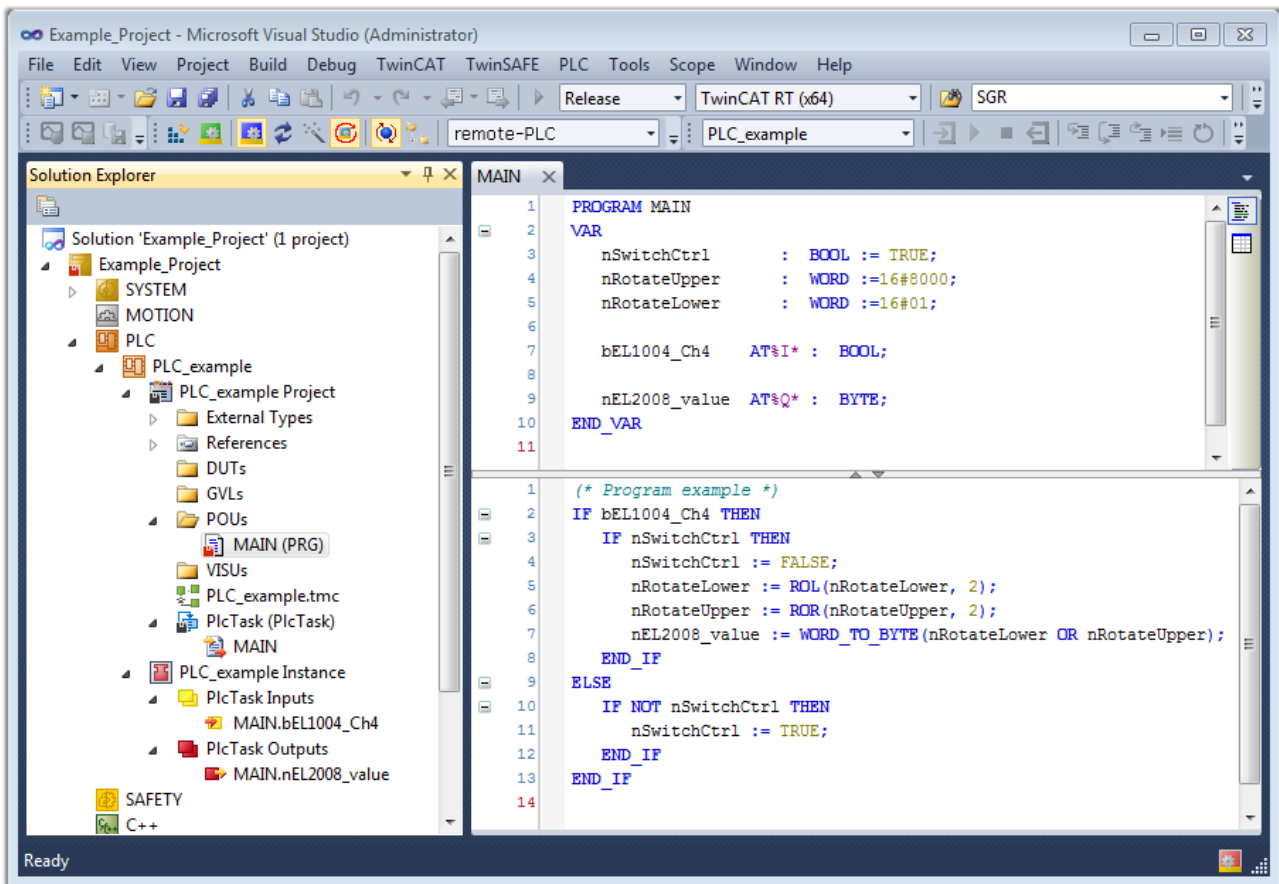


Fig. 64: Sample program with variables after a compile process (without variable integration)

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

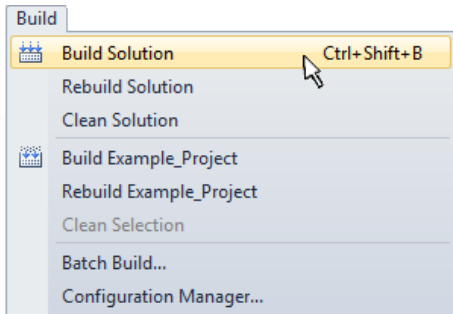
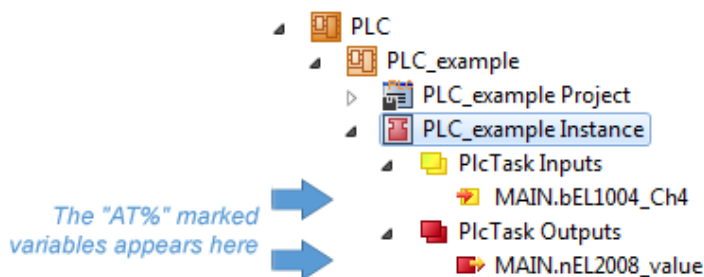


Fig. 65: Start program compilation

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:

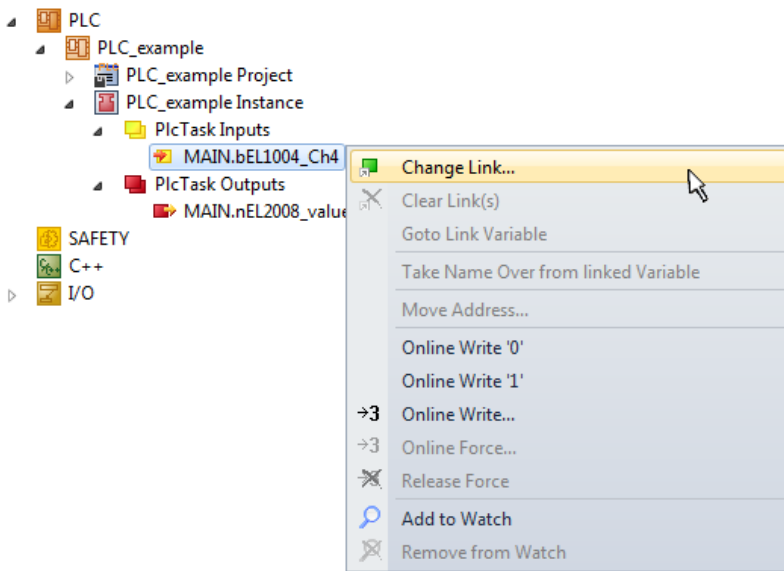


Fig. 66: Creating the links between PLC variables and process objects

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

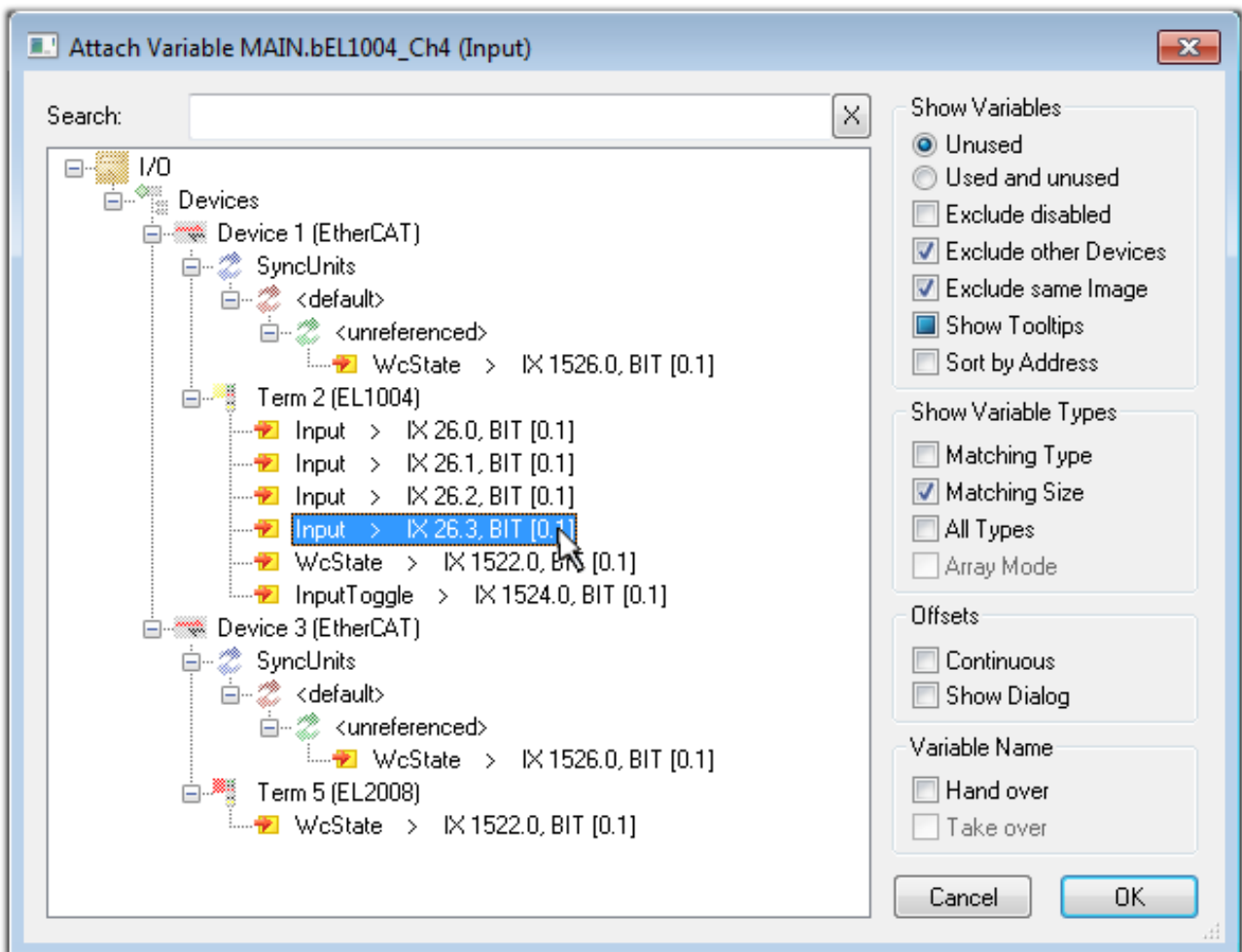


Fig. 67: Selecting PDO of type BOOL

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:

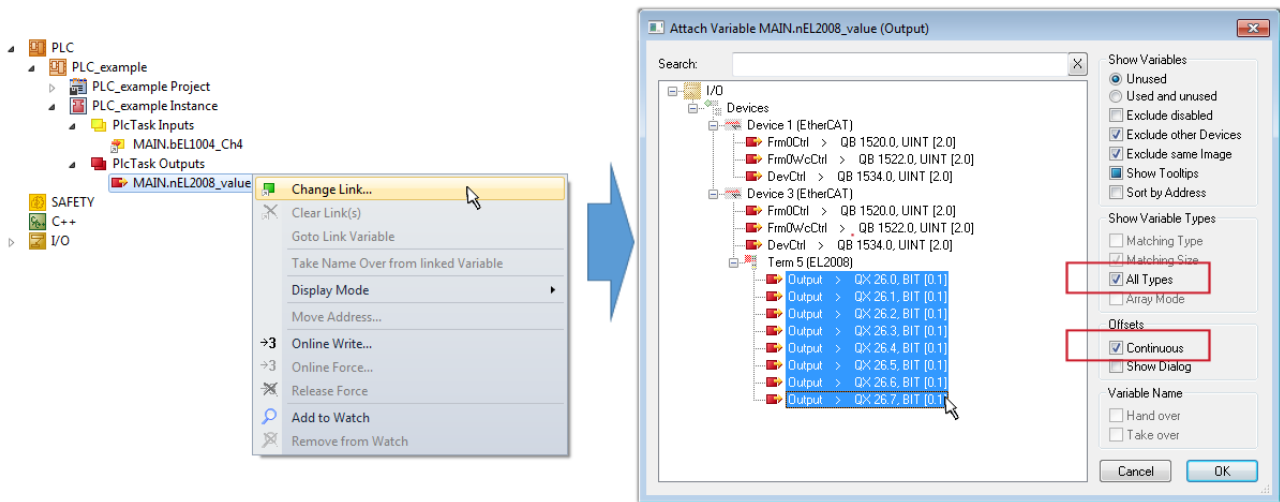



Fig. 68: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

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:

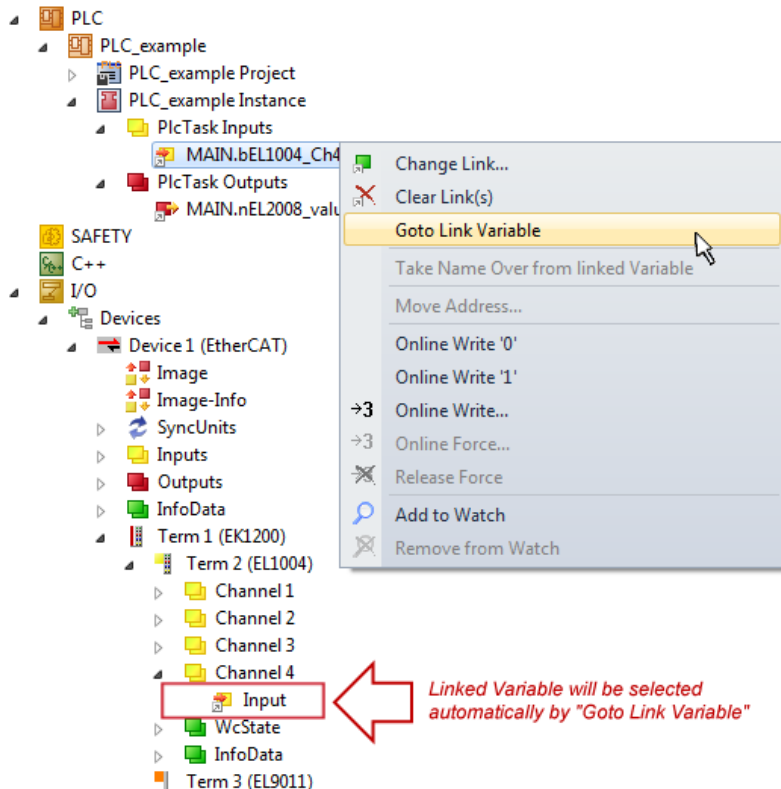


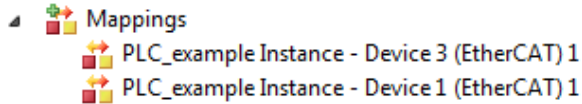
Fig. 69: Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample

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-standardised 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 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:

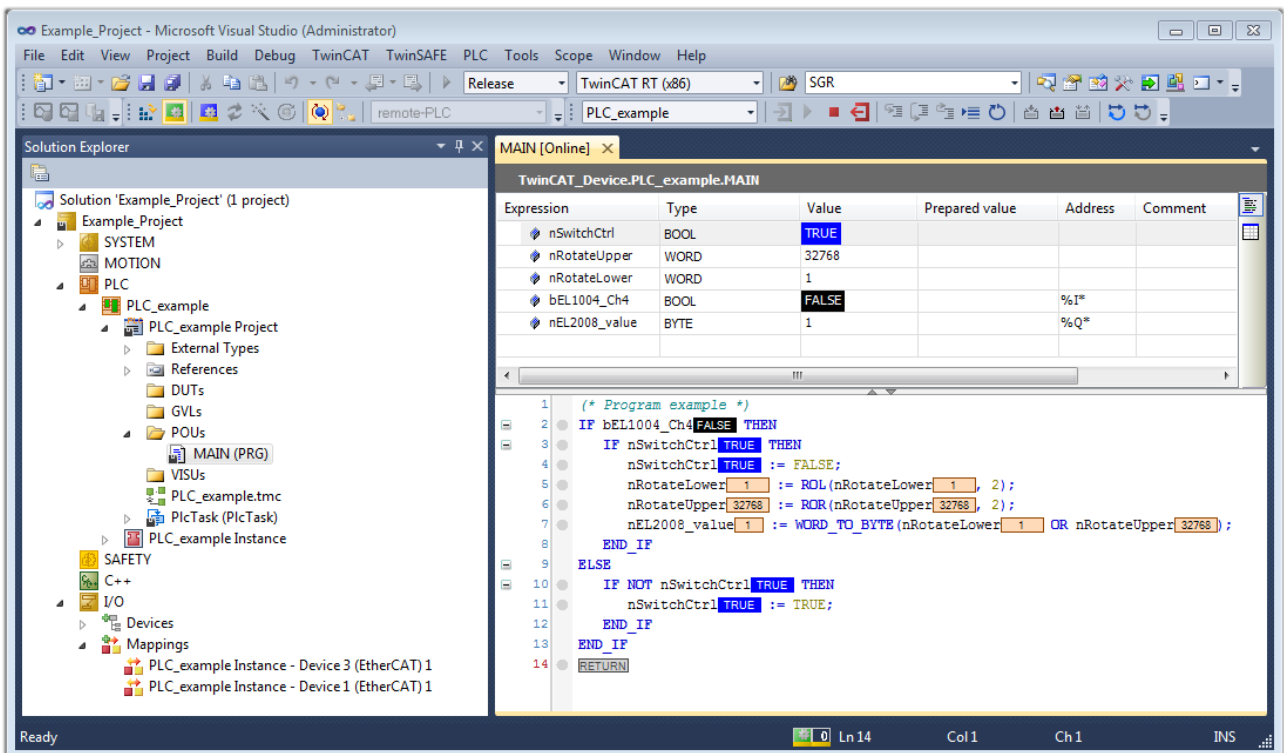


Fig. 70: TwinCAT development environment (VS shell): logged-in, after program startup

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).

5.1.2 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>.

5.1.2.1 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.

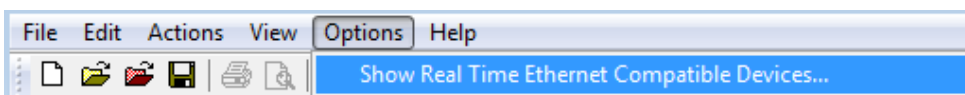


Fig. 71: System Manager "Options" (TwinCAT 2)

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

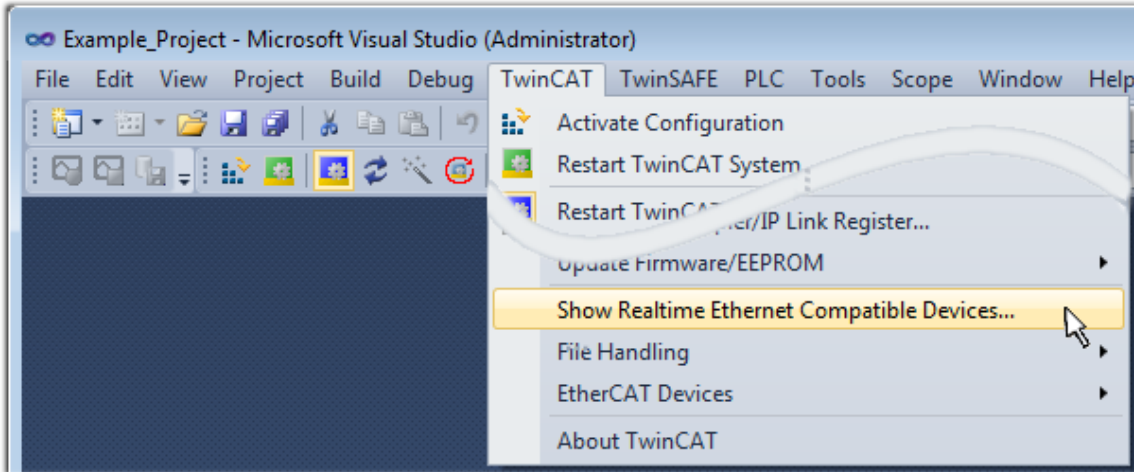


Fig. 72: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

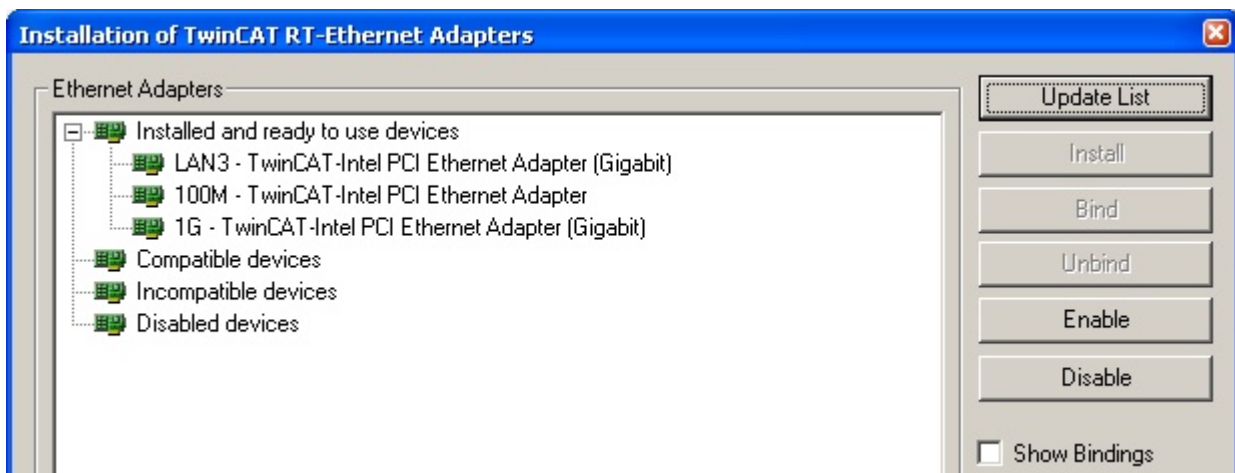


Fig. 73: Overview of network interfaces

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

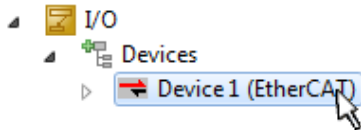
A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter [Offline configuration creation, section “Creating the EtherCAT device” \[► 77\]](#) in order to view the compatible ethernet ports via its EtherCAT properties (tab „Adapter“, button „Compatible Devices...“):



Fig. 74: EtherCAT device properties(TwinCAT 2): click on „Compatible Devices...“ of tab “Adapter”

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



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

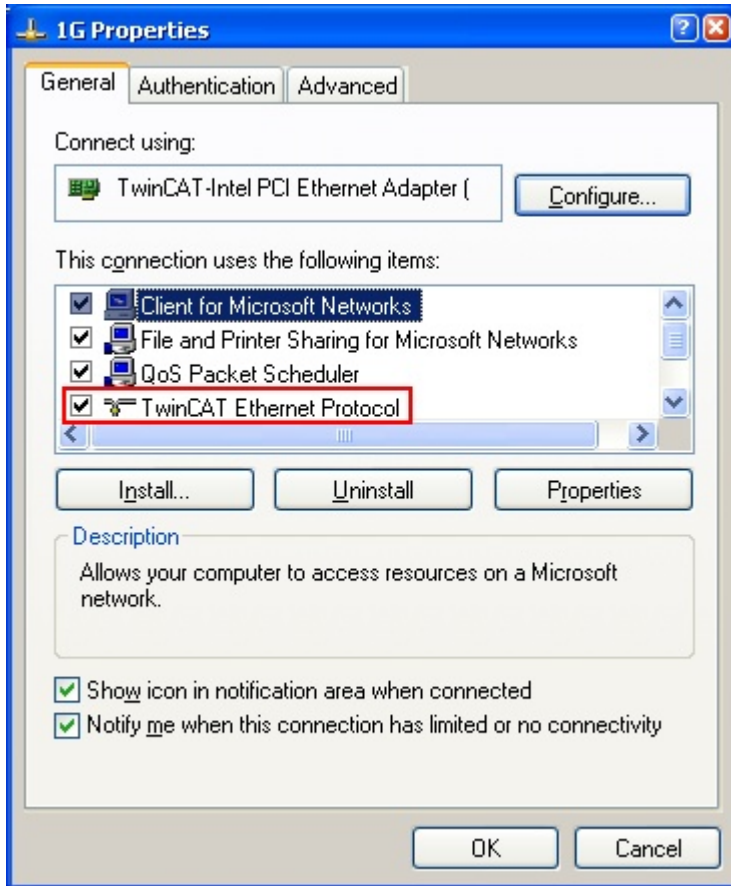


Fig. 75: Windows properties of the network interface

A correct setting of the driver could be:

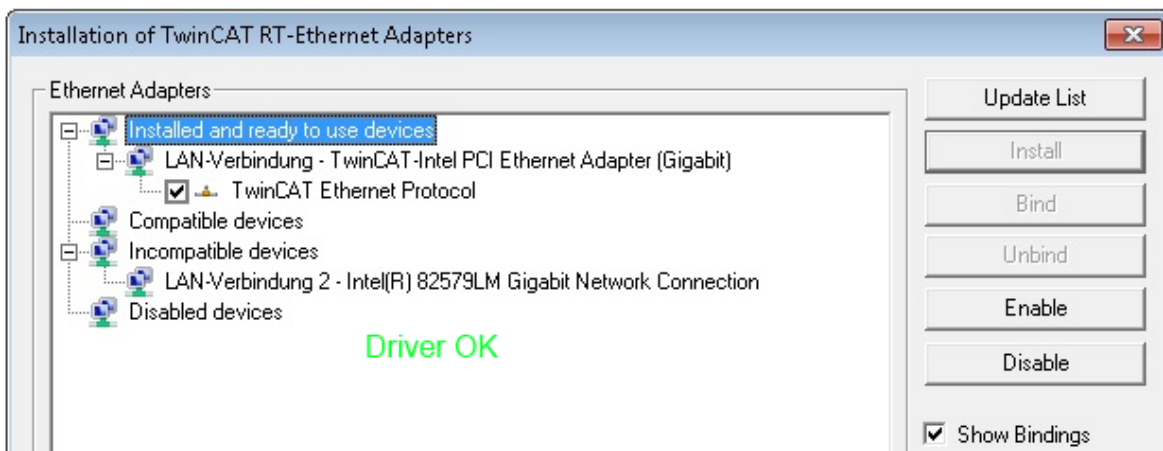


Fig. 76: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

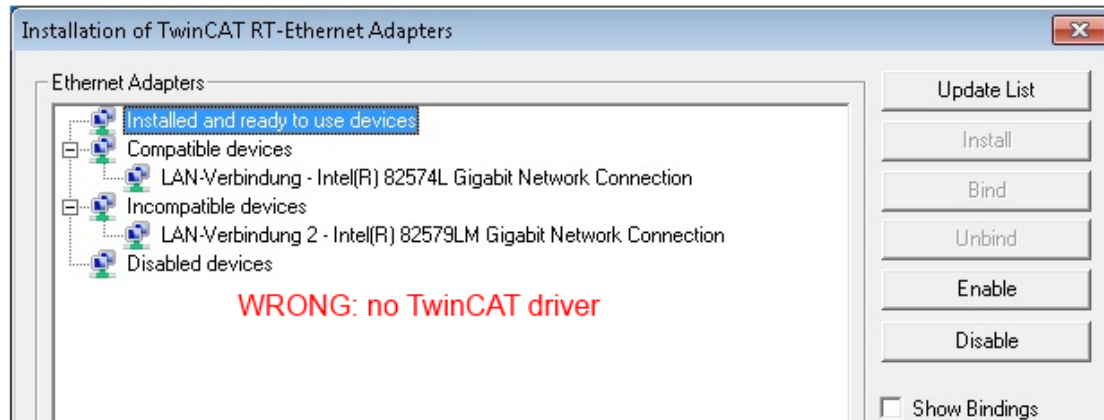
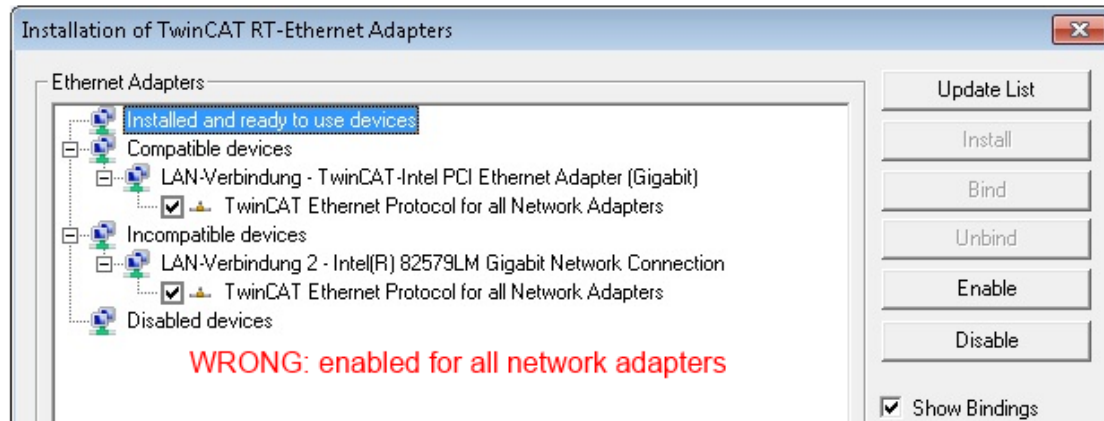
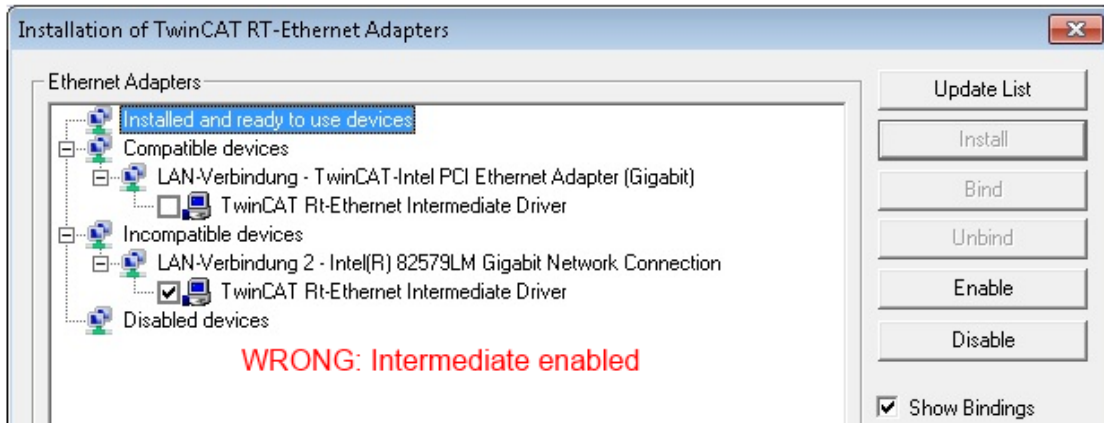
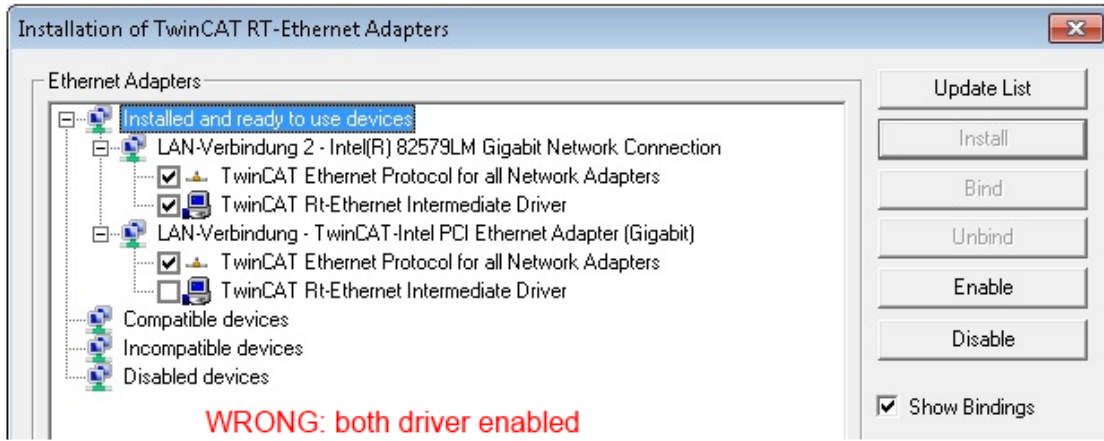


Fig. 77: Incorrect driver settings for the Ethernet port

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.

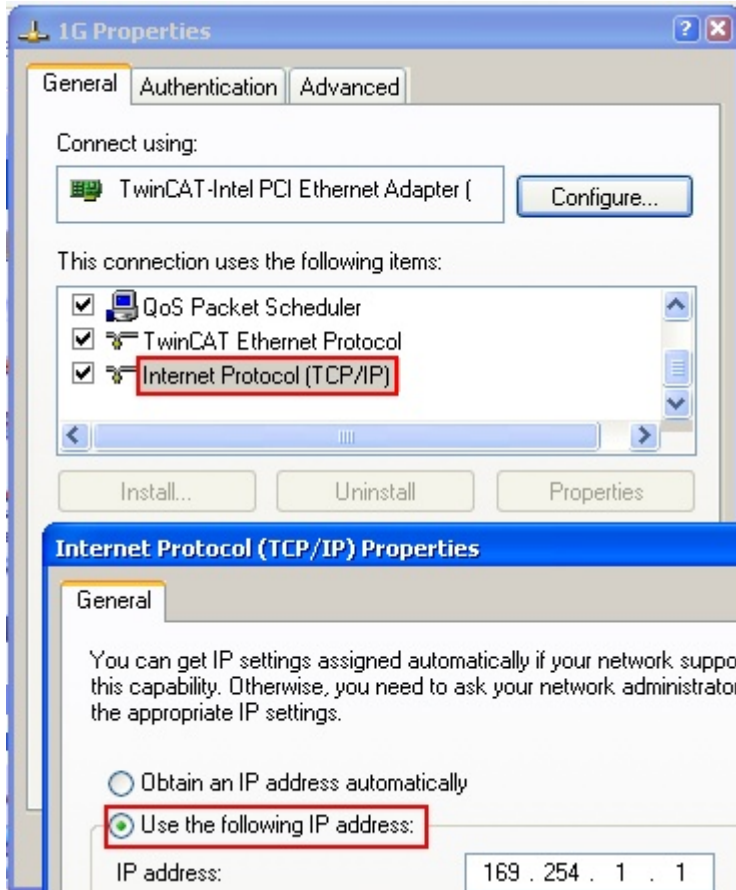


Fig. 78: TCP/IP setting for the Ethernet port

5.1.2.2 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](#) [► 76] 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”

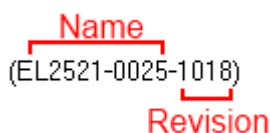


Fig. 79: Identifier structure

The order identifier consisting of name + type (here: EL2521-0010) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See [further notes](#) [► 7].

Online description

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

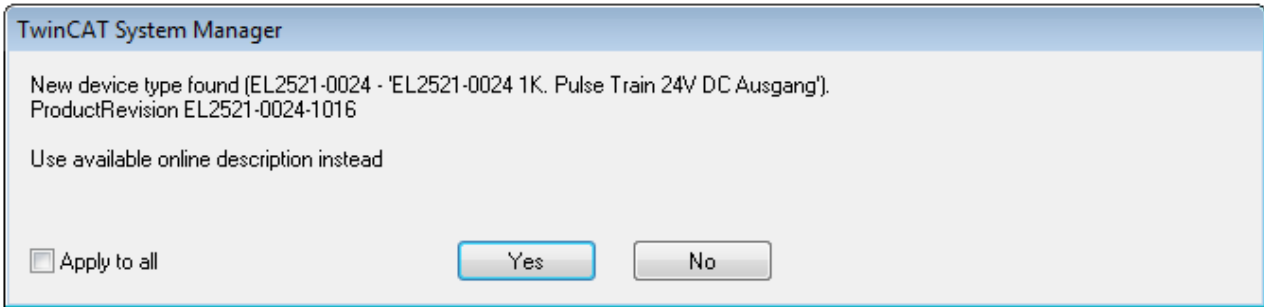


Fig. 80: *OnlineDescription information window (TwinCAT 2)*

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

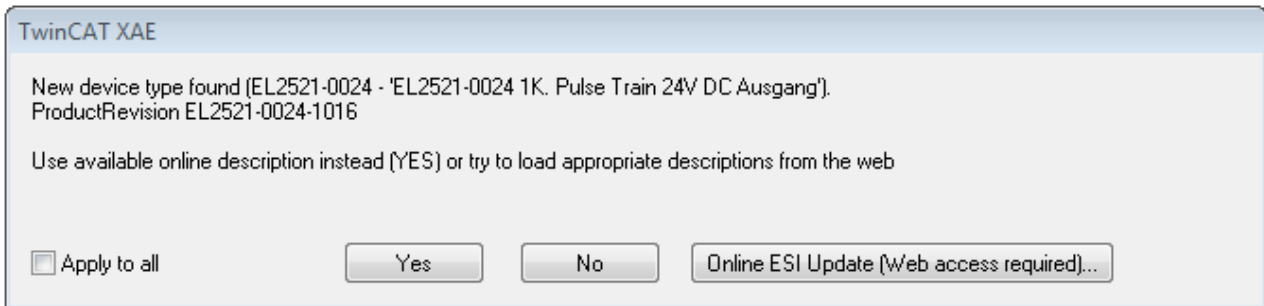


Fig. 81: *Information window OnlineDescription (TwinCAT 3)*

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](#)’ [► 77].

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. 82: 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").

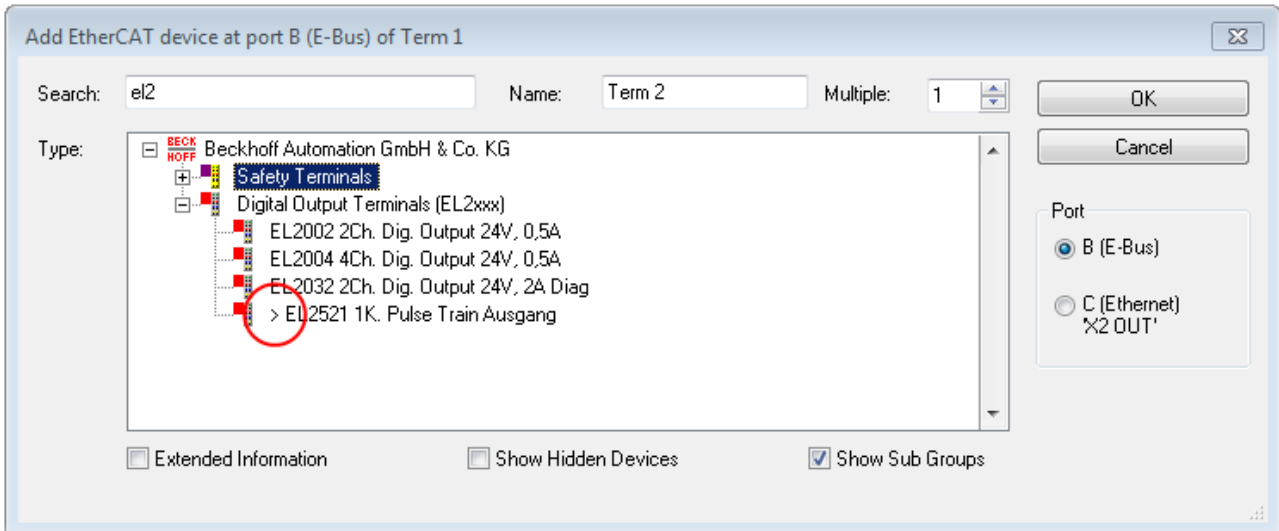


Fig. 83: Indication of an online recorded ESI of EL2521 as an example

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.

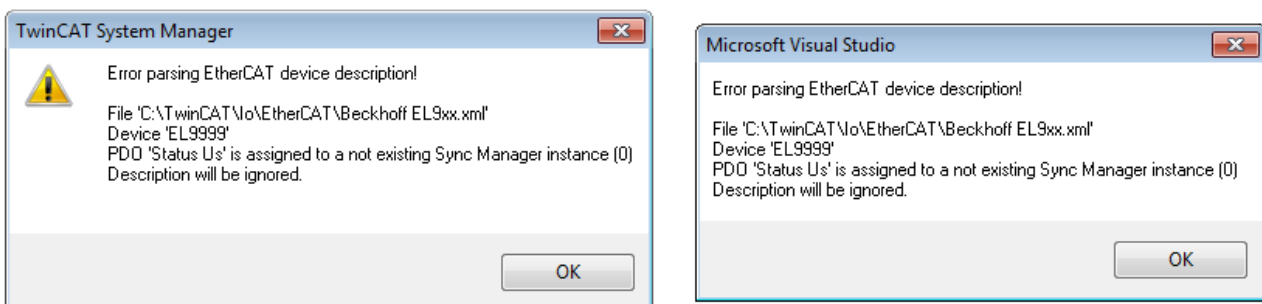


Fig. 84: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

5.1.2.3 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:

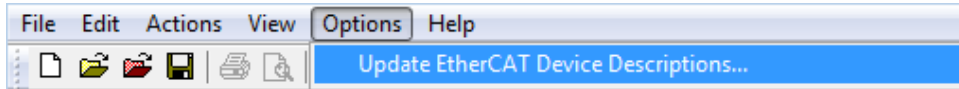


Fig. 85: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

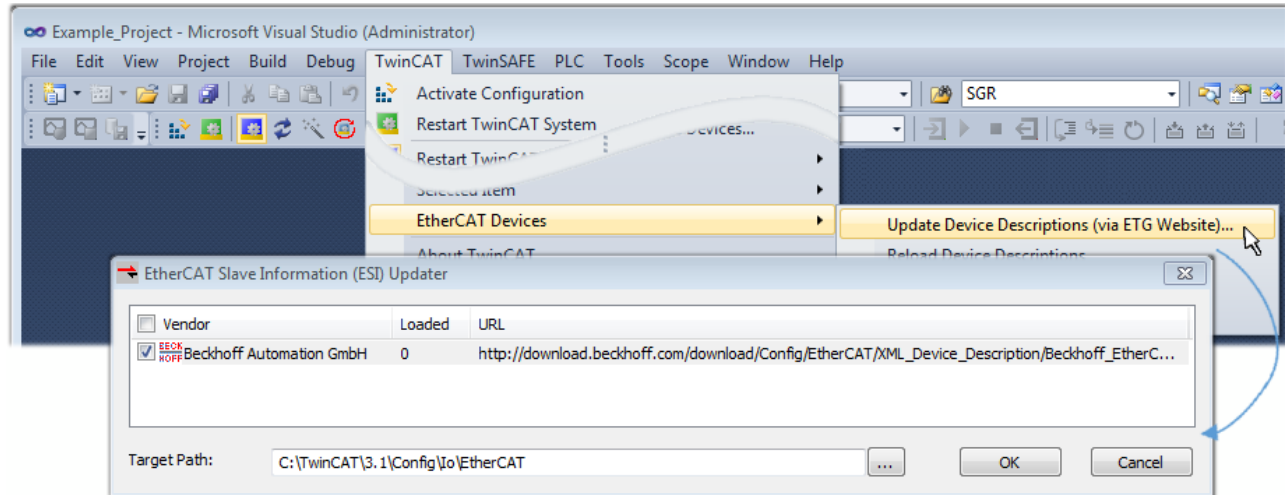


Fig. 86: Using the ESI Updater (TwinCAT 3)

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)...".

5.1.2.4 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" \[▶ 72\]](#).

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 [▶ 82] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 83]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 86]

The scan with existing configuration [▶ 87] can also be carried out for comparison.

5.1.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

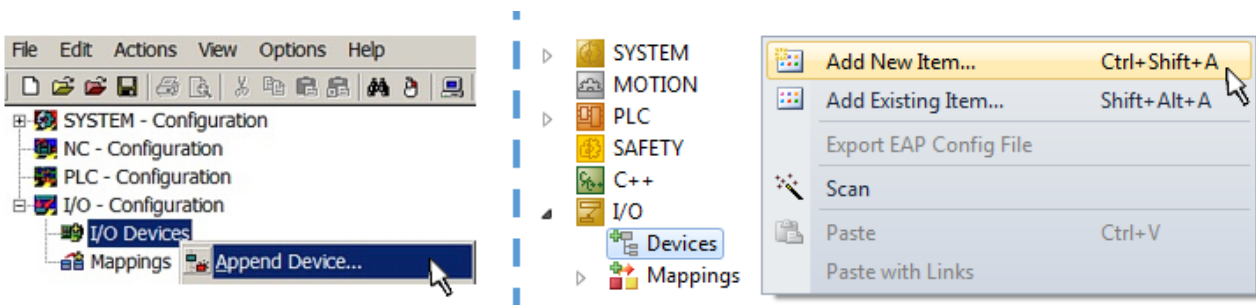


Fig. 87: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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".

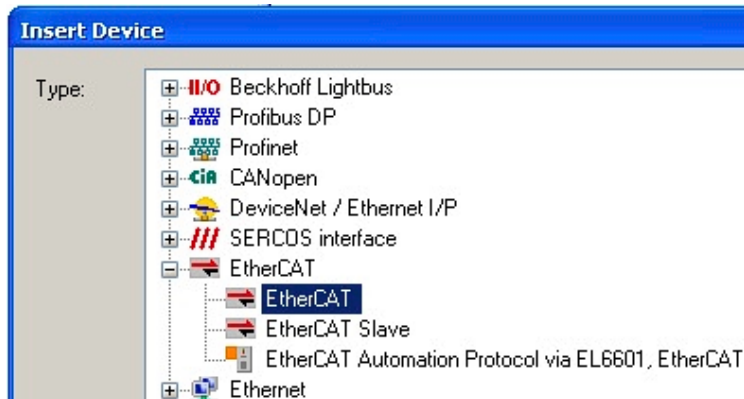


Fig. 88: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

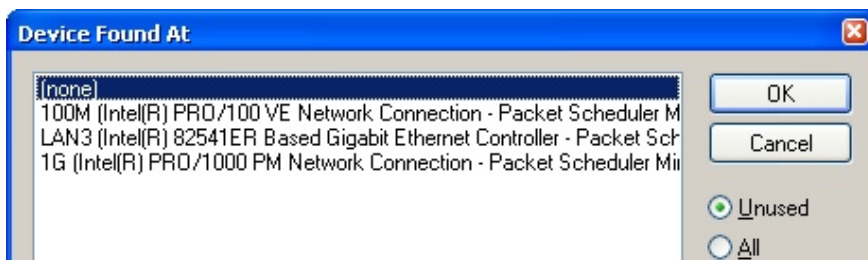


Fig. 89: Selecting the Ethernet port

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)”.

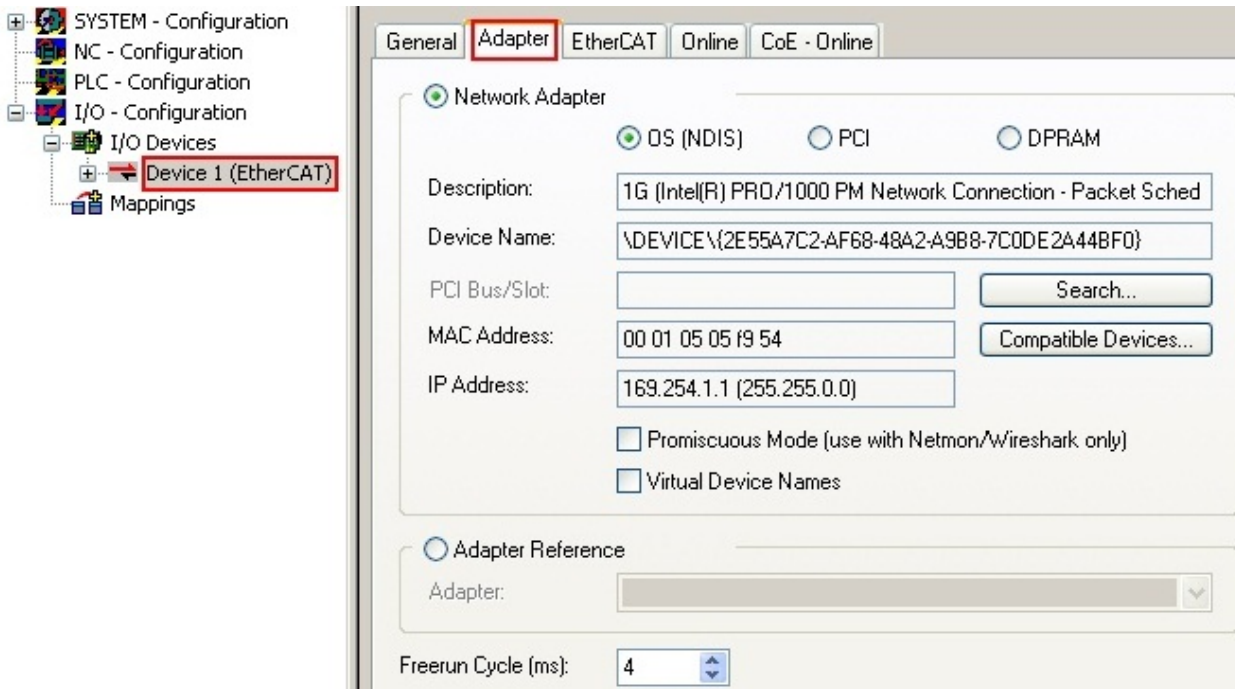
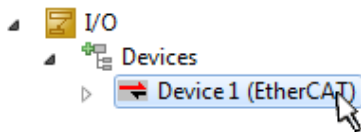


Fig. 90: EtherCAT device properties (TwinCAT 2)

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 \[p. 67\]](#).

Defining EtherCAT slaves

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

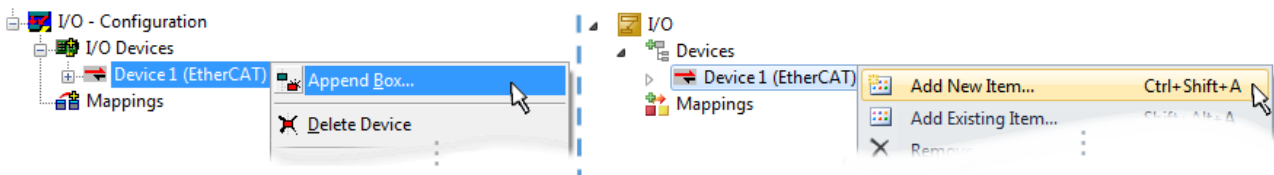


Fig. 91: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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).

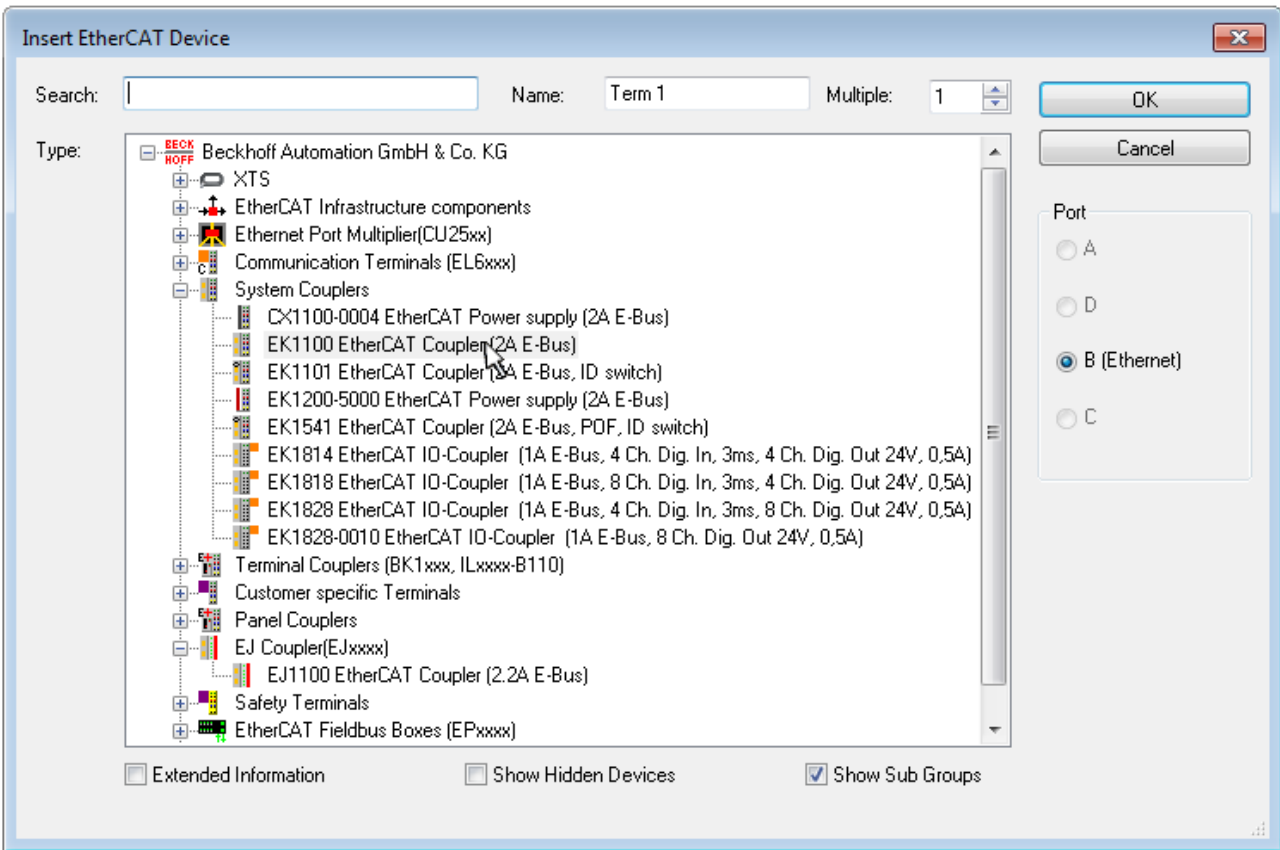


Fig. 92: Selection dialog for new EtherCAT device

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”.

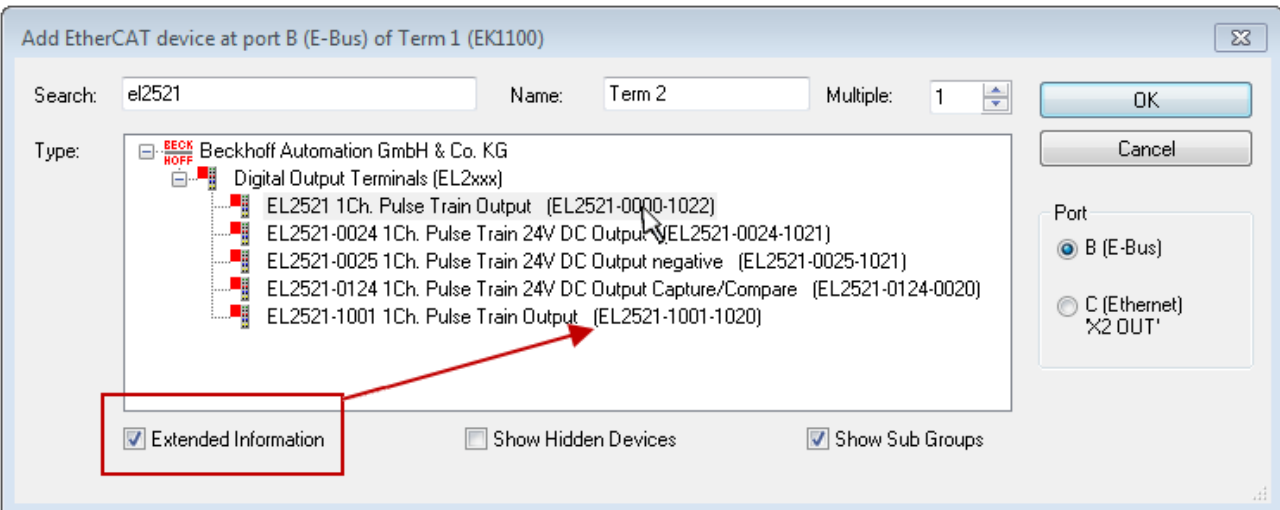


Fig. 93: Display of device revision

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”.

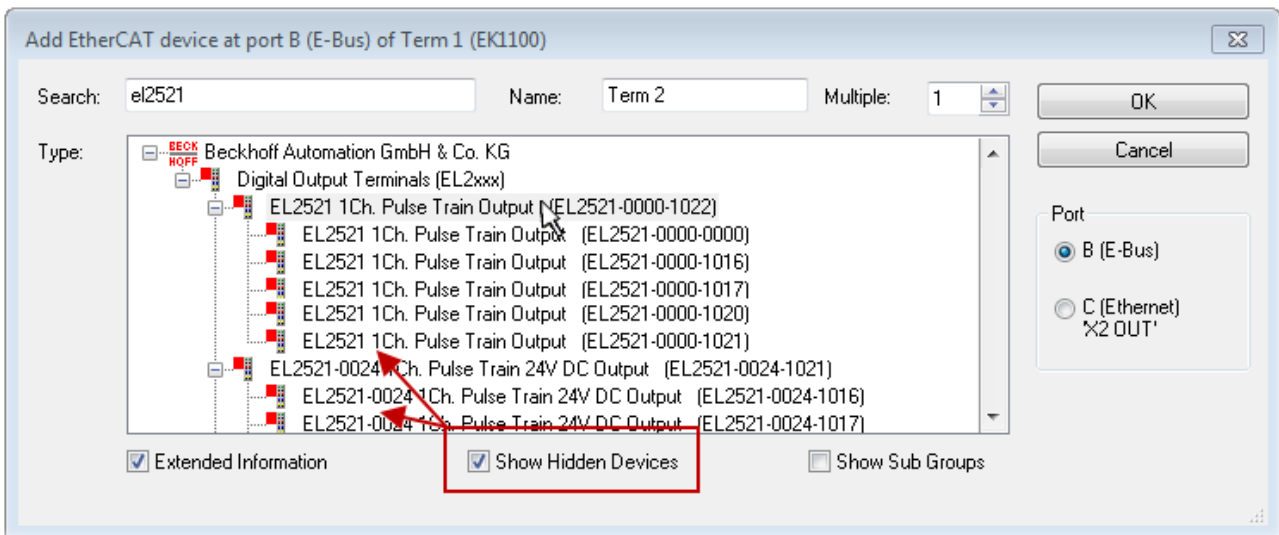


Fig. 94: *Display of previous revisions*

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.

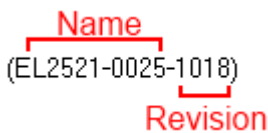


Fig. 95: *Name/revision of the terminal*

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 parameterised as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...

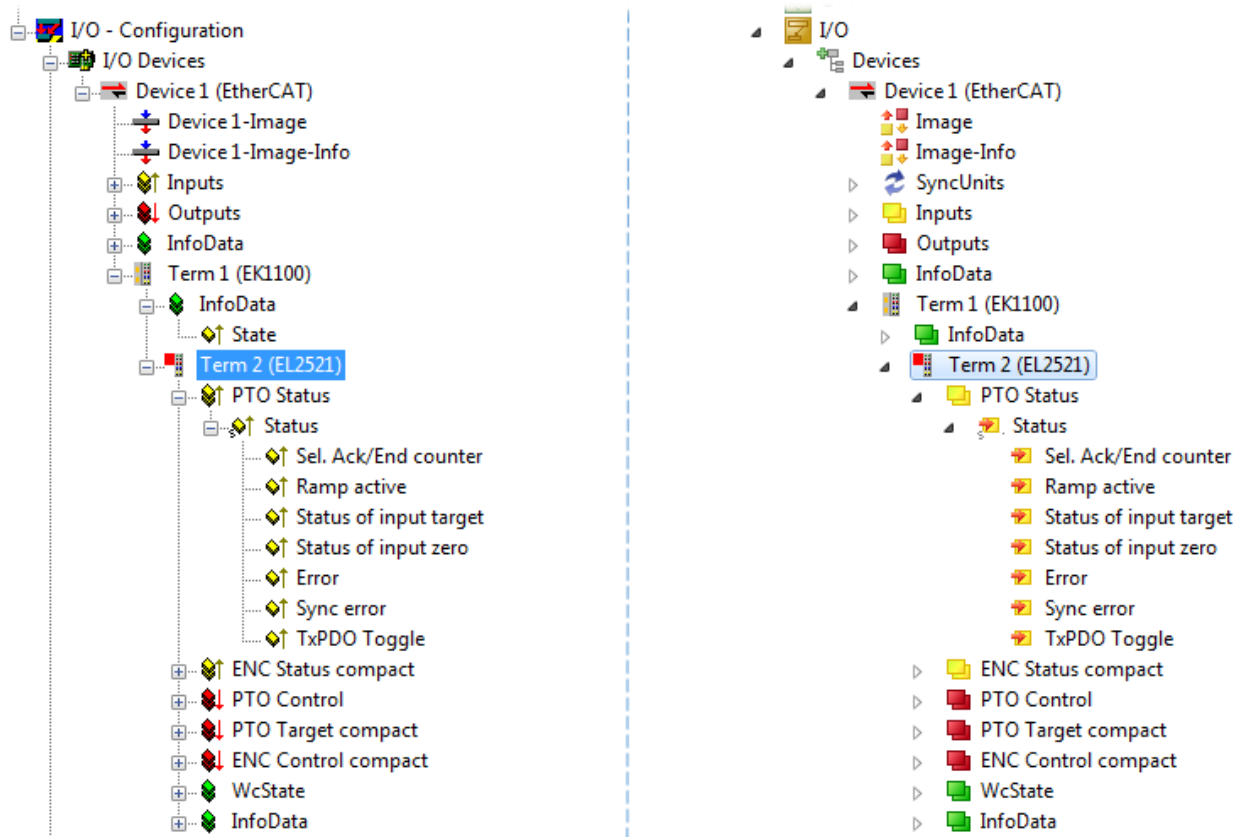




Fig. 96: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



5.1.2.6 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)”

i Online scanning in Config mode

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

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

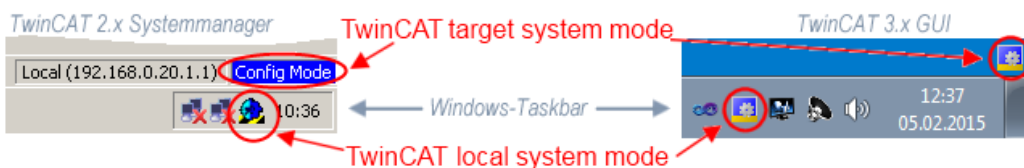


Fig. 97: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

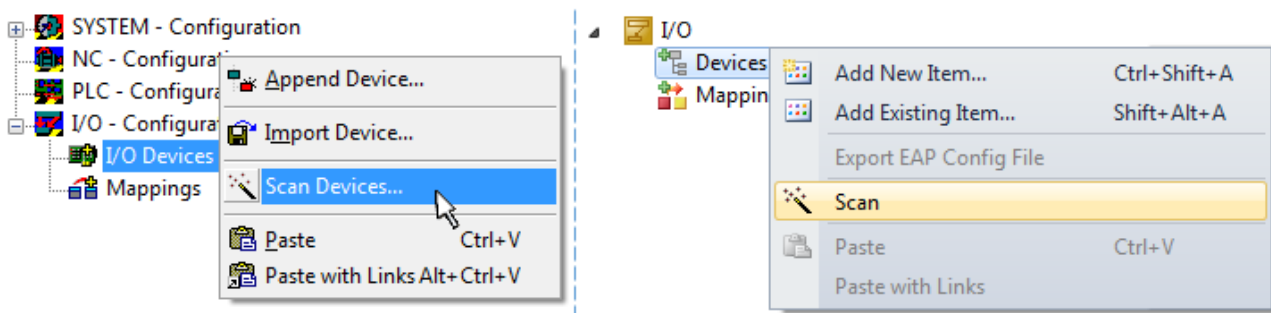


Fig. 98: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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

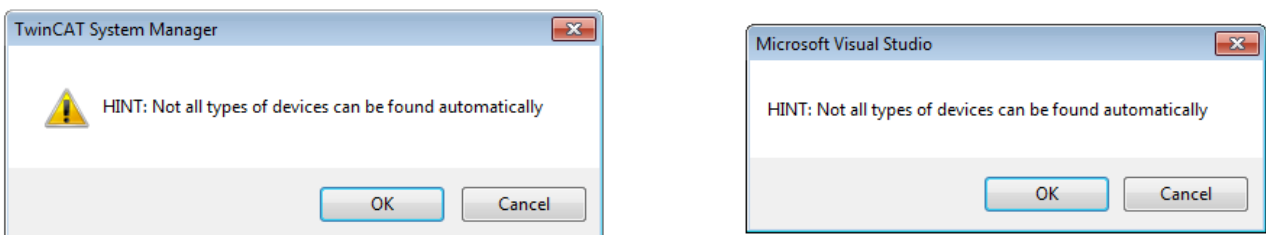


Fig. 99: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

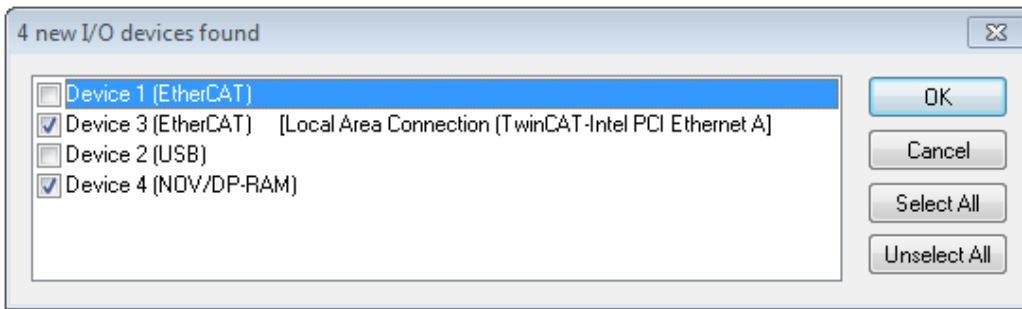


Fig. 100: Detected Ethernet devices

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](#) [▶ 67].

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.

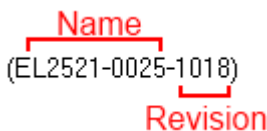


Fig. 101: Example default state

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](#) [▶ 87] 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:

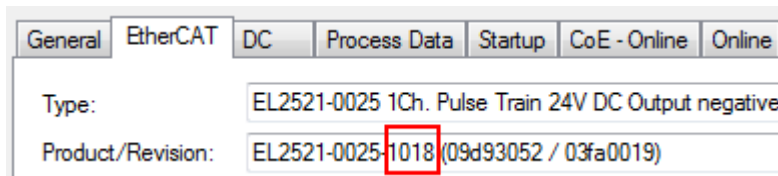


Fig. 102: Installing EtherCAT terminal with revision -1018

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 [▶ 87](#) 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:

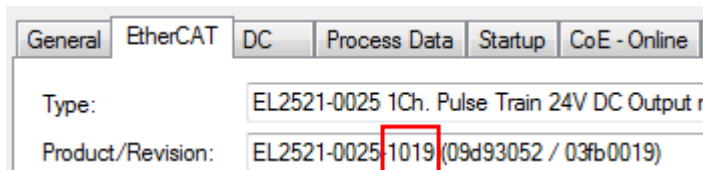


Fig. 103: Detection of EtherCAT terminal with revision -1019

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.

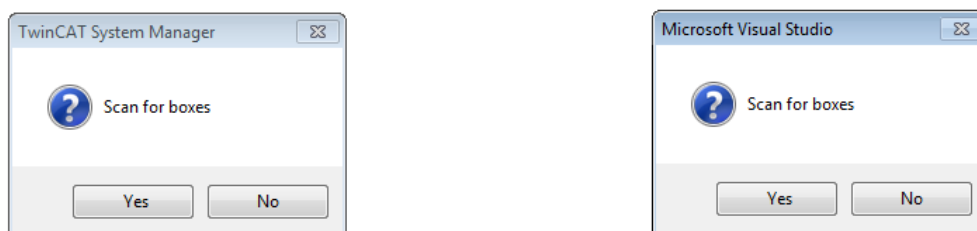


Fig. 104: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

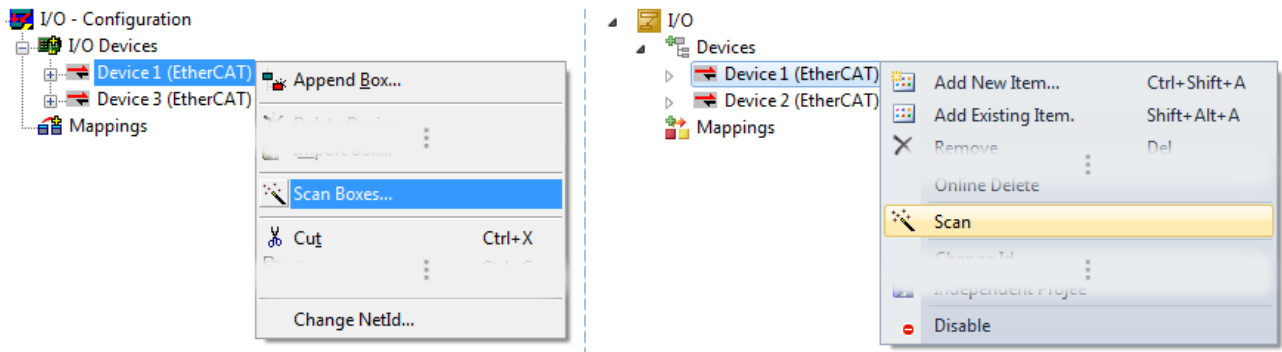


Fig. 105: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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. 106: Scan progress exemplary by TwinCAT 2

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



Fig. 107: 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. 108: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 109: 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”.

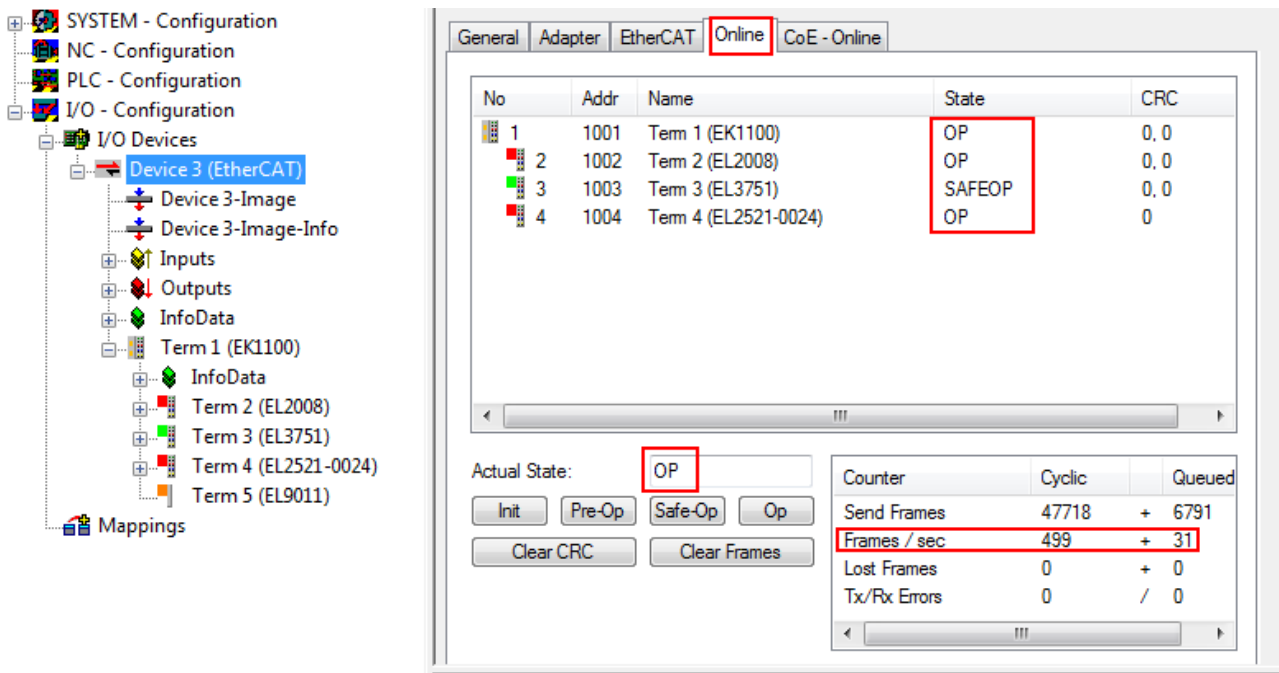


Fig. 110: Online display example

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 \[► 77\]](#).

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.

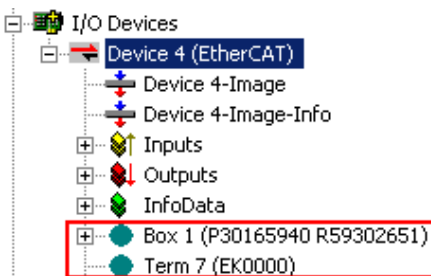


Fig. 111: Faulty identification

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. 112: 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.

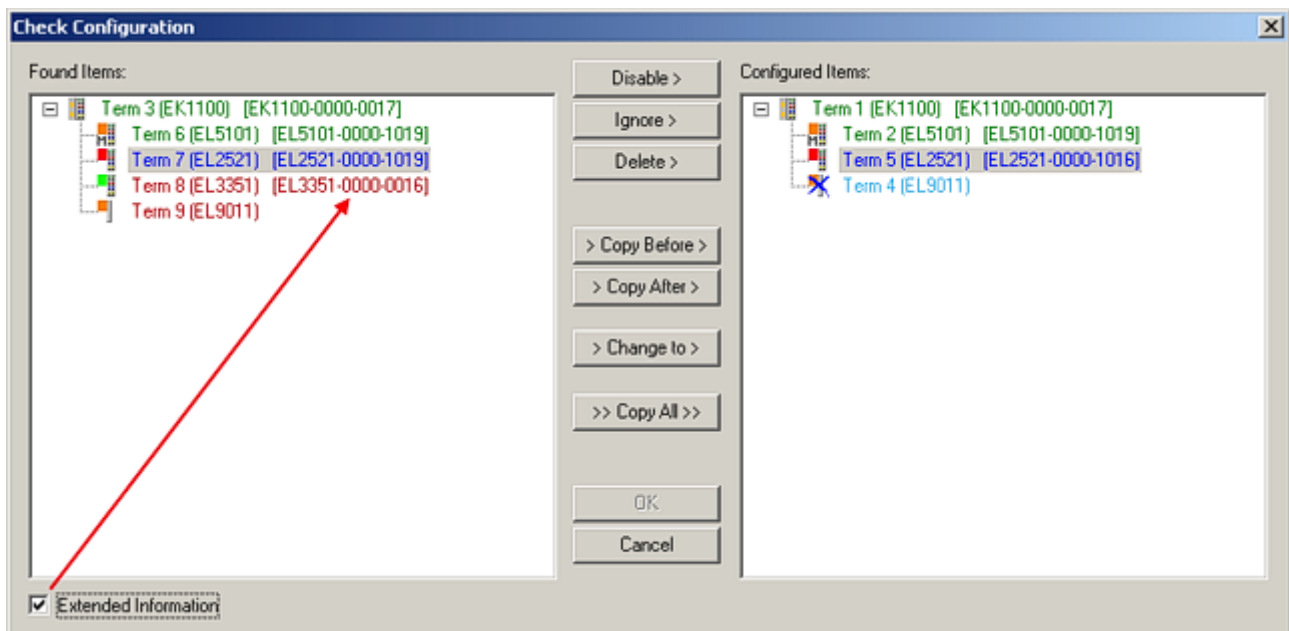


Fig. 113: Correction dialog

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

Colour	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. <p>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.</p> <p>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.</p>

● Device selection based on revision, compatibility

i 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. 114: 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 parameterised as follows: linking with the task, CoE/DC settings, plug-in definition, startup settings, ...



Fig. 115: 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.

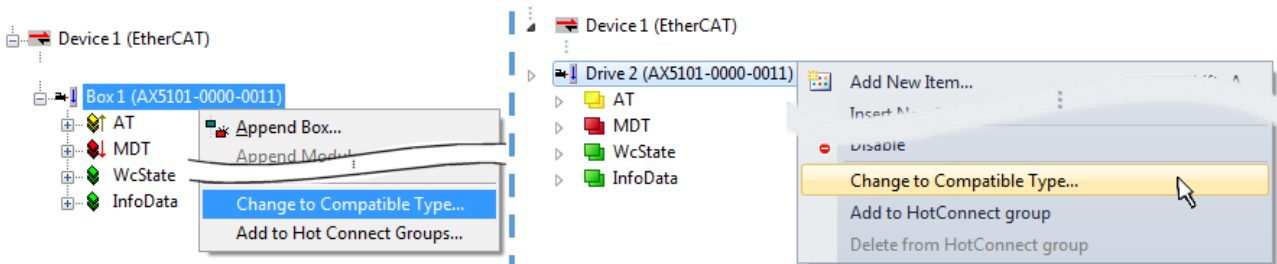


Fig. 116: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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*

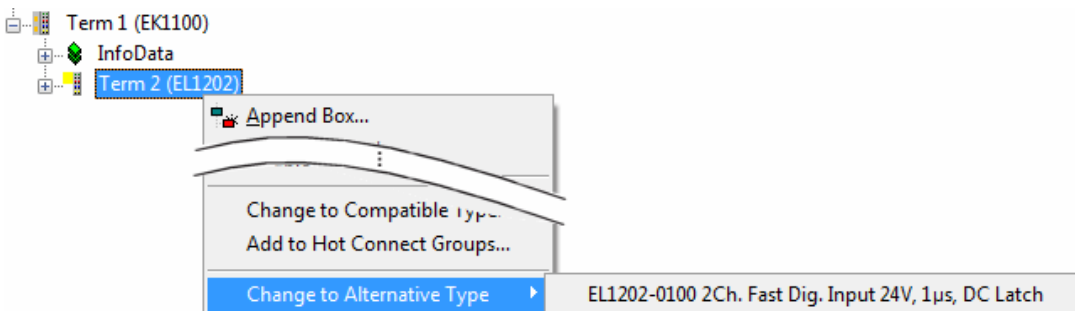


Fig. 117: TwinCAT 2 Dialog Change to Alternative Type

If called, the System Manager searches in the procured device ESI (in this example: EL 1202-0000) 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).

5.1.2.7 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).

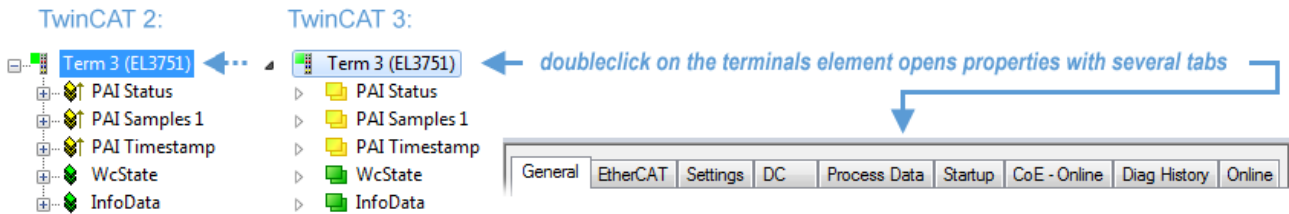


Fig. 118: Branch element as terminal EL3751

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

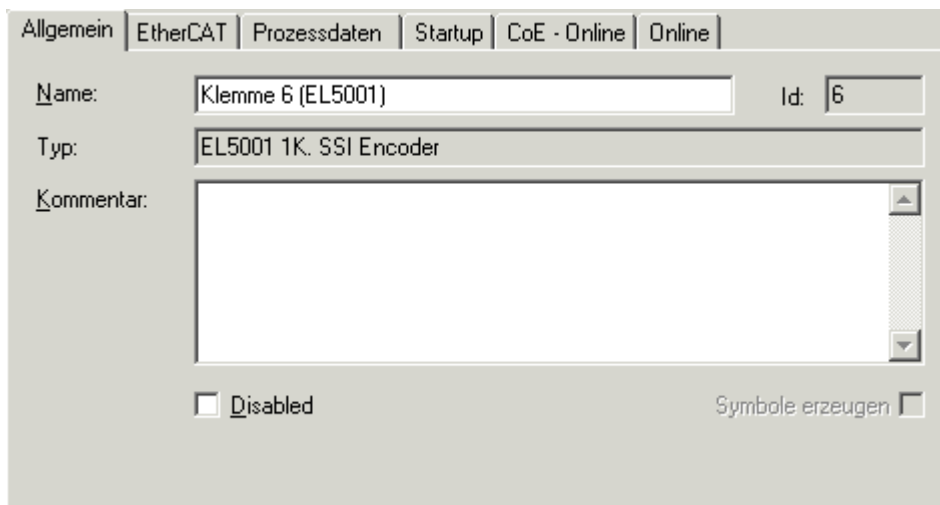


Fig. 119: "General" tab

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

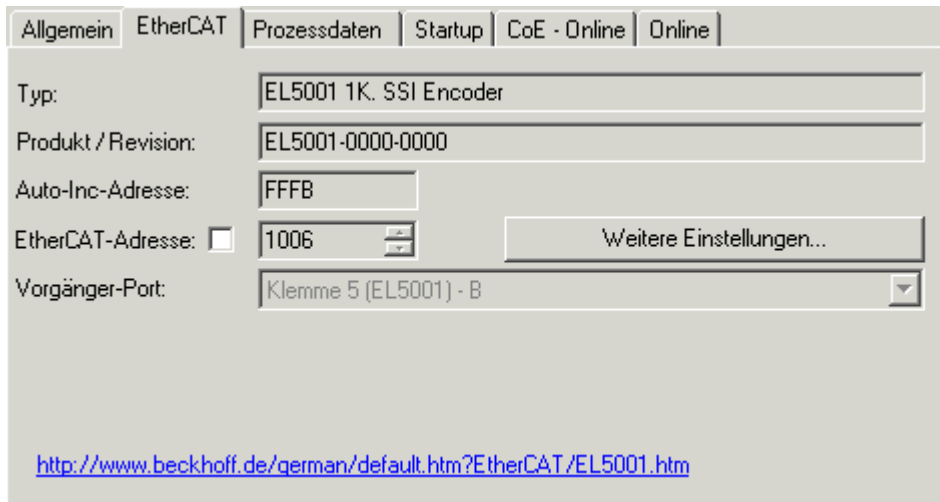


Fig. 120: „EtherCAT“ tab

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

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

“Process Data” tab

Indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (Process Data Objects, 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.

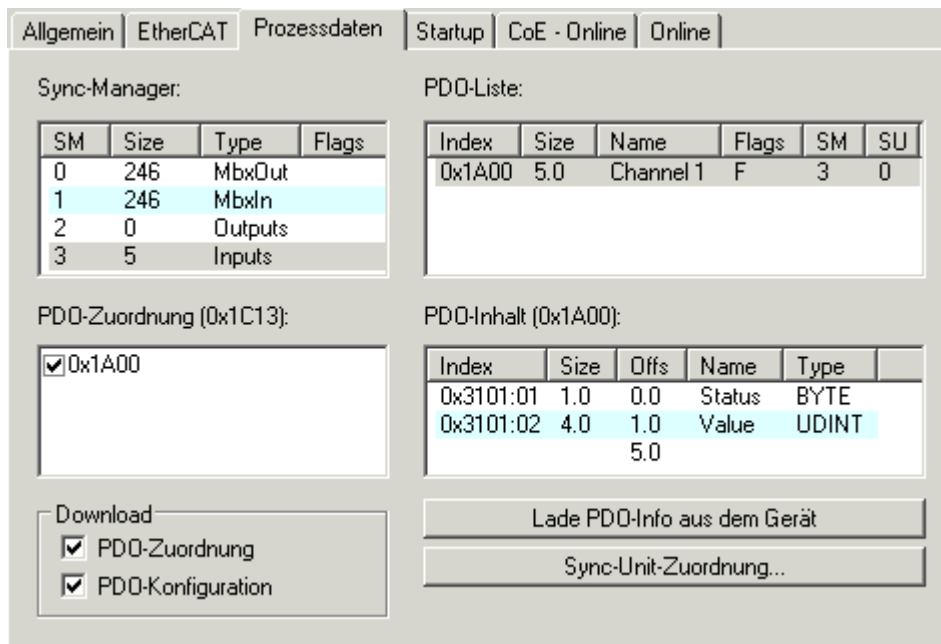


Fig. 121: "Process Data" tab

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").

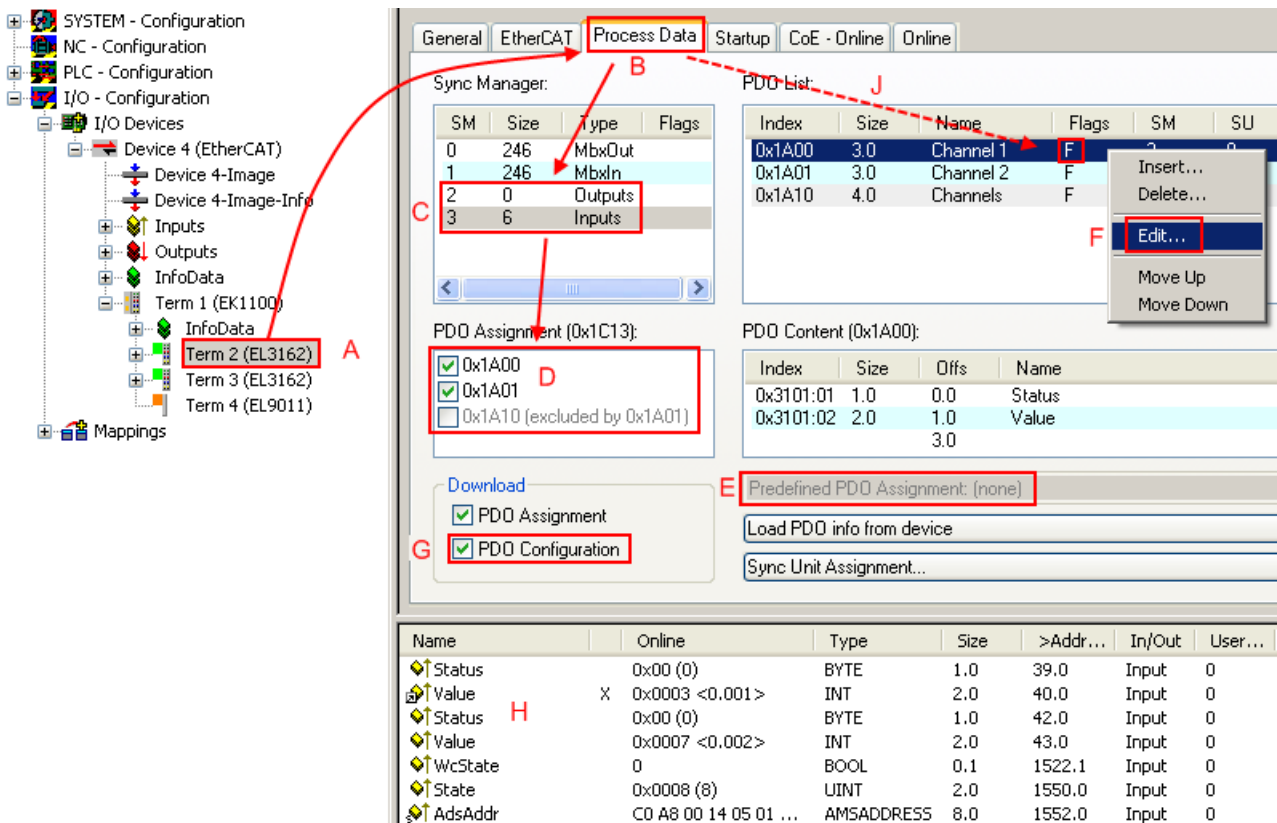


Fig. 122: Configuring the process data

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” log-ger message: This error message (“invalid SM IN cfg” or “invalid SM OUT cfg”) also indicates the reason for the failed start.

A detailed description [► 98] 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.

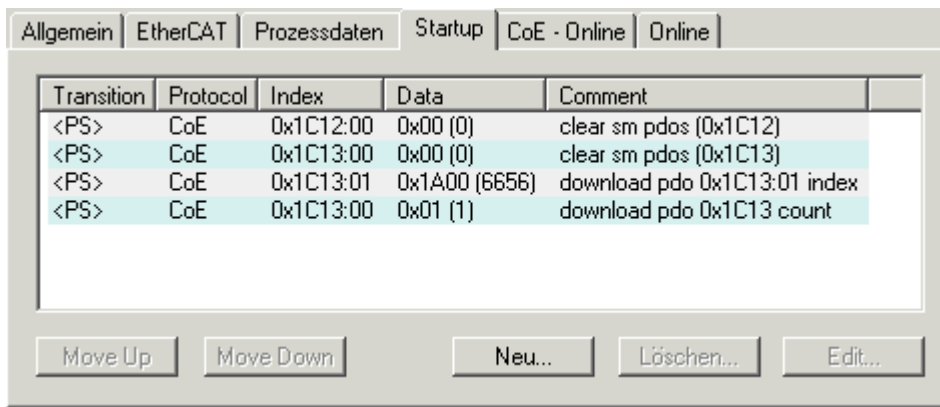


Fig. 123: „Startup“ tab

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.

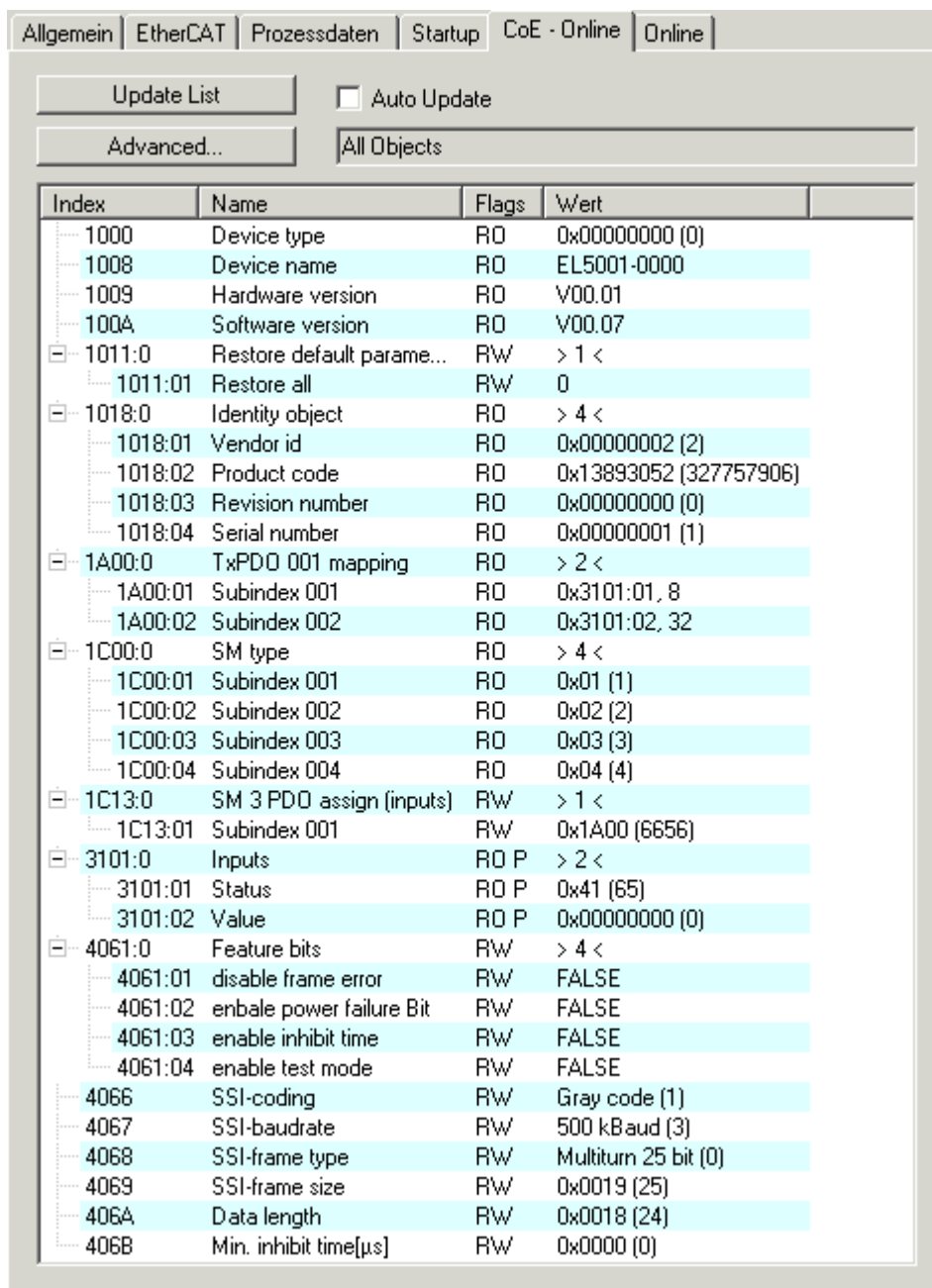


Fig. 124: "CoE – Online" tab

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.

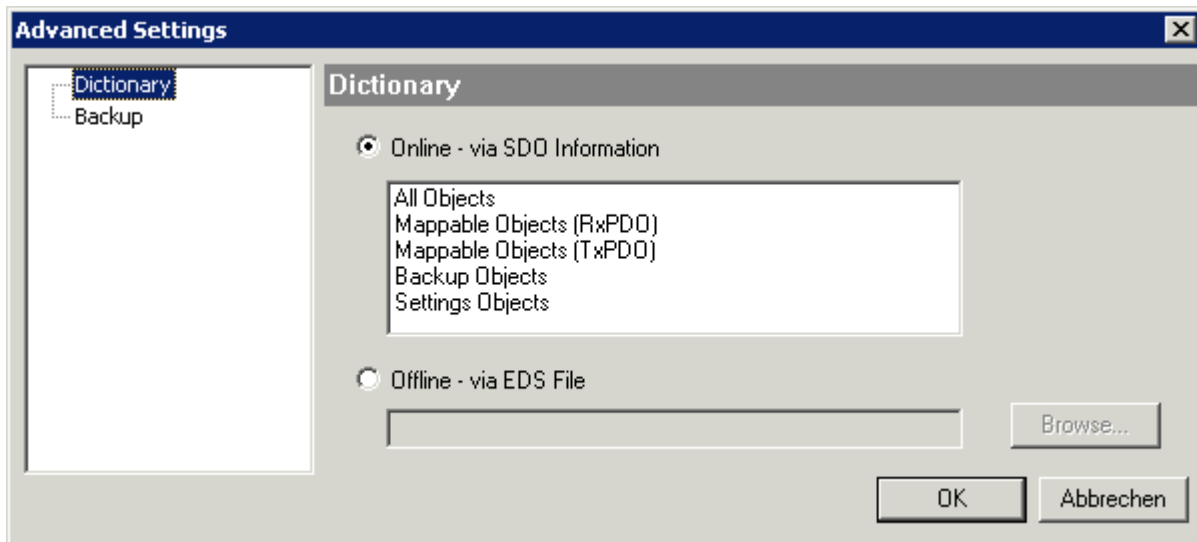


Fig. 125: Dialog "Advanced settings"

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

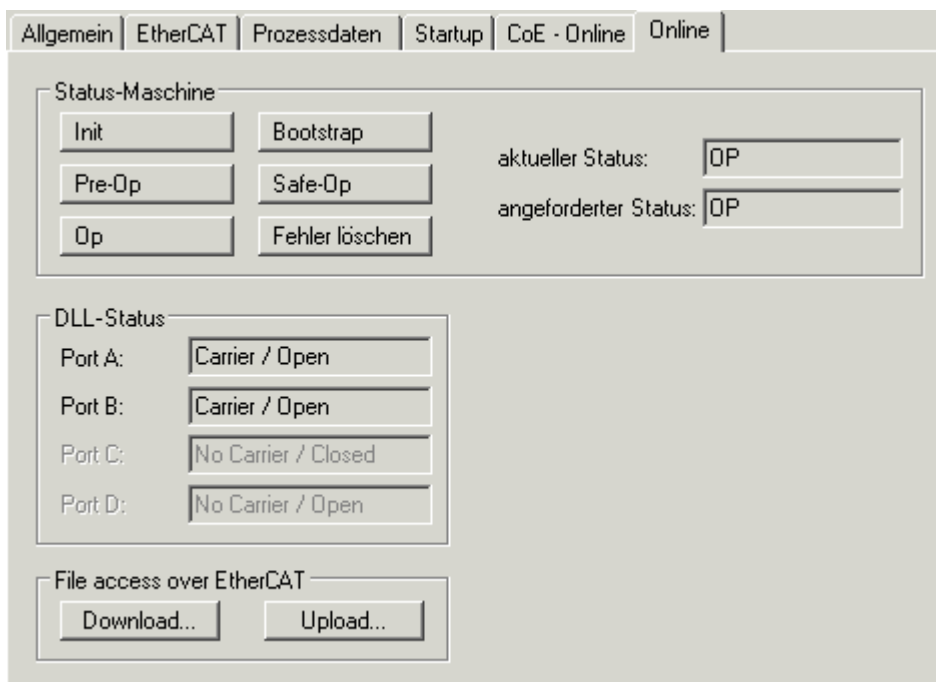


Fig. 126: „Online“ tab

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)

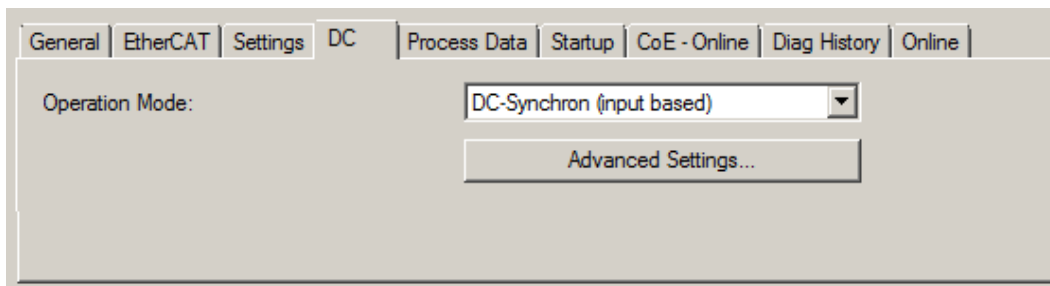


Fig. 127: "DC" tab (Distributed Clocks)

- 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 are specified on <http://infosys.beckhoff.com>:

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

5.1.2.7.1 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 \[▶ 96\]](#)),
 - 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 \[► 93\]](#) 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 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.

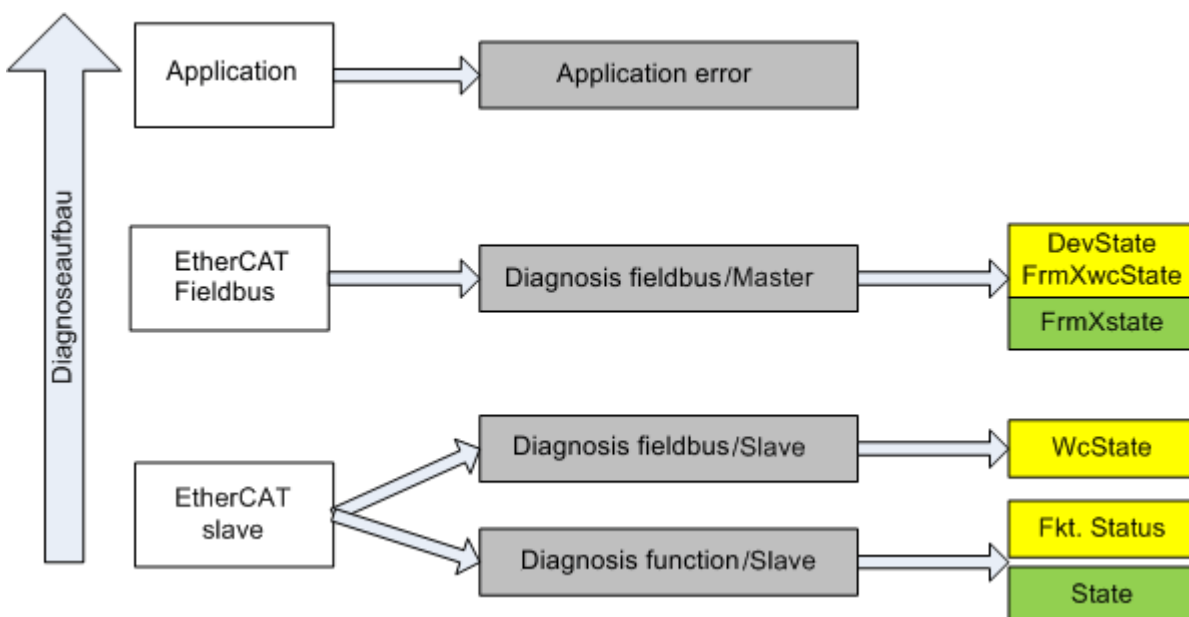


Fig. 128: Selection of the diagnostic information of an EtherCAT Slave

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.

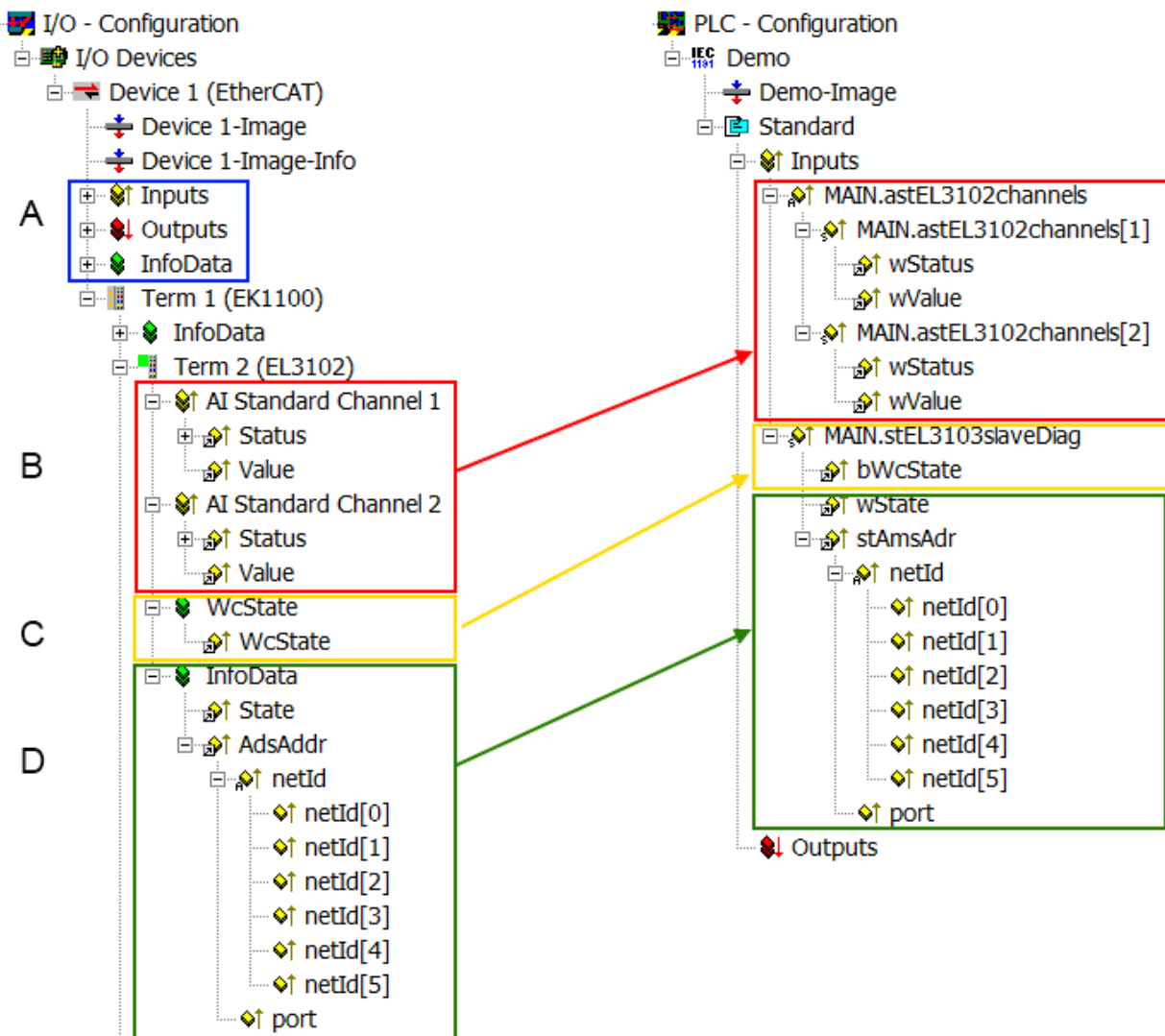


Fig. 129: Basic EtherCAT Slave Diagnosis in the PLC

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".

Index	Name	Flags	Value
6010:0	AI Inputs Ch.2	RO	> 17 <
6401:0	Channels	RO	> 2 <
8000:0	AI Settings Ch.1	RW	> 24 <
8000:01	Enable user scale	RW	FALSE
8000:02	Presentation	RW	Signed (0)
8000:05	Siemens bits	RW	FALSE
8000:06	Enable filter	RW	FALSE
8000:07	Enable limit 1	RW	FALSE
8000:08	Enable limit 2	RW	FALSE
8000:0A	Enable user calibration	RW	FALSE
8000:0B	Enable vendor calibration	RW	TRUE

Fig. 130: EL3102, CoE directory

● EtherCAT System Documentation

i 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.

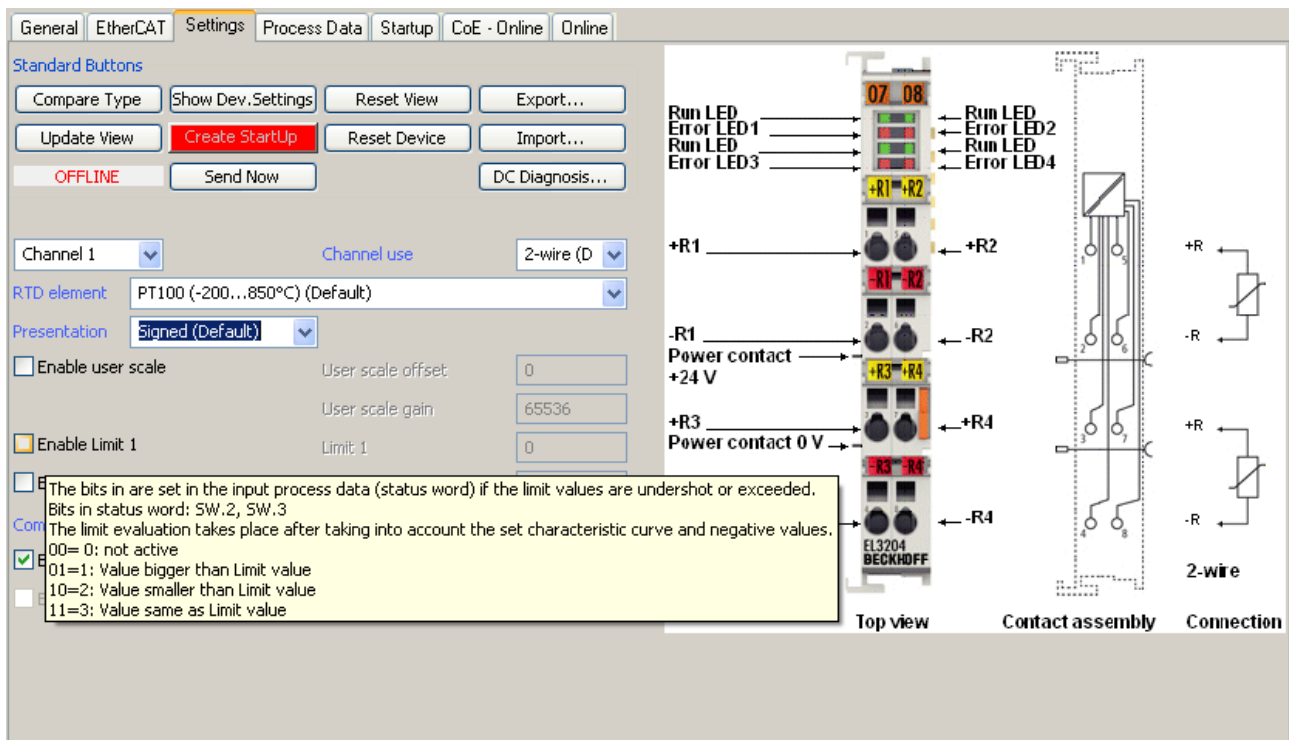


Fig. 131: Example of commissioning aid for a EL3204

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 \[► 20\]](#)" 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.

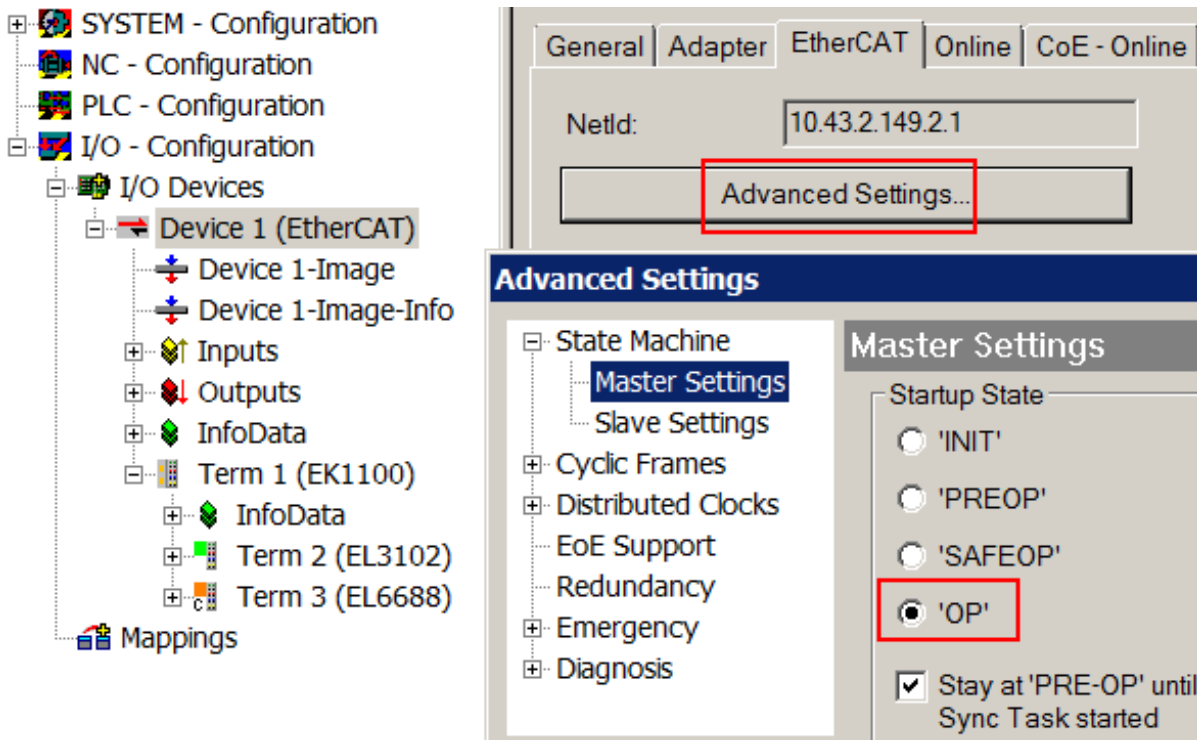


Fig. 132: Default behaviour of the System Manager

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

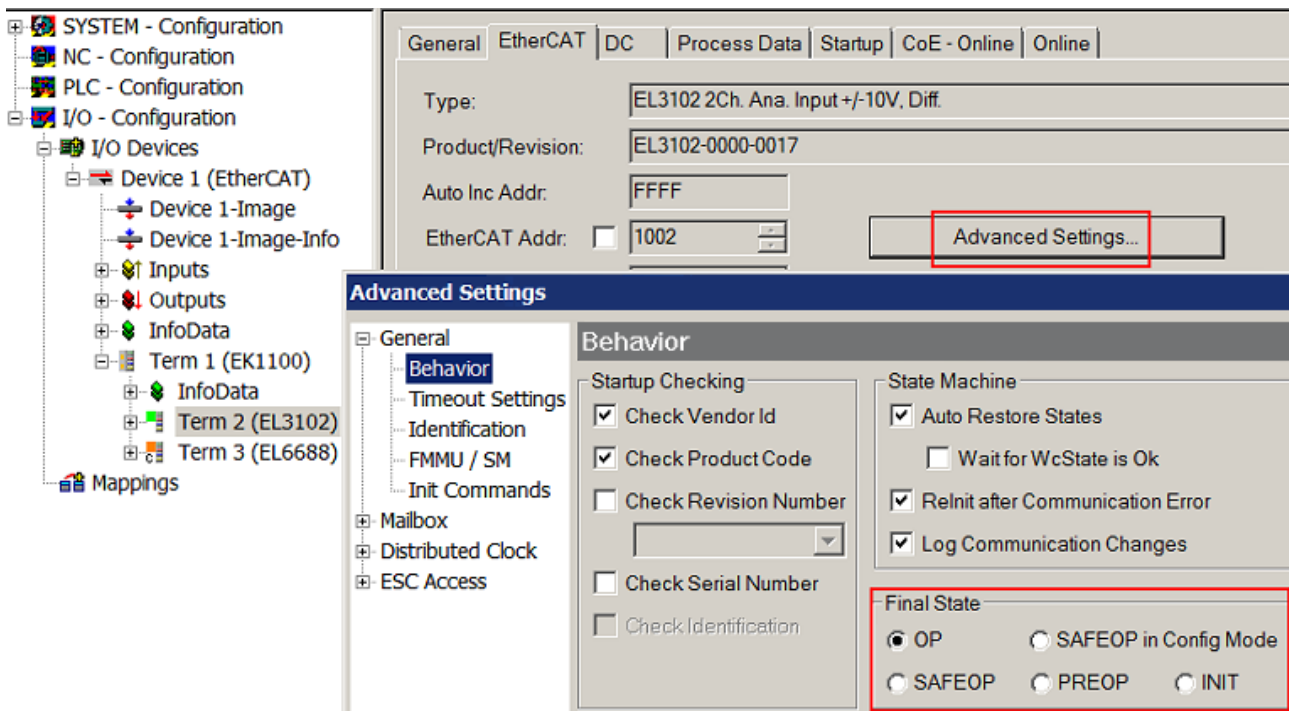


Fig. 133: Default target state in the Slave

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.

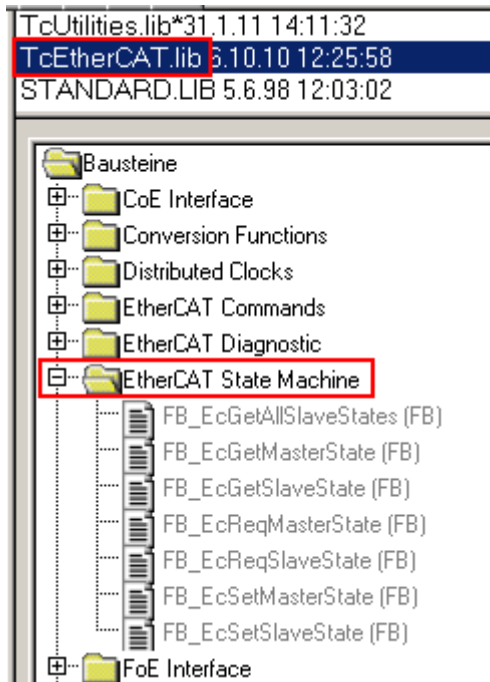


Fig. 134: PLC function blocks

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. 135: 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:

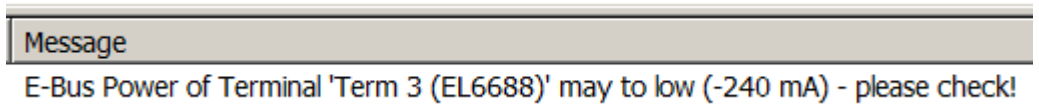


Fig. 136: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

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

5.3 Basic function principles

● RMS value (rms) specifications



All AC value specifications in this documentation such as RMS specifications (r_{ms}) refer to a 50/60 Hz 3-phase mains network with a sinusoidal waveform.

Introduction

The EL3783 has 6 analog input channels which are measured simultaneously in their respective measuring ranges. The 3 analog input channels for current measurement have different measuring ranges. These can be set via CoE or selected automatically by the EL3783 on the basis of the momentary measured values.

Each channel has an anti-aliasing low-pass filter with a limit frequency of 5.5 KHz. However, the high sampling rate of 20 KSps also allows frequency components up to higher than 9 KHz to be detected. Further information is available upon request.

● Using the sample programs



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

Automatic current range switching

The EL3783 features automatic current range switching. This can be configured via the sub-indices in object F80D [▶ 139].

During a switching event, processes within the terminal lead to measured data lying outside the specified accuracy for about 250 μ s. This short-term inaccuracy is signaled to the PLC via Object F603 [▶ 141]. Depending on the application there must be a reaction to this on the software side.

In order to use this function, it must be activated via the CoE object "F80D:11 [▶ 139] Current range" and configured as desired. Various dynamics are pre-programmed according to which the switching takes place quickly (Dynamic 0) or slowly (Dynamic 3).

The following sample project (see appendix) shows amongst other things how a conversion of the PDO values can be carried out, including automatic range switching.

Sample program (<https://infosys.beckhoff.com/content/1033/el3783/Resources/zip/3264881931.zip>)

Declaration of variables and source code:

```

"
EL3783_HCBorderCnt := 0;
EL3783_5ARange := FALSE;
CF := 1.8415054;

FOR i:= 0 TO 19 DO

    IF i = EL3783_HCRange[EL3783_HCBorderCnt] THEN
        EL3783_5ARange := NOT EL3783_5ARange;
        IF EL3783_5ARange THEN
            CF := 9.207527;
        ELSE
            CF := 1.8415054;
        END_IF
        EL3783_HCBorderCnt := EL3783_HCBorderCnt + 1;
    END_IF

    EL3783_UL1_DC20[i] := INT_TO_LREAL(EL3783_UL1_DC_020[i])
* 737.2575 / 32768;
    EL3783_UL2_DC20[i] := INT_TO_LREAL(EL3783_UL2_DC_020[i])
* 737.2575 / 32768;
    EL3783_UL3_DC20[i] := INT_TO_LREAL(EL3783_UL3_DC_020[i])

```

```
* 737.2575 / 32768;

      EL3783_IL1_DC20[i] := INT_TO_LREAL(EL3783_IL1_DC_020[i])
* CF / 32768;
      EL3783_IL2_DC20[i] := INT_TO_LREAL(EL3783_IL2_DC_020[i])
* CF / 32768;
      EL3783_IL3_DC20[i] := INT_TO_LREAL(EL3783_IL3_DC_020[i])
* CF / 32768;

END_FOR
"
```

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

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

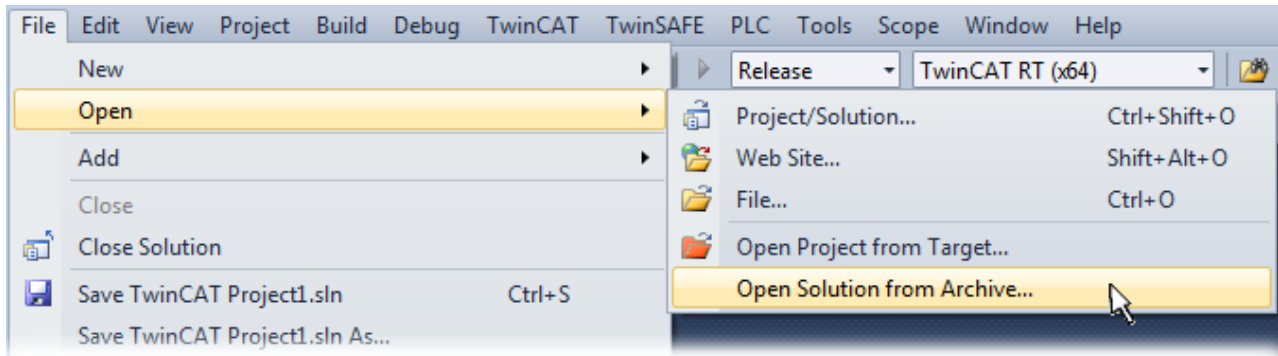


Fig. 137: Opening the *.tnzip archive

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

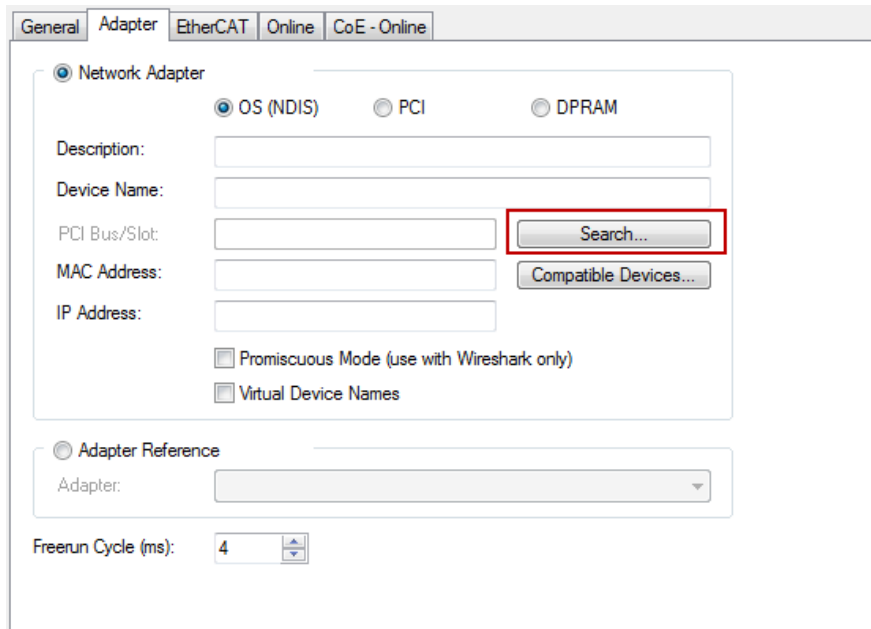
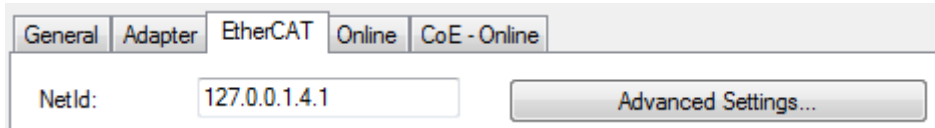


Fig. 138: Search of the existing HW configuration for the EtherCAT configuration of the example

- Checking NetId: the "EtherCAT" tab of the EtherCAT device shows the configured NetId:



The first 4 numbers have to be identical with the project NetId of the target system. The project NetId can be viewed within the TwinCAT environment above, where a pull down menu can be opened to choose a target system (by clicking right in the text field). The number blocks are placed in brackets there next to each computer name of a target system.

- Modify the NetId: By right clicking on "EtherCAT device" within the solution explorer a context menu opens where "Change NetId..." have to be selected. The first 4 numbers of the NetId of the target computer have to be entered; the both last values are 4.1 usually.
Example:

- NetId of project: myComputer (123.45.67.89.1.1)
- Entry via „Change NetId...“: 123.45.67.89.4.1

Measuring ranges 1 A, 5 A

Fixed current ranges up to 1.3 times the respective rated value for sinusoidal currents. The 1 A range is the default value on delivery from the factory.

DYN 0

By default the terminal is in the 1 A current measuring range. The terminal switches over to the 5 A measuring range as soon as the current in one of the three channels exceeds the permanently coded threshold value $T_{1A \rightarrow 5A}$ of approx. 1.74 A (momentary). The switchover takes 250 μ s at the most. Switching back takes place as soon as the maximum values of a full half-wave on all the current channels lie below the threshold value $T_{5A \rightarrow 1A}$ of approx. 1.69 A (momentary).

DYN1, DYN2, DYN3

By default the terminal is in the 1 A current measuring range. The terminal determines the maximum value of the current separately for each half-wave and each current channel. The determined maximum values

\hat{a}_{max} for each half-wave are averaged channel-by-channel with an exponential filter of the form.

$$S_N = S_{N-1} * \alpha + |\hat{a}_{max}| * (1 - \alpha)$$

The terminal switches over to the 5 A measuring range as soon as the filtered value S_N exceeds the threshold value $T_{1A \rightarrow 5A}$. The difference between the characteristics for the switchover is due to the choice of the respective coefficients α . The coefficient is formed by $\alpha = 2^p$, where p is the constant from the name of the setting ("DYN p "). In turn, it follows that the setting variant DYN 3 requires more half-waves exceeding the threshold value in order to switch over than a smaller setting such as DYN 1.

Switching from the 5 A measuring range back to the 1 A measuring range takes place analogously with the aid of the filter. Switching back takes place as soon as the average of the maximum values of all the current channels lies below the threshold value $T_{5A \rightarrow 1A}$.

The settings DYN1, DYN2 and DYN3 are intended for current measurements in grids in which temporary current peaks occur. However, these peaks should not cause any direct switchover of the terminal. If there should be an error that results in a lengthy fault current, however, the terminal will switch over.

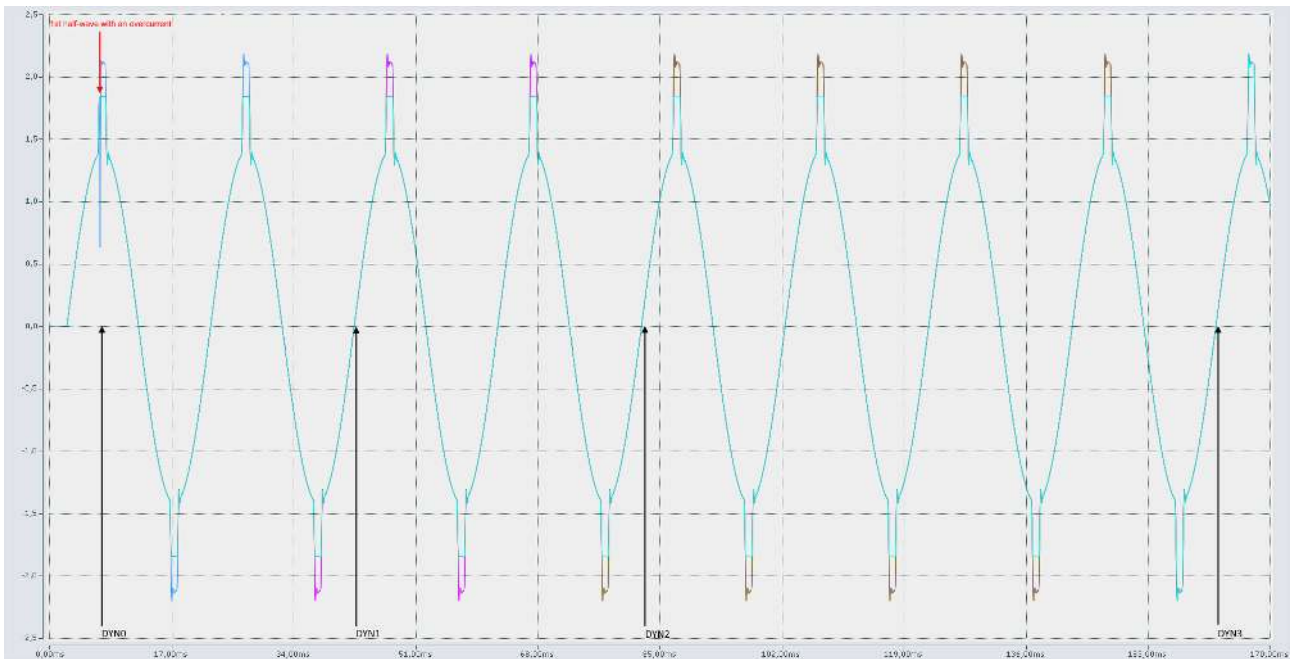


Fig. 139: Switchover characteristic, DYN0 - DYN3

Rated voltages, list of grids

The tables below show suitable rated grid voltages according to EN60664-1:2003 for the voltage measuring inputs of the EL3783

three-phase, 4-conductor grid with earthed protective conductor	
U_{L-N}	U_{L-L}
66 V	115 V
120 V	208 V
127 V	220 V
220 V	380 V
230 V	400 V
240 V	415 V
260 V	440 V
277 V	480 V
347 V	600 V
400 V*	690 V*

three-phase, 3-conductor grid, not earthed
U _{L-L}
66 V
115 V
120 V
127 V
200 V
220 V
230 V
240 V
260 V
277 V
347 V
380 V
400 V
415 V
440 V
480 V
500 V
577 V
600 V*

*) maximum rated voltage of the grid



Technical data

All technical specifications apply to "3-phase, 4-conductor system with earthed neutral conductor" unless explicitly stated otherwise (see also [Examples of measurement methods \[▶ 39\]](#)).



Changes in the CoE directory

In case of changes to the CoE default parameters, it is essential that corresponding values are entered in the start-up list, so that in the case of exchange the EL3783 operates again as foreseen in the application.

5.4 Oversampling terminals/boxes and TwinCAT Scope

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

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

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

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

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

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

EL3102

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

EL3702

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

Fig. 140: Oversampling PDO of the EL37xx series and in the comparison with EL31xx

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

5.4.1 TwinCAT 3 procedure

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

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

Recording a PLC Variable with the TwinCAT 3 – ScopeView

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

Step 1: Adding a project „Scope YT”

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

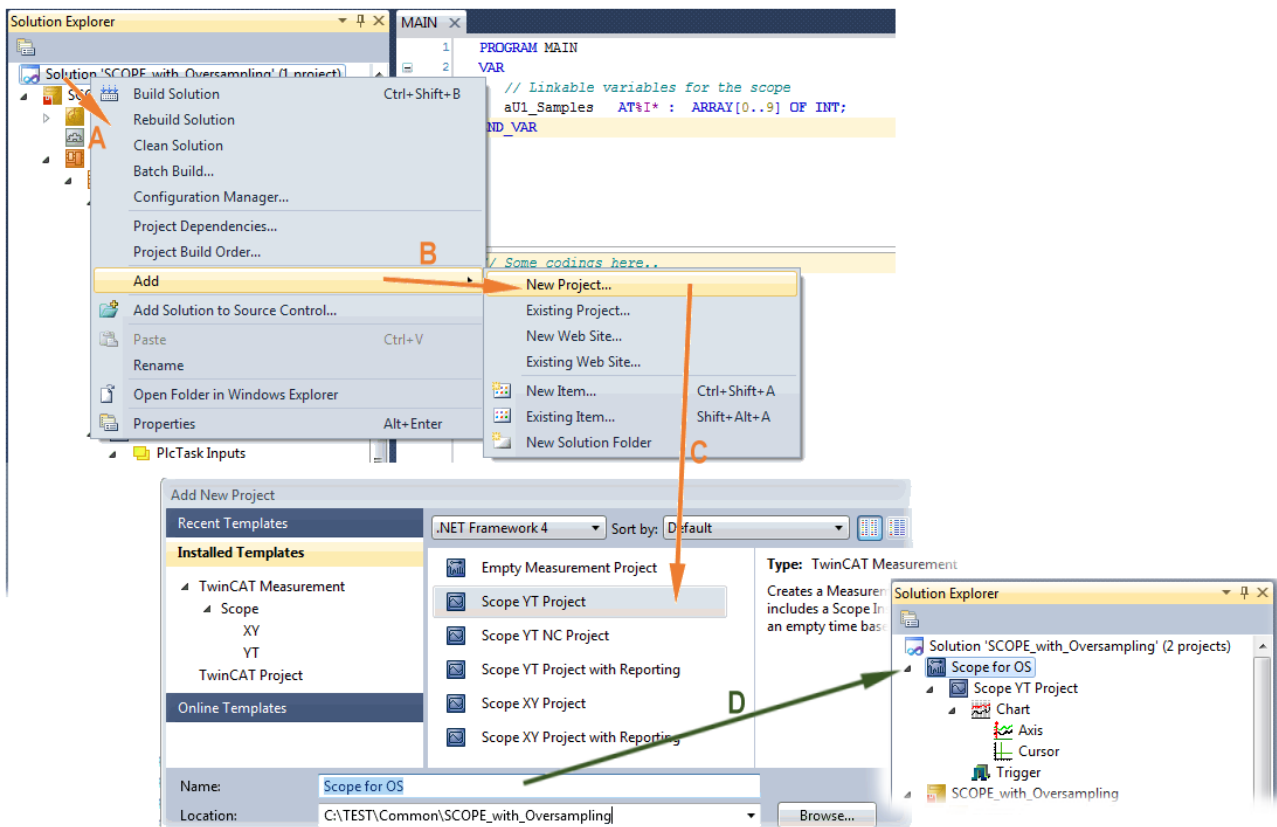


Fig. 141: Adding a Scope project into an already existing project

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

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

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

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

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

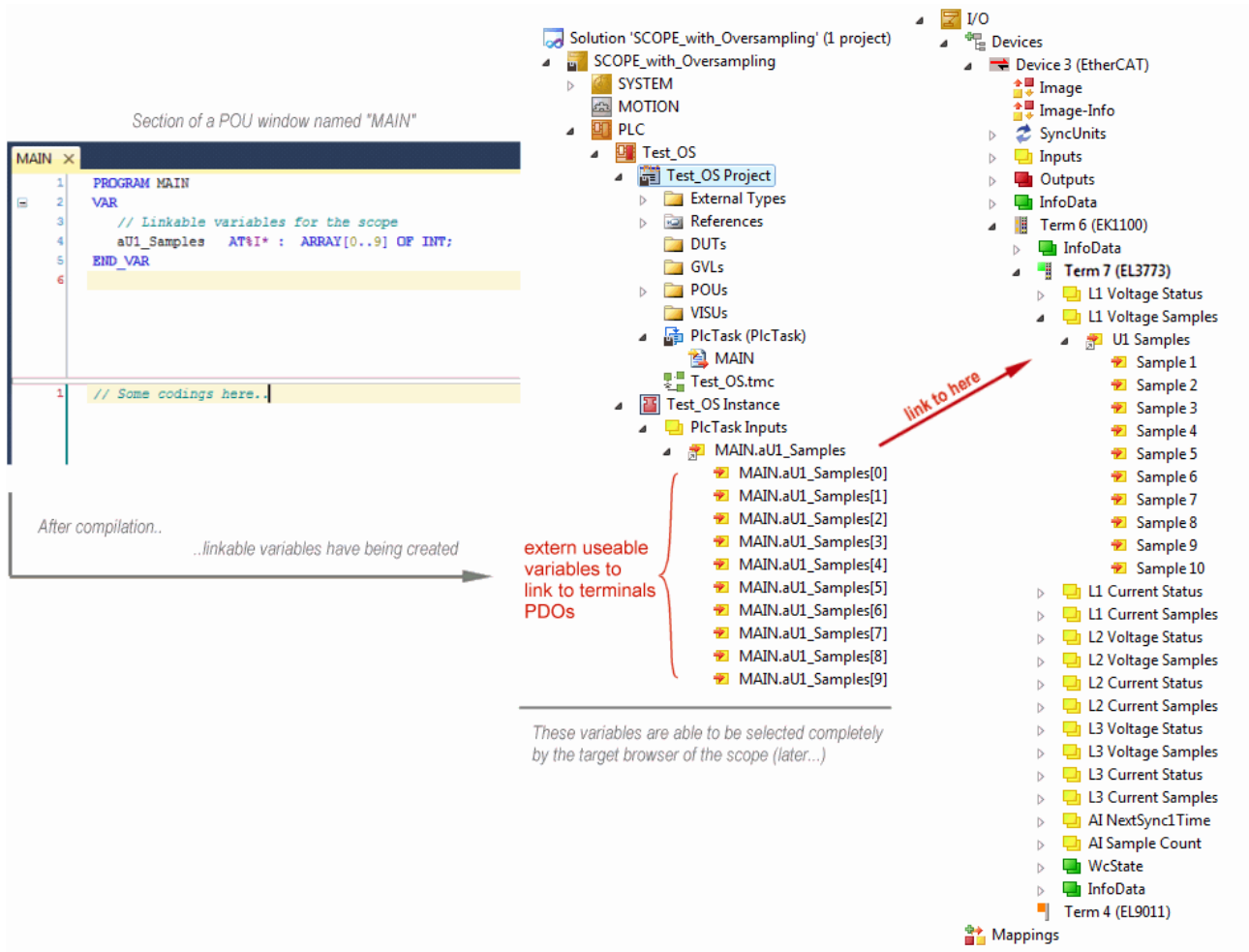


Fig. 142: Representation of a created PLC array variable („aU1_Samples“) to link with oversampling PDOs of EL3773

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

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

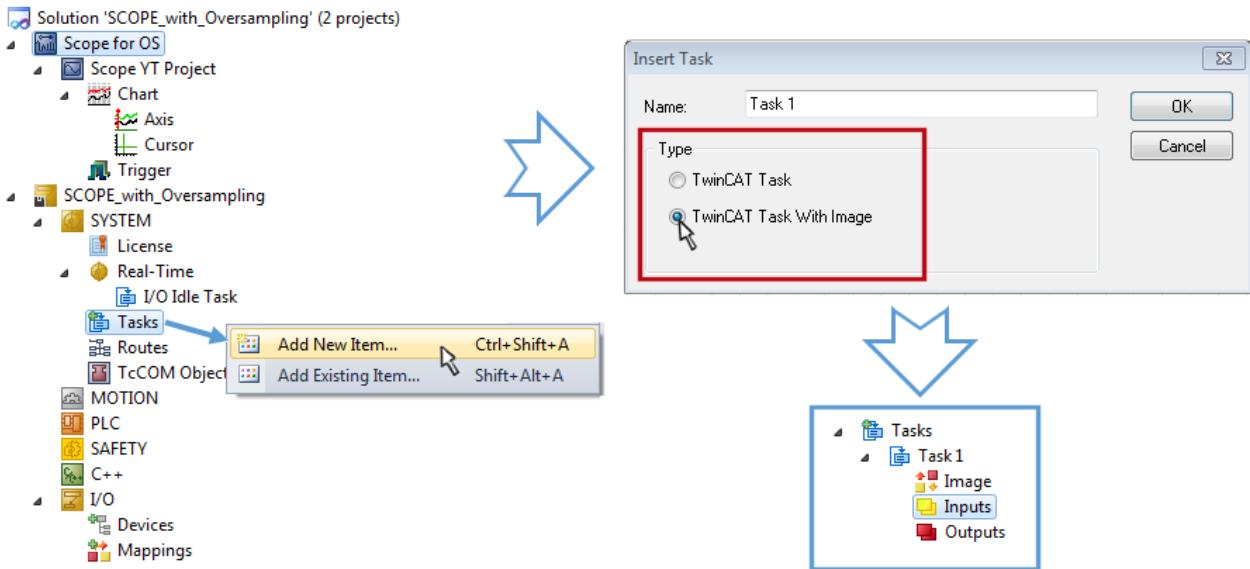


Fig. 143: Insertion of a free task

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

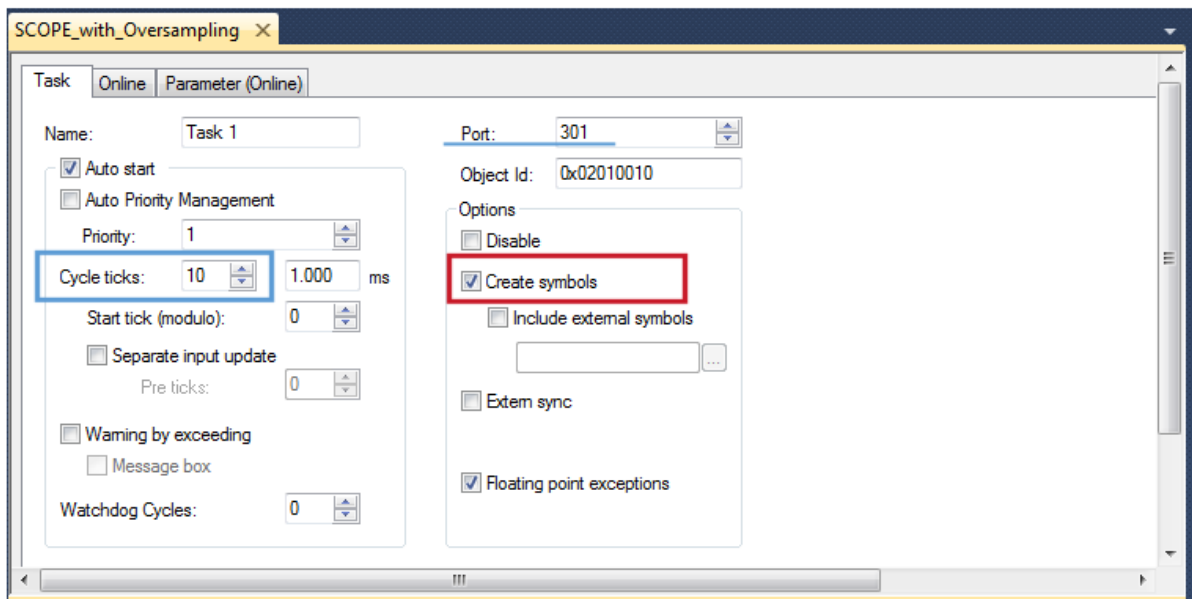


Fig. 144: Task property "Create symbols" must be activated

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

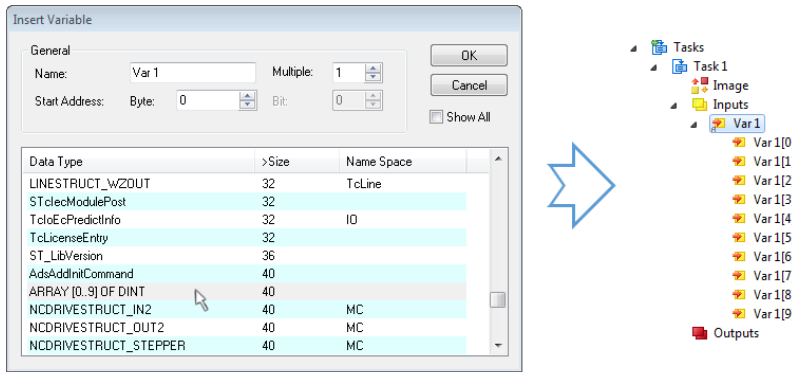


Fig. 145: Insertion of variable "Var 1" fitting to the oversampling (-factor)

Step 3: Linking an array variable with an oversampling PDO

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

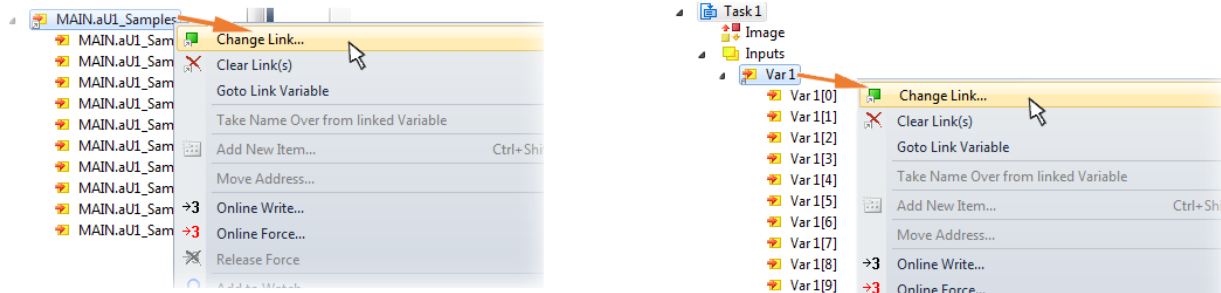


Fig. 146: Set up the link of the PLC array variable (left: for the last preceding paragraph Step 2a, right: for the last preceding paragraph Step 2b)

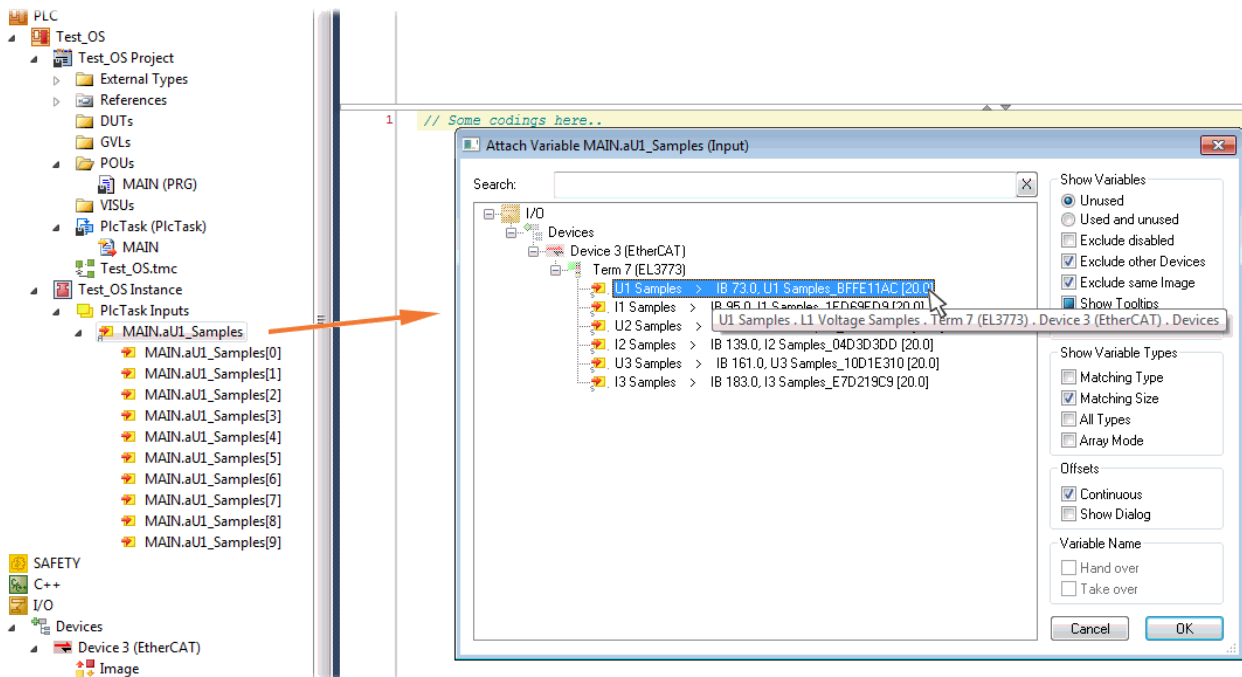




Fig. 147: Select the EL3773 PDO "L1 Voltage Samples" to create a link to the PLC array variable „aUI_Samples“

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

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

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

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

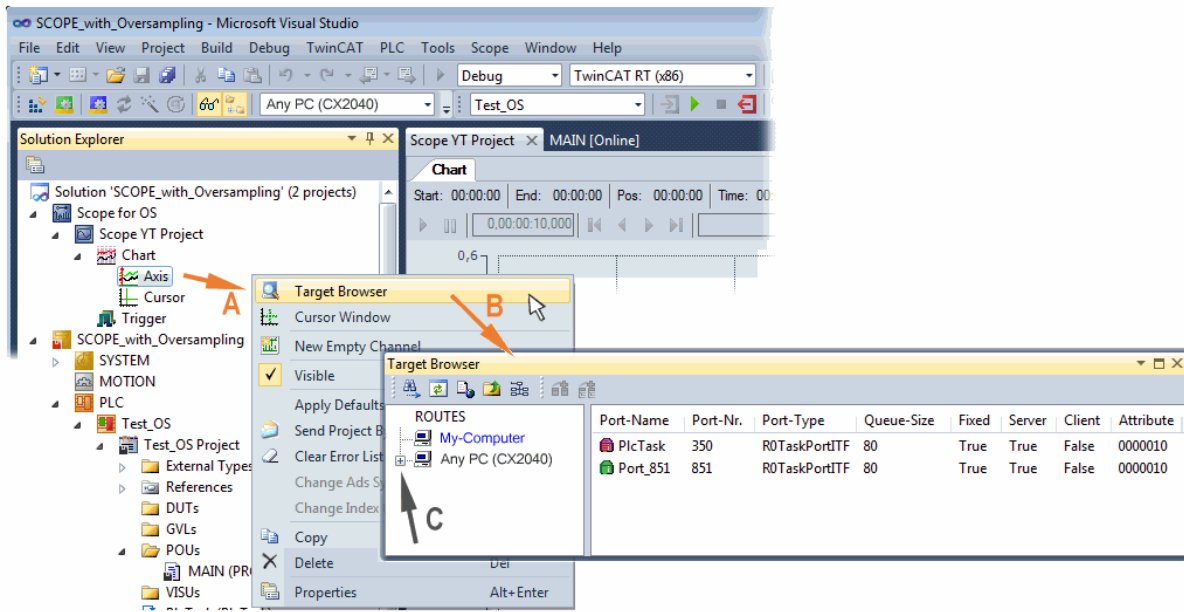


Fig. 148: Selection of the oversampling variable with the target browser

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

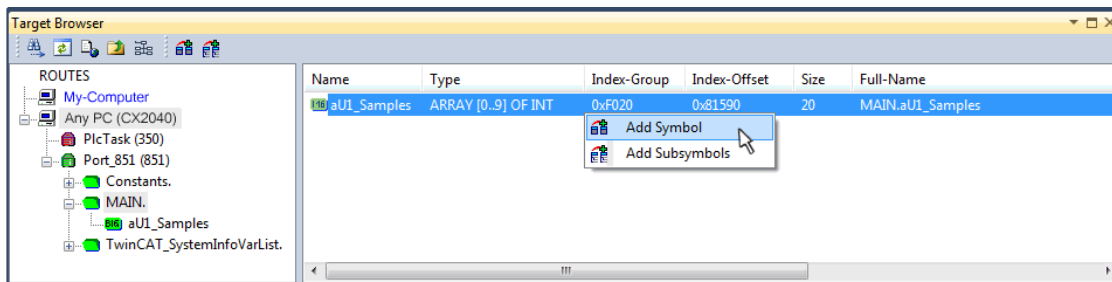


Fig. 149: Appending the variable "aUI_Samples" below "axis" within the scope project of the solution explorer

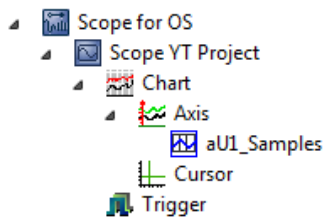
Variable don't appears into the target browser


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




"Enable Server Ports"

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



Now the program start has to be done with  formally although there's no program still yet. Using "Start

Recording"  the process data value of the oversampling PDO "L1 Voltage Samples " via the linked PLC array variable can be recorded time dependent now.

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

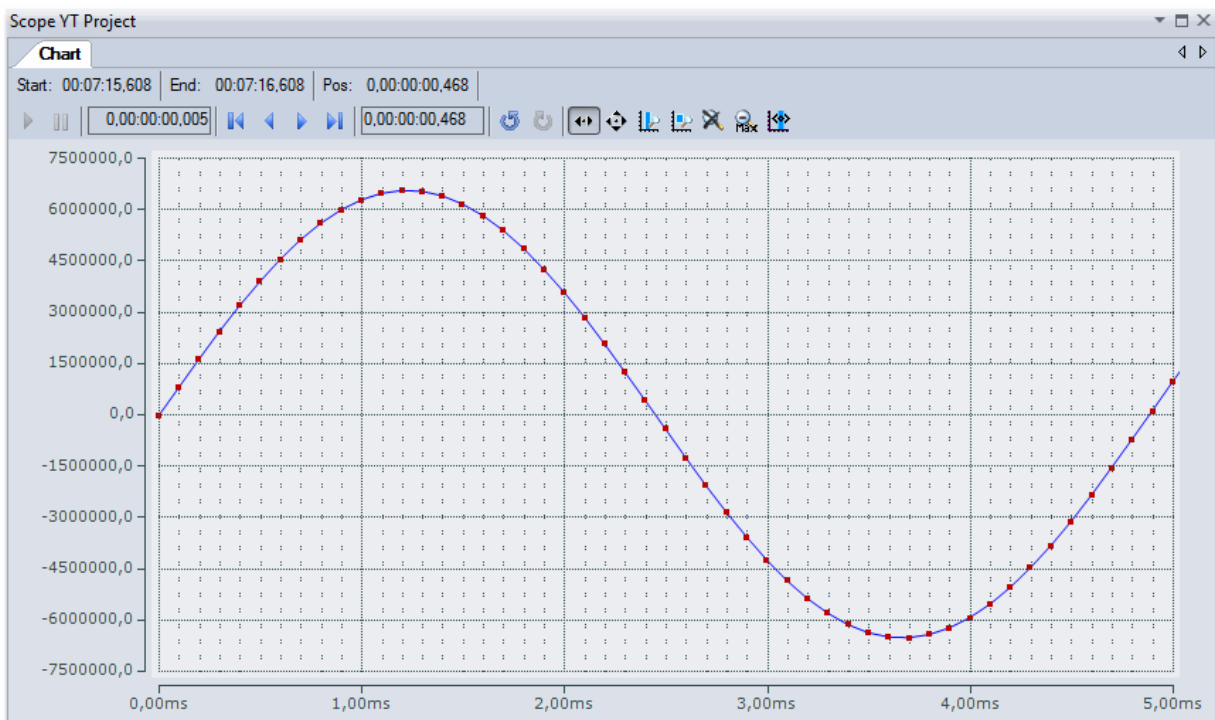




Fig. 150: Example of recording a sine signal with 10 x oversampling at 1 ms measurement cycle time

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

Proceeding with / via ADS alternatively

In former TwinCAT 3 versions (or a lower revision as specified in the [table \[▶ 112\]](#) above) the oversampling PDO of the respective oversampling able terminal/box can be made visible for the ScopeView by activation of the ADS server.

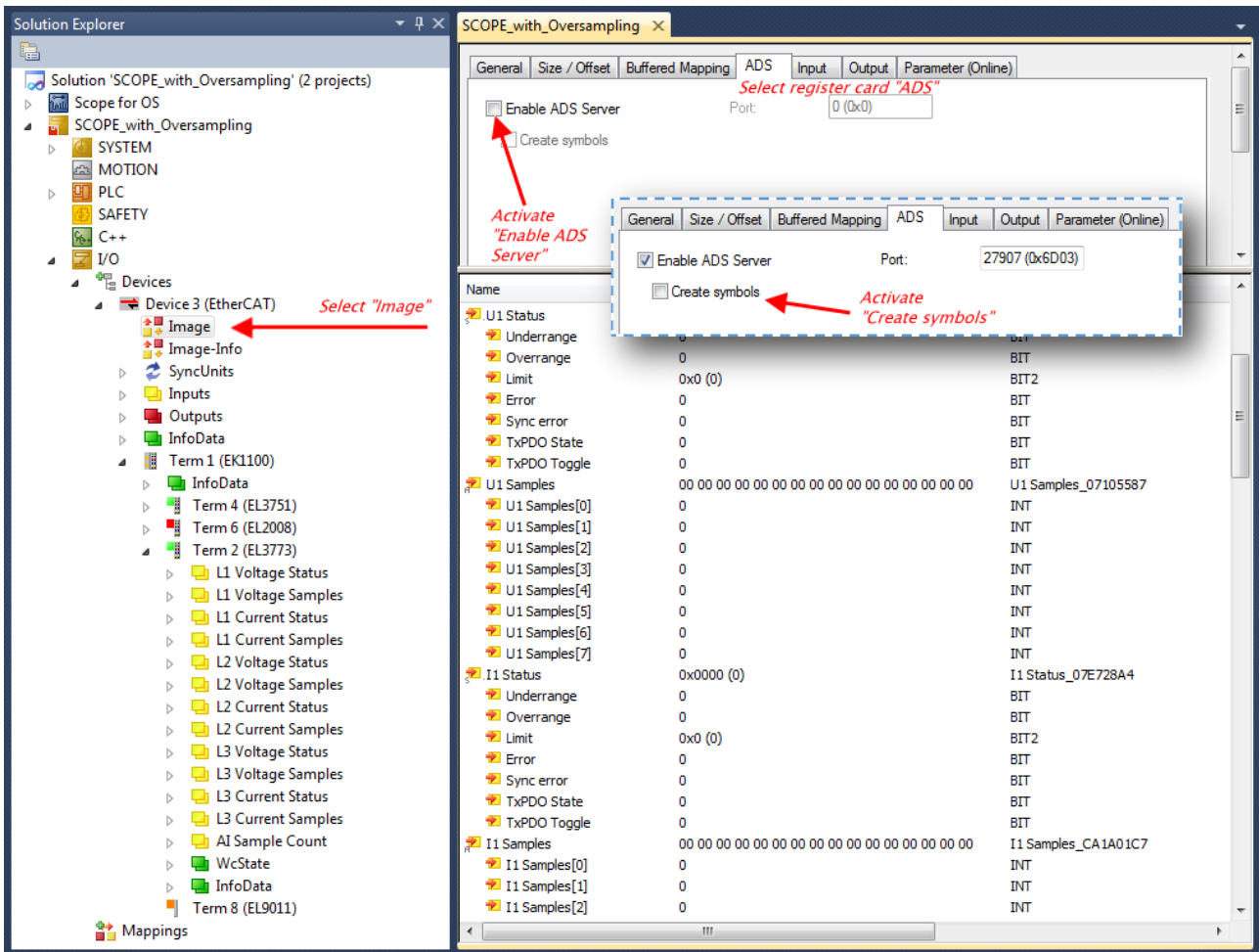


Fig. 151: Activation of the ADS server of the EtherCAT device (TwinCAT 3)

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

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

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

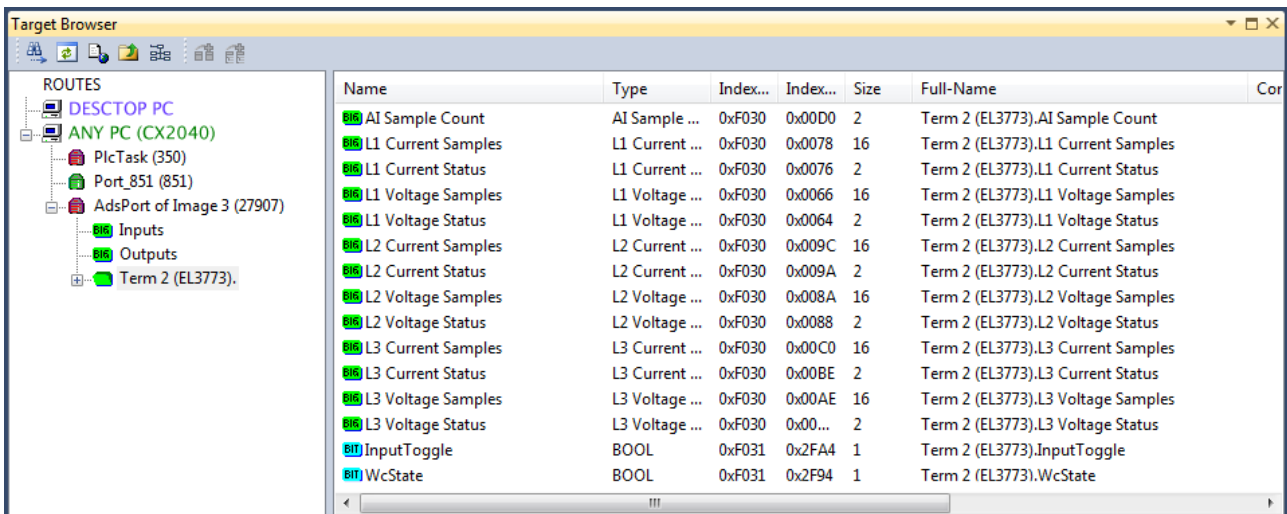
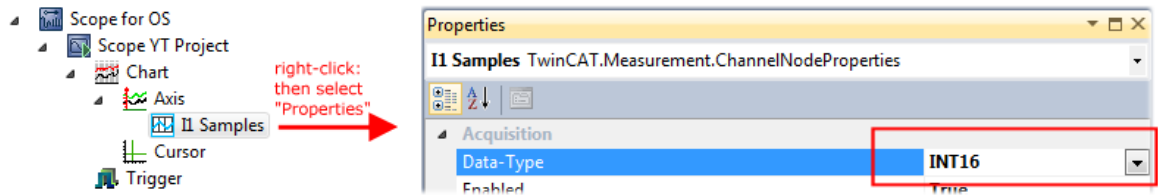


Fig. 152: Direct access to PDOs of the terminal by ScopeView

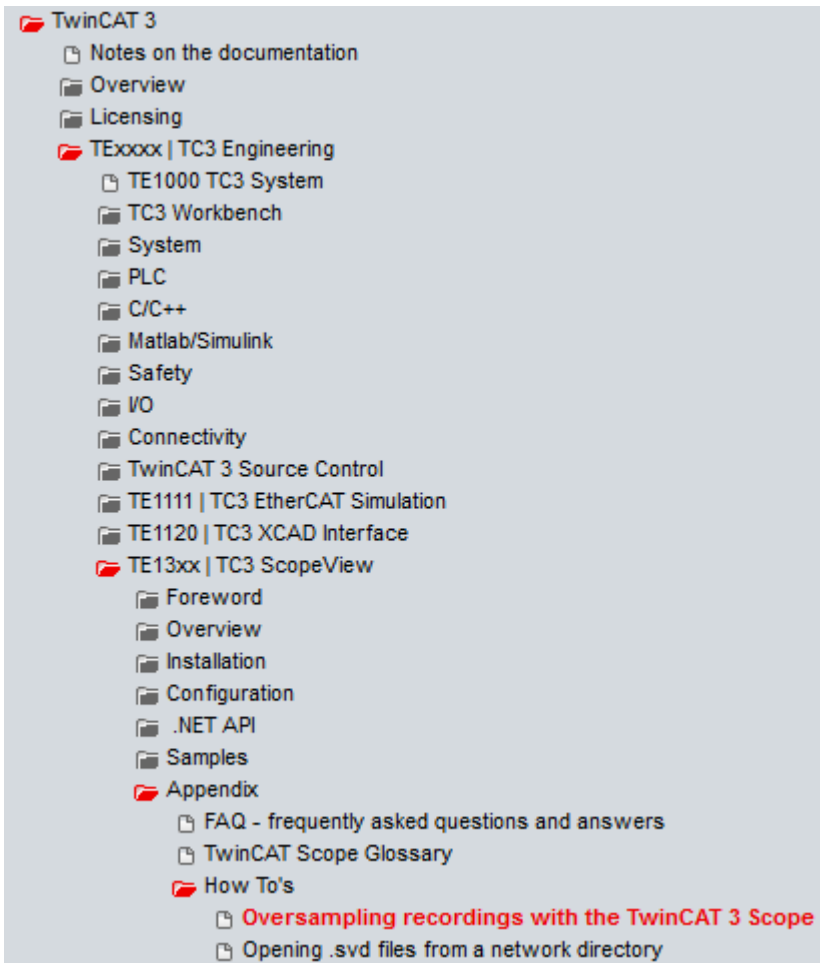
i Data type not valid

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



i TwinCAT 3: Activate the ADS Server of an EtherCAT device

Also see Beckhoff Information System:



5.4.2 TwinCAT 2 procedure

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

i System requirements

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

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

- in the PLC, see [step 1a](#) | [121](#)

- or directly in the System Manager if only one free task is present, see [step 1b](#) [▶ 121]

The same settings are to be made in the Scope2 for both cases, see [step 2](#) [▶ 123]

Recording of a PLC variable with the TwinCAT 2 – Scope2

Step 1a: TwinCAT 2 PLC

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

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

Fig. 153: PLC declaration

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

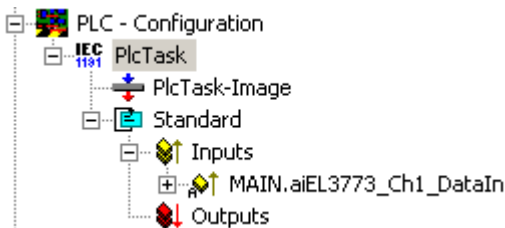


Fig. 154: PLC in the System Manager

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

Step 1b: TwinCAT 2 - free task

So that the linking works, an array variable with the channel data must be present in the system manager; i.e. each oversampling data package must be present in an array. This array variable must be defined and created manually in the System Manager.

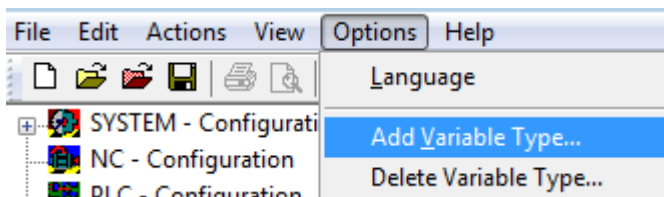


Fig. 155: Add Variable Type

An ARRAY variable of the type as known by the PLC must be created in the syntax as known from the PLC. In this example an array of 0..9 of type INT, i.e. with 10 fields.

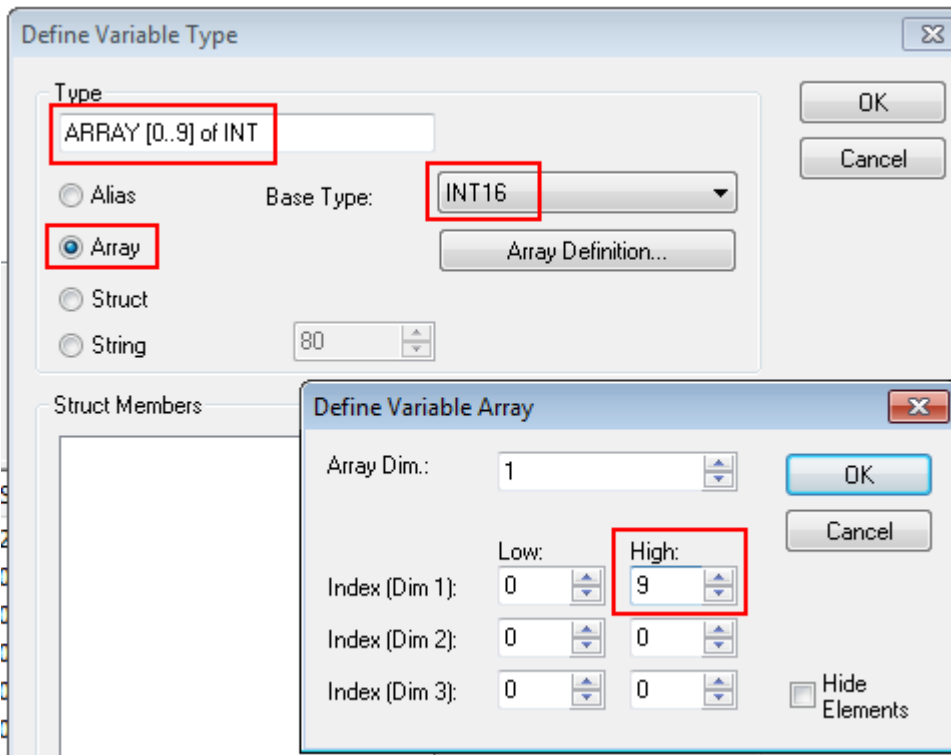


Fig. 156: Definition of the variable type

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

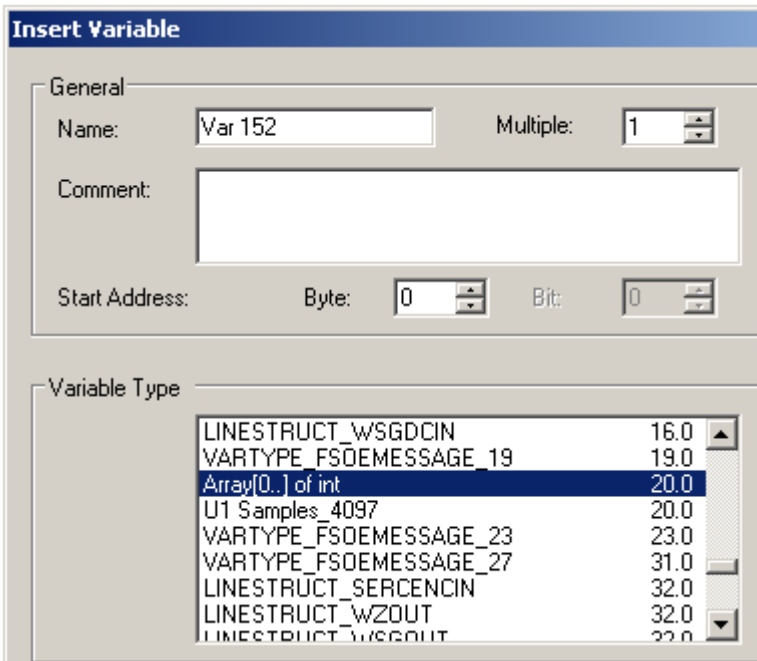


Fig. 157: Overview of declared types

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

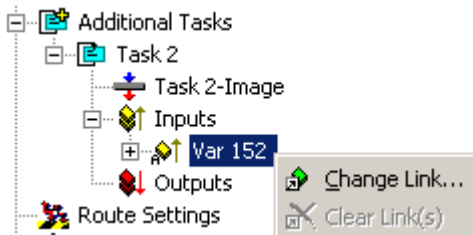


Fig. 158: Linking

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

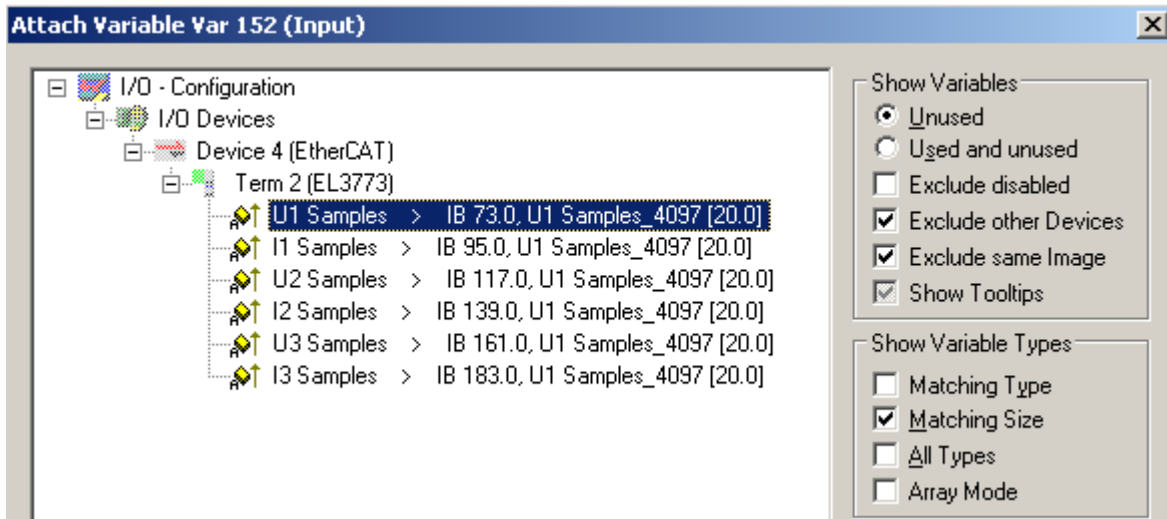


Fig. 159: Array variables of an oversampling terminal

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

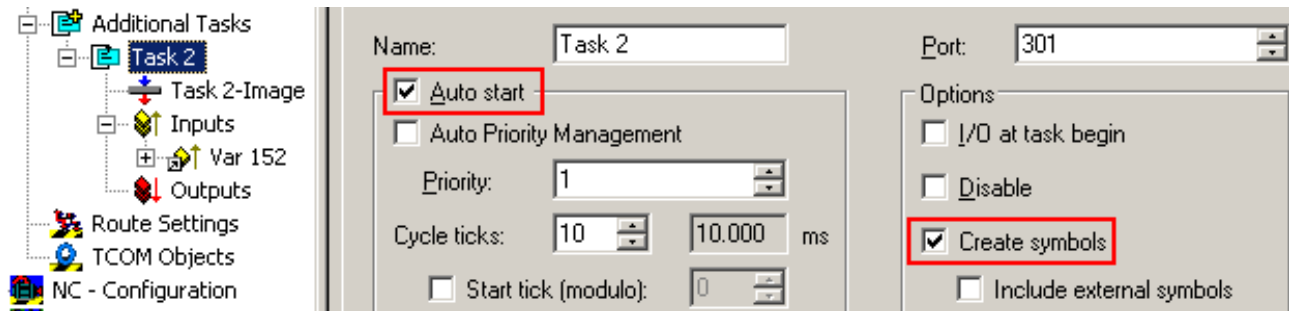


Fig. 160: Settings in the additional task

Step 2: Configuration in the Scope2

So that the linking works, an array variable with the channel data of the respective terminal/box must be present in the system manager; i.e. each oversampling data package must be present in an array. This array variable must be defined and created manually; [see above \[▶ 121\]](#).

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

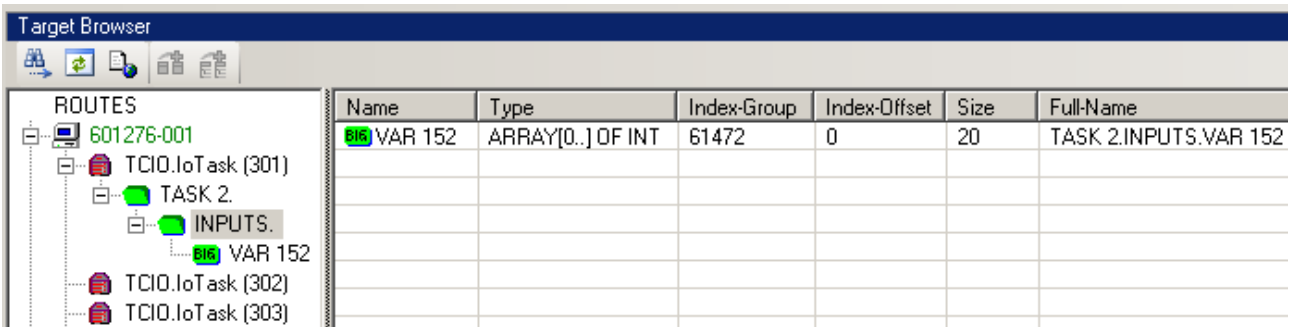


Fig. 161: Variable browser up to the array VAR152

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

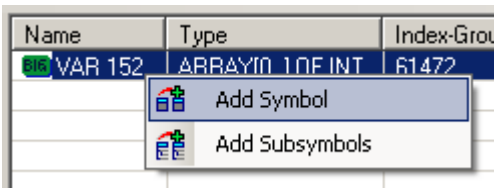


Fig. 162: AddSymbol on the array

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

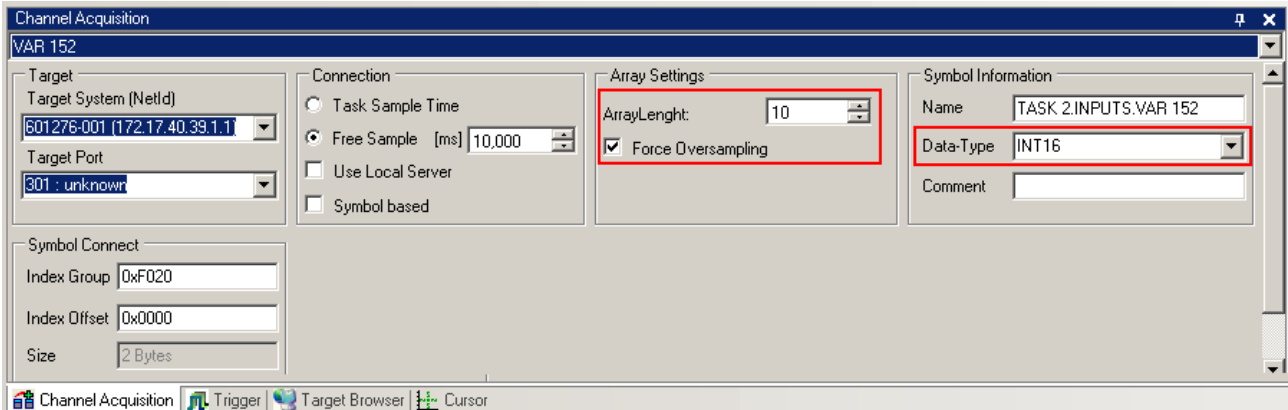


Fig. 163: Channel settings

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

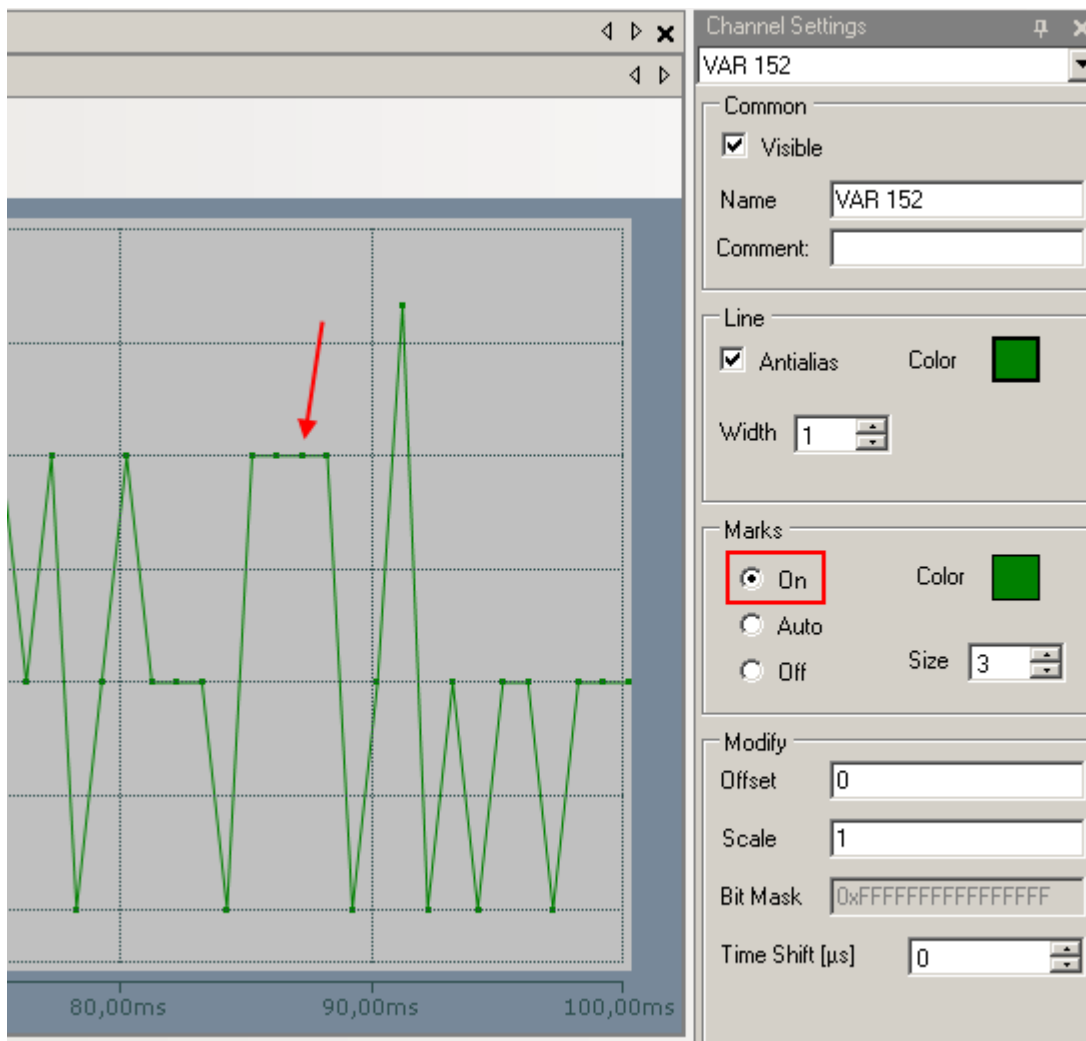


Fig. 164: Activation of the marks

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

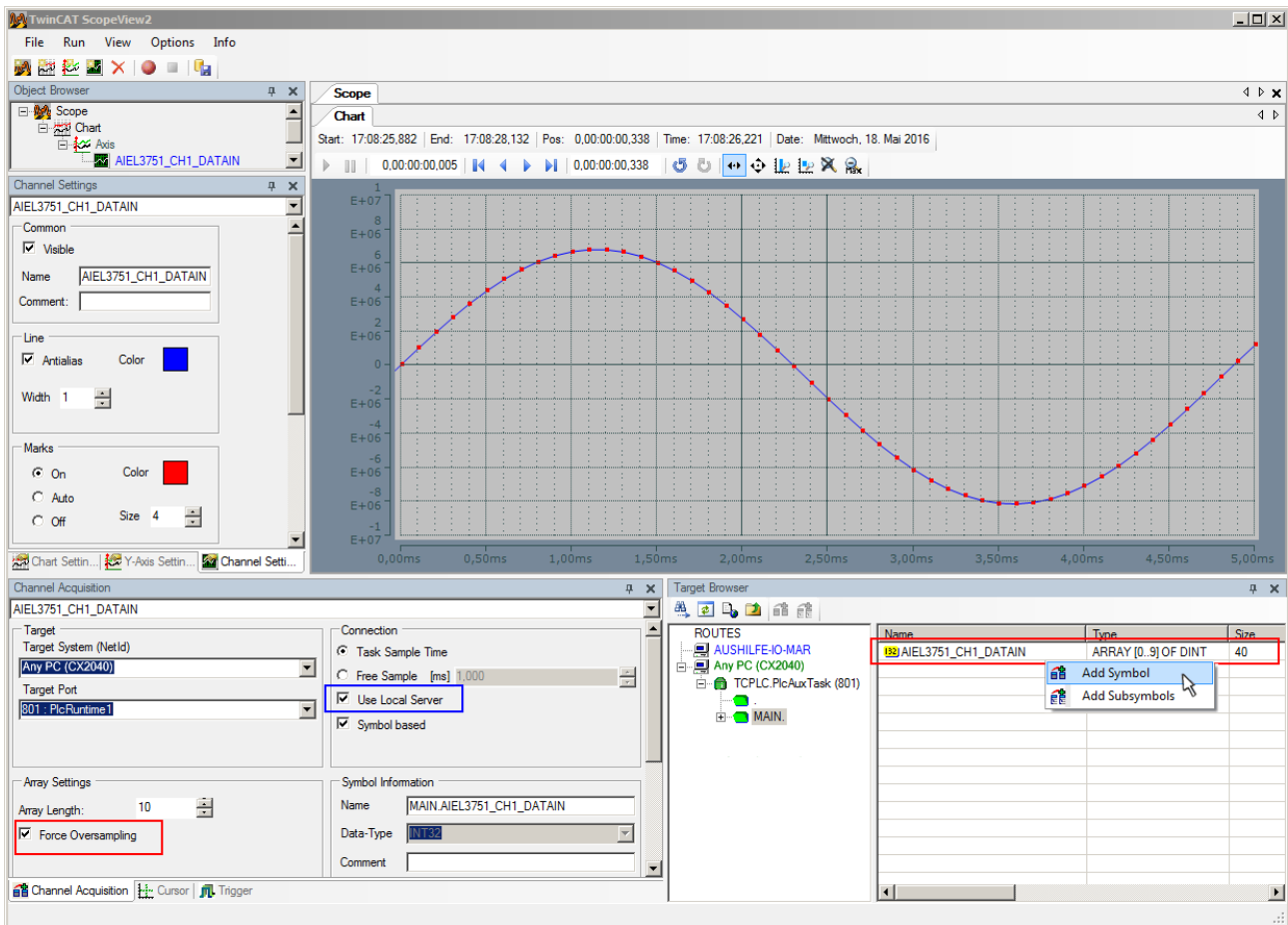


Fig. 165: Illustration of a 10 x oversampling variable of the EL3751 by the Scope2

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

Proceeding with TwinCAT 2/ alternatively via ADS

In former TwinCAT 2 versions (or a lower revision as specified in the [table \[▶ 112\]](#) above) the oversampling PDO of the respective oversampling able terminal/box can be made visible for the Scope2 by activation of the ADS server.

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

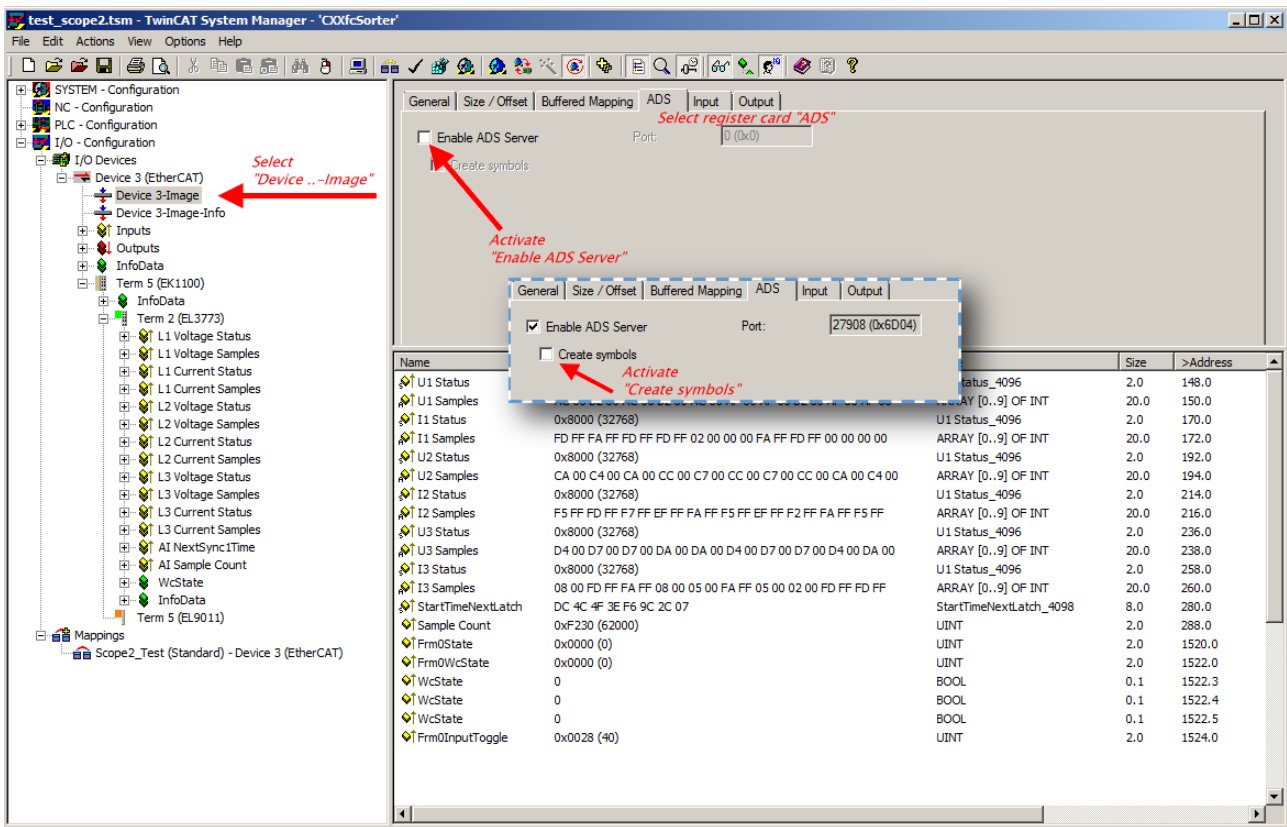


Fig. 166: Activation of the ADS server of the EtherCAT Device (TwinCAT 2)

The activation of the ADS server have to be carried out by selection of the “Device – Image” on the left sided configuration tree:

„I/O – Configuration → I/O Devices → Device .. (EtherCAT) → Device .. – Image“.

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

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

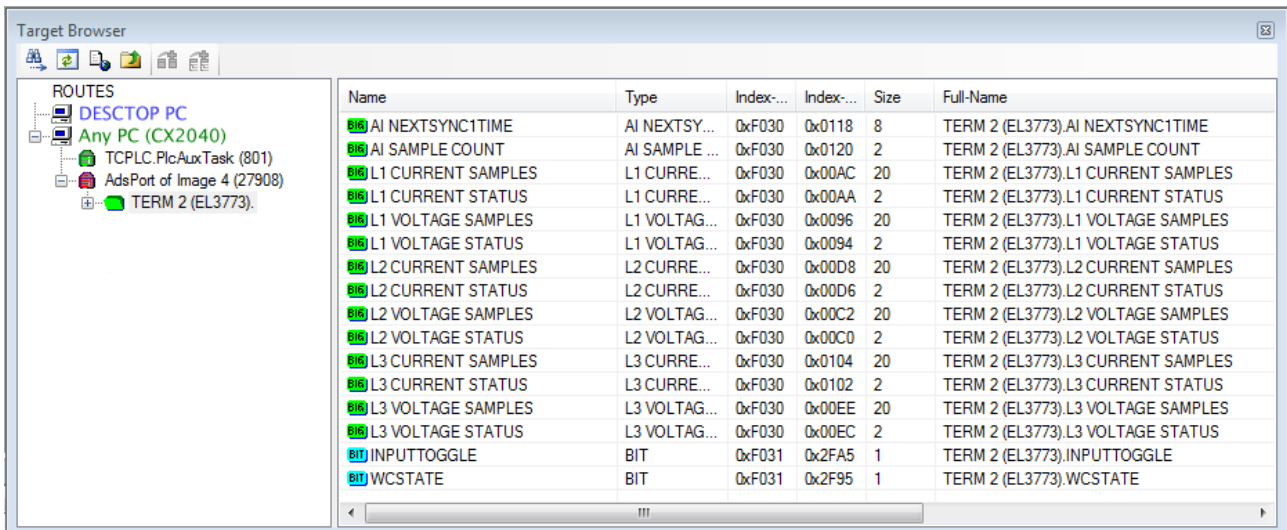
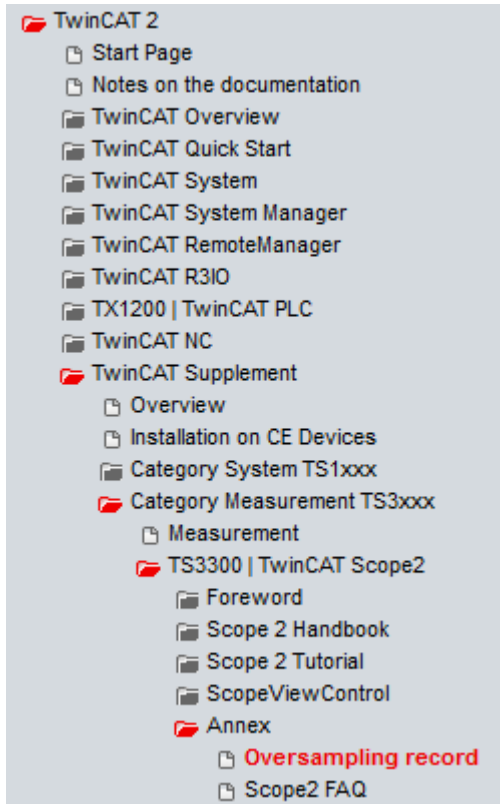


Fig. 167: Direct access of the Scope2 to the terminal's PDOs

Also see Beckhoff Information System

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



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

Name	Type	Index...	Index...	Size	Full-Name
CH1 SAMPLE 0[0]	CH1 SAMPLE_0_TYPE	61488	73	2	TERM 2 (EL3702).CH1 SAMPLE 0[0]
CH1 SAMPLE 0[1]	CH1 SAMPLE_0_TYPE	61488	75	2	TERM 2 (EL3702).CH1 SAMPLE 0[1]
CH1 SAMPLE 0[2]	CH1 SAMPLE_0_TYPE	61488	77	2	TERM 2 (EL3702).CH1 SAMPLE 0[2]
CH1 SAMPLE 0[3]	CH1 SAMPLE_0_TYPE	61488	79	2	TERM 2 (EL3702).CH1 SAMPLE 0[3]
CH1 SAMPLE 0[4]	CH1 SAMPLE_0_TYPE	61488	81	2	TERM 2 (EL3702).CH1 SAMPLE 0[4]
CH1 SAMPLE 0[5]	CH1 SAMPLE_0_TYPE	61488	83	2	TERM 2 (EL3702).CH1 SAMPLE 0[5]
CH1 SAMPLE 0[6]	CH1 SAMPLE_0_TYPE	61488	85	2	TERM 2 (EL3702).CH1 SAMPLE 0[6]
CH1 SAMPLE 0[7]	CH1 SAMPLE_0_TYPE	61488	87	2	TERM 2 (EL3702).CH1 SAMPLE 0[7]
CH1 SAMPLE 0[8]	CH1 SAMPLE_0_TYPE	61488	89	2	TERM 2 (EL3702).CH1 SAMPLE 0[8]
CH1 SAMPLE 0[9]	CH1 SAMPLE_0_TYPE	61488	91	2	TERM 2 (EL3702).CH1 SAMPLE 0[9]
CH1 SAMPLE 0[T10]	CH1 SAMPLE_0_TYPE	61488	73	2	TERM 2 (EL3702).CH1 SAMPLE 0[T10]

Name	Type	Index...	Index...	Size	Full-Name
CH1 VALUE	INT16	61488	73	2	TERM 2 (EL3702).CH1 SAMPLE 0[T10].CH1 VALUE

Fig. 168: Automatically calculated array variable (red) in the Scope2

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

5.5 Process data

Data flow

The amplitude values are measured with 16-bit resolution and provided as oversampling packets in the process image for cyclic transmission.

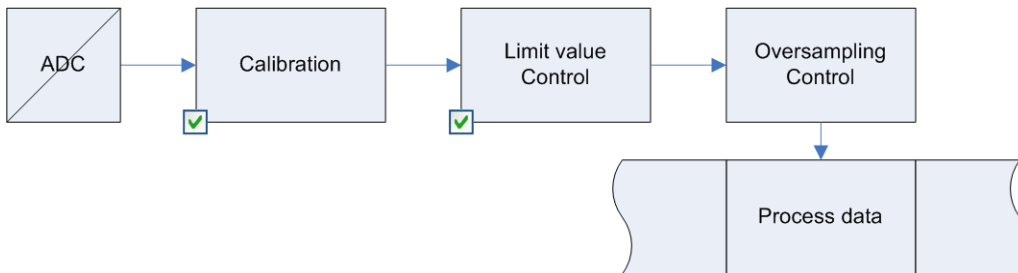


Fig. 169: Process data flow EL3783

Data processing

The data processing is done simultaneously for all channels in the EL3783 with 16-bit conversion in the ADC.

Scaling related to the respective (constant) measuring range:

Output value presentation

Voltage

MBE = 737,2575 V (~= 130% * 400V * Wurzel(2)) --> **22,5 mV / digit**

Current

5 A range: MBE = 9,207527 (~= 130% * 5A * root(2)) --> **281µA / digit**

1 A range: MBE = 1,8415054 (~= 130% * 1A * root(2)) --> **56,2µA / digit**

1. Vendor calibration (0x80p0:0B [▶ 139])

Activation: CoE 0x80p0:0B

$$X_1 = (X_{ADC} - \text{Offset}_{Vendor}) * \text{Gain}_{Vendor}$$

2. User calibration (0x80p0:0A [▶ 139])

Activation: CoE 0x80n0:0A

$$X_2 = (X_1 - \text{Offset}_{User}) * \text{Gain}_{User}$$

3. Limit value evaluation (0x80p0:13 [▶ 139], 0x80p0:14 [▶ 139])

Display in the status word [▶ 135] of the channel

If the value exceeds or falls below these values, which can be entered in the indices 0x80p0:13 [▶ 139] and 0x80p0:14 [▶ 139], then the bits in the indices 0x60p0:03 [▶ 140] and 0x60p0:05 [▶ 140] are set accordingly (see example below).

The entry 0x80p0:07 [▶ 139] serves to activate the limit value monitoring.

Output limit n (2-bit):

- 0: not active

- 1: One or more values <= Limit n
- 2: One or more values >= Limit n
- 3: Cases 1 and 2 simultaneously

i Linking in the PLC with 2-bit values

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

- PLC:
IEC61131-PLC contains no 2-bit data type that can be linked with this process data directly. In order to transmit the limit information, therefore, define an input byte, e.g.

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

and link the limit with the *VariableSizeMismatch* dialog.

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

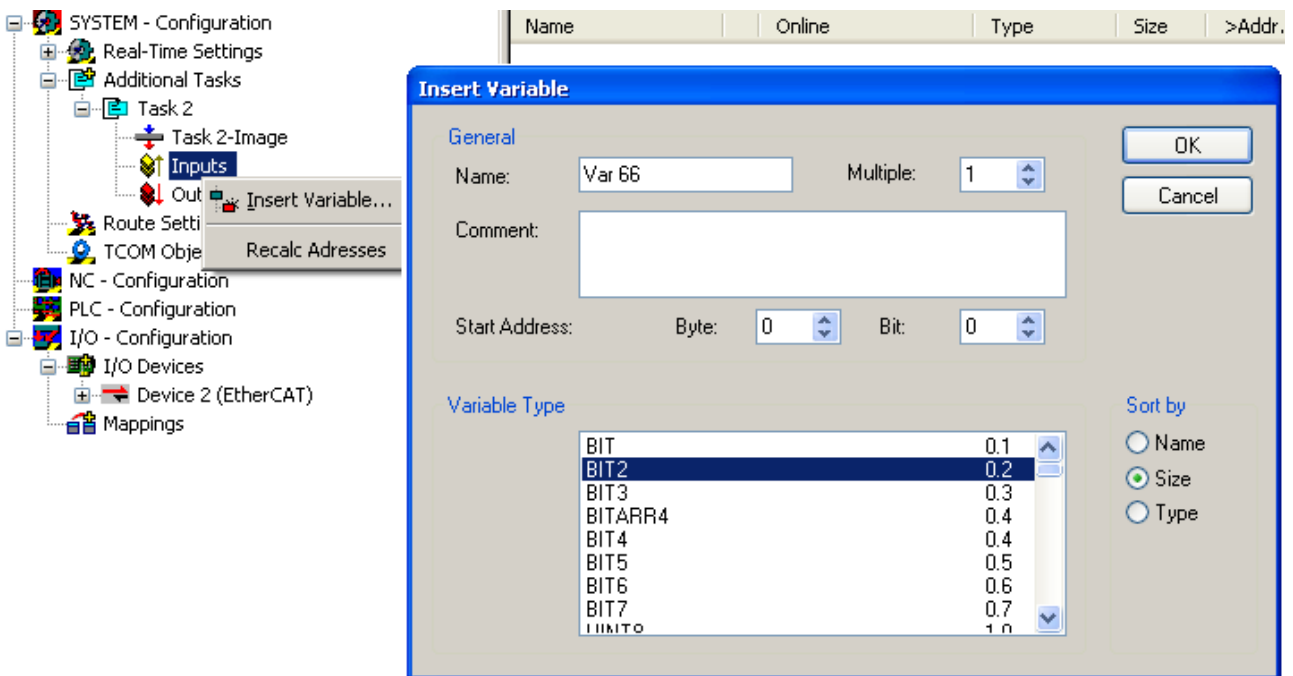


Fig. 170: Linking of 2-bit variable to additional task

Example of voltage measurement with EL3783:

Channel 1; Limit 1 and Limit 2 enabled, Limit 1 = 100 V, Limit 2 = 200 V, representation: signed integer

Entry in index (Limit 1): 0x8000:13 [▶ 139]
 $100 \text{ V} / (0,0225 \text{ V} / \text{digit}) = 4444_{\text{dec}}$

Entry in index (Limit 2): 0x8000:14 [▶ 139]
 $200 \text{ V} / (0,0225 \text{ V} / \text{digit}) = 8889_{\text{dec}}$

Output:

Input channel 1	Index 0x6000:03 [▶ 140]	Index 0x6000:05 [▶ 140]
50 V	0x01 _{hex} , (Limit 1, limit value undershot)	0x01 _{hex} , (Limit 2, limit value undershot)
100 V	0x03 _{hex} , (Limit 1, limit value reached)	0x01 _{hex} , (Limit 2, limit value undershot)
150 V	0x02 _{hex} , (Limit 1, limit value exceeded)	0x01 _{hex} , (Limit 2, limit value undershot)
210 V	0x02 _{hex} , (Limit 1, limit value exceeded)	0x02 _{hex} , (Limit 2, limit value exceeded)

4. Provision in the process data

The influencing parameters can be changed in the CoE of the respective channel.

Index	Parameter Name	Access	Value
8000:0	AI Settings	RW	> 24 <
8000:07	Enable limit	RW	FALSE
8000:0A	Enable user calibration	RW	FALSE
8000:0B	Enable vendor calibration	RW	TRUE
8000:13	Limit 1	RW	29490
8000:14	Limit 2	RW	-29490
8000:17	User calibration offset	RW	0
8000:18	User calibration gain	RW	16384

Fig. 171: User settings and vendor calibration in the CoE online directory

● Changes in the CoE directory

i In case of changes to the CoE default parameters, it is essential that corresponding values are entered in the start-up list, so that in the case of exchange the EL3783 operates again as foreseen in the application.

Predefined PDO Assignment

The "Predefined PDO Assignment" enables a simplified selection of the process data. The desired function is selected on the lower part of the "Process Data" tab. As a result, all necessary PDOs are automatically activated and the unnecessary PDOs are deactivated. 8 PDO assignments are available:

Name	SM3, PDO assignment, meaning
Synchronous	0x1A00, 0x1A01, 0x1A10, 0x1A11, 0x1A20, 0x1A21, 0x1A30, 0x1A31, 0x1A40, 0x1A41, 0x1A50, 0x1A51 "Synchronous"
Synchronous oversampling 2	0x1A00, 0x1A02, 0x1A10, 0x1A12, 0x1A20, 0x1A22, 0x1A30, 0x1A32, 0x1A40, 0x1A42, 0x1A50, 0x1A52 0x1A61 "Synchronous, with 2-fold oversampling"
Synchronous oversampling 4	0x1A00, 0x1A03, 0x1A10, 0x1A13, 0x1A20, 0x1A23, 0x1A30, 0x1A33, 0x1A40, 0x1A43, 0x1A50, 0x1A53 0x1A61 "Synchronous, with 4-fold oversampling"
Synchronous oversampling 8	0x1A00, 0x1A05, 0x1A10, 0x1A15, 0x1A20, 0x1A25, 0x1A30, 0x1A35, 0x1A40, 0x1A45, 0x1A50, 0x1A55 0x1A61 "Synchronous, with 8-fold oversampling"
Synchronous oversampling 16	0x1A00, 0x1A07, 0x1A10, 0x1A17, 0x1A20, 0x1A27, 0x1A30, 0x1A37, 0x1A40, 0x1A47, 0x1A50, 0x1A57 0x1A61 "Synchronous, with 16-fold oversampling"
Synchronous oversampling 32	0x1A00, 0x1A0A, 0x1A10, 0x1A1A, 0x1A20, 0x1A2A, 0x1A30, 0x1A3A, 0x1A40, 0x1A4A, 0x1A50, 0x1A5A 0x1A61 "Synchronous, with 32-fold oversampling"
Synchronous oversampling 64	0x1A00, 0x1A0D, 0x1A10, 0x1A1D, 0x1A20, 0x1A2D, 0x1A30, 0x1A3D, 0x1A40, 0x1A4D, 0x1A50, 0x1A5D 0x1A61 "Synchronous, with 64-fold oversampling"
DC (activate Mode on DC tab)	0x1A00, 0x1A05, 0x1A10, 0x1A15, 0x1A20, 0x1A25, 0x1A30, 0x1A35, 0x1A40, 0x1A45, 0x1A50, 0x1A55 0x1A60, 0x1A61 "Distributed Clocks, settings via DC tab [▶ 136]"

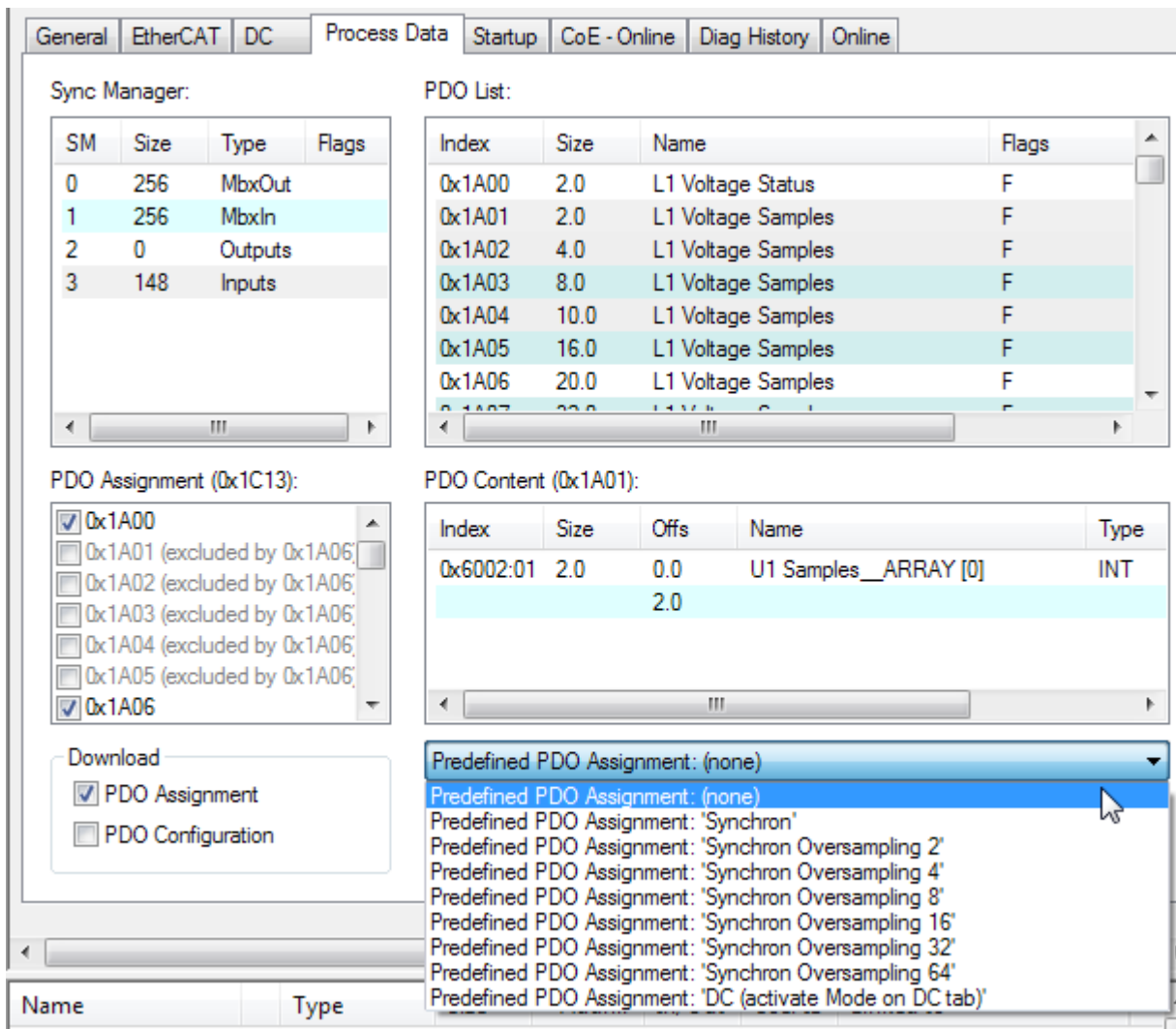


Fig. 172: EL3783 Selection dialog "Predefined PDO Assignment"

Process image

The EL3783 is inserted into the configuration by default with the 10-fold oversampling process image and DC timestamp:

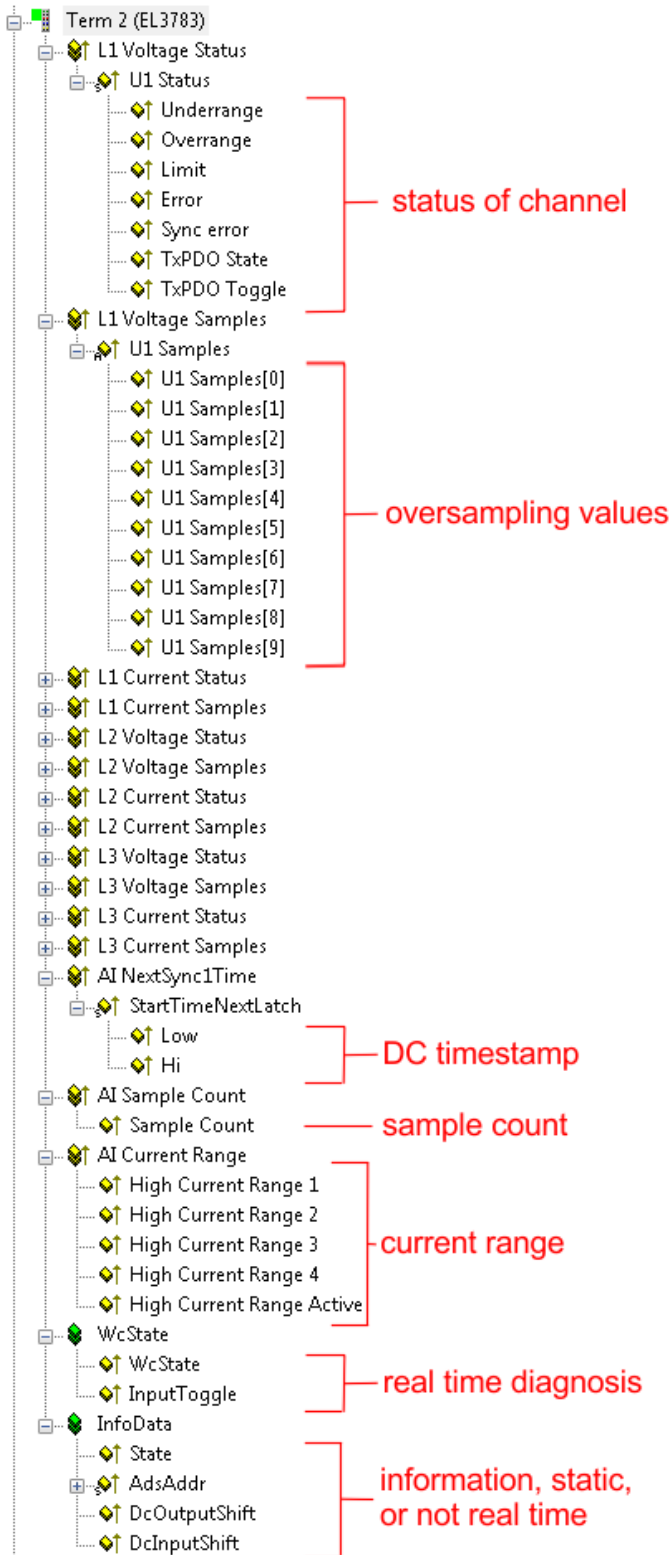


Fig. 173: Process image of the EL3783 in the TwinCAT System Manager

In particular the EL3783 offers the following cyclic information:

Variable	Meaning
Status word	See below [► 135]
Overrange	Value after calibration > 0x7FFF
Underrange	Value after calibration < 0x8000
SyncError	- In DC mode: indicates whether a synchronization error occurred in the expired cycle. This means a SYNC signal was triggered in the terminal, although no new process data were available (0=OK, 1=NOK). - Error in the synchronous oversampling, e.g. the number of ADC values and the number of PDO values do not match.
StartTimeNextLatch (LO/HI)	In the 64-bit-wide process data <i>StartTimeNextLatch</i> , as was also the case with previous EL37xx terminals, the time is specified in each process data cycle when the <i>next</i> SYNC1 pulse and thus the next block of sample values begins, referenced to the currently transmitted block. <i>StartTimeNextLatch</i> thus changes in each cycle by the amount of that task cycle time with which this terminal is operated. This time specification is based on the terminal's local Distributed Clocks time. By means of this time specification a concrete time can to be assigned to each individual sample with the known oversampling factor. Example: With a cycle time of 1 ms (= 1,000,000 ns) and an oversampling factor of 20 in the regarded cycle, the EL3783 supplies a $StartTimeNextLatch = 7,777,216_{dec}$ and 2 x 20 measured values at 16 bits each as process data. The time of measurement of the 5 th supplied sample is now to be determined, i.e. the Distributed Clocks time at which the 5 th sample was determined. The currently supplied set of 20 samples was started at the time $7,777,216 - 1,000,000$ (cycle time) = 6,777,216 ns. The time interval between the samples is $1,000,000/20 = 50,000$ ns. Hence, the 5 th sample was determined at the time $6,777,216 + ((5 - 1) * 50,000) = 6,977,216$ ns.
Sample Counter (Cycle count)	The Sample Counter is incremented by one unit with each process data cycle. The Sample Counter enables the higher-level controller to check whether a data record has possibly been omitted or transmitted twice. In that case the DC shift time of the terminal usually has to be adapted.
Current range	High Current Range 1-4 geben Auskunft über den Strombereich jedes einzelnen Samples. High Current Range Active beschreibt zyklusweise, ob der 5A Bereich eingeschaltet war oder nicht.
DcOutputShift, DcInputShift	In these static variables the System Manager announces the shift time to which this terminal has been set. The value is set once on activating/calculating the configuration and also depends on the customer-specific settings in the extended slave settings. It can be linked to offset calculations in the PLC.

i Evaluation of the diagnostic values

It is urgently recommended to evaluate the offered diagnostic values in the controller, e.g. see the [notes page](#) [► 99].
The evaluation of the Sample Counter (see above) e.g. provides a simple detection of communication errors and allow users to react accordingly in the software application.

Control/status word

Status word

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

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

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

Table 1: Legend

Bit	Name	Description
SW.15	TxPDO Toggle	1 _{bin} Toggles with each new analog process value
SW.14	TxPDO State	1 _{bin} TRUE in the case of an internal error
SW.13*	Sync error	1 _{bin} TRUE (DC mode): a synchronization error occurred in the expired cycle.
SW.6	ERROR	1 _{bin} General error bit, is also set together with overrange and underrange
SW.3	Limit 1	1 _{bin} See Limit [▶ 140]
SW.2		1 _{bin}
SW.1	Overrange	1 _{bin} Analog input signal lies above the upper permissible threshold for this terminal
SW.0	Underrange	1 _{bin} Analog input signal lies under the lower permissible threshold for this terminal

Control word

The EL3783 has no control word

Synchronization, trigger and conversion time

The EL3783 generally operates in oversampling mode. It can be operated with and without Distributed Clocks activated.

Distributed Clocks activated

- the EtherCAT system-wide synchronized Distributed Clocks also encompass this EL3783
- the terminal then operates synchronously with all other DC devices
- the sampling clock is derived from the local DC in the EtherCAT slave controller (SYNC0: Oversampling, SYNC1: Provision of the data)

Advantages

- several EL3783 record the measured values of all their channels synchronously
- accurate time stamp of the individual samples

Cycle times

The minimum EtherCAT cycle time for the EL3783 is 50 µs (see table in chapter "Quickstart"). In order to activate Distributed Clocks, "DC" is to be selected in the Predefined PDO list, so that amongst others the time stamp PDO is selected

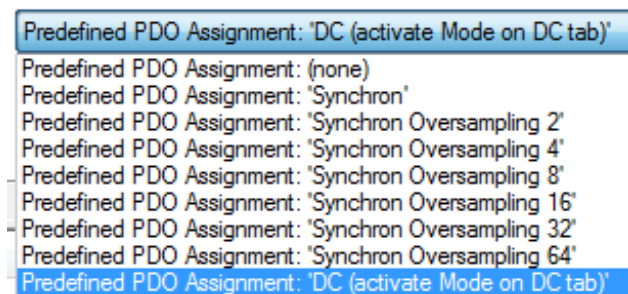


Fig. 174: EL3783 Selection dialog "Predefined PDO Assignment"

The permissible oversampling rates are set via the System Manager *DC* tab:

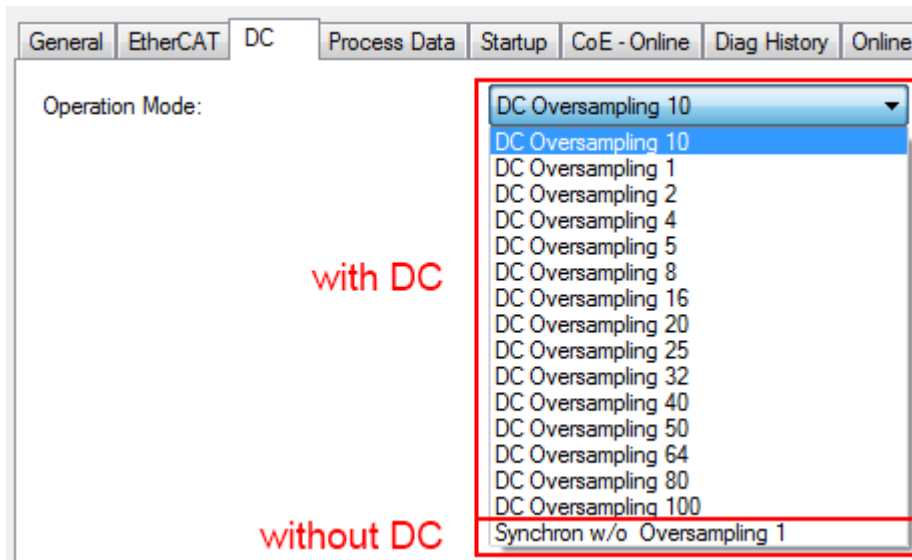


Fig. 175: Setting of the oversampling rates via System Manager, "DC" tab

Distributed Clocks not activated

- the EL3783 operates according to an internal clock, which is synchronized to the EtherCAT cycle time. The trigger point in this case is the access to the input Sync Manager SM3.
- As a result the EL3783 can compensate fluctuations in the cycle time to a large extent
- The EL3783 can thus be used in systems that have no Distributed Clocks functionality. The time accuracy of the measured values is reduced, however.

The EtherCAT cycle time must be selected such that

- it is not less than 200 µs (see table in chapter "Quickstart")
- the oversampling interval does not produce a non-periodic time value – for example, 33.3 µs is not allowed for the oversampling cycle

The Distributed Clock is to be deactivated in the DC tab

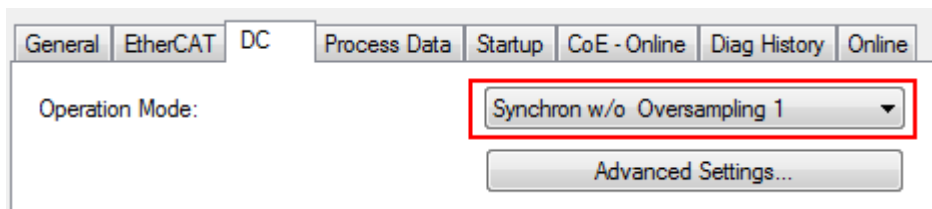


Fig. 176: Deactivation of the Distributed Clock in the "DC" tab

The permissible oversampling rates are then set via the System Manager:

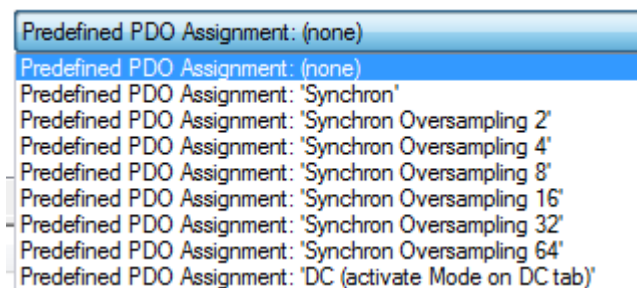


Fig. 177: EL3783 Selection dialog "Predefined PDO Assignment"

5.6 Object description and parameterization

● EtherCAT XML Device Description



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

● Parameterization via the CoE list (CAN over EtherCAT)



The EtherCAT device is parameterized via the CoE-Online tab [► 94] (double-click on the respective object) or via the Process Data tab [► 91] (allocation of PDOs). Please note the following general CoE notes [► 22] when using/manipulating the CoE parameters:

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

Introduction

The CoE overview contains objects for different intended applications:

- Objects required for parameterization during commissioning:
 - Restore object index 0x1011
 - Configuration data index 0x80n0
- Objects intended for regular operation, e.g. through ADS access.
- Profile-specific objects:
 - Configuration data (vendor-specific) index 0x80nF
 - Input data index 0x60n0
 - Information and diagnostic data index 0x80nE, 0xF000, 0xF008, 0xF010
- Standard objects

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

5.6.1 Restore object

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	<u>Restore default parameters [► 181]</u>	Restore default parameters	UINT8	RO	0x01 (1 _{dec})
1011:01	SubIndex 001	If this object is set to " 0x64616F6C " in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0 _{dec})

5.6.2 Configuration data

Index 80p0 AI Settings (for p = 0...5, corresponding to channel 1...6)

Index (hex)	Name	Meaning	Data type	Flags	Default
80p0:0	AI Settings	Maximum subindex	UINT8	RO	> 24 <
80p0:07	Enable limit	Limit 1 enabled	BOOLEAN	RW	FALSE
80p0:0A	Enable user calibration	Enabling of the user calibration	BOOLEAN	RW	FALSE
80p0:0B	Enable vendor calibration	Enabling of the vendor calibration	BOOLEAN	RW	TRUE
80p0:13	Limit 1	First limit value for setting the status bits	INT16	RW	0 _{dec}
80p0:14	Limit 2	Second limit value for setting the status bits	INT16	RW	0 _{dec}
80p0:17	User calibration offset	User calibration offset	INT16	RW	0 _{dec}
80p0:18	User calibration gain	User calibration gain	INT16	RW	16384 _{dec}

Index F80D AI Device Settings

Index (hex)	Name	Meaning	Data type	Flags	Default
F80D:0	AI Device Settings	Maximum subindex	UINT8	RO	> 18 <
F80D:11	Current range	100: Measuring range 1 A – no automatic switchover 500: Measuring range 5 A no automatic switchover 600: Dynamic 0 – fast switching 601: Dynamic 1 602: Dynamic 2 603: Dynamic 3 – inert switching	INT16	RW	1 A (100)
F80D:12	DCTimeStampShift	Distributed Clocks Time Stamp Shift	INT16	RW	0

Index 80pF AI Vendor data (for p = 0...5, corresponding to channel 1...6)

Index (hex)	Name	Meaning	Data type	Flags	Default
80pF:0	AI Vendor data	Maximum subindex	UINT8	RO	> 9 <
80pF:01	Calibration offset	Offset (vendor calibration) voltage or 1 A range	INT16	RW	0
80pF:02	Calibration gain	Gain (vendor calibration) voltage or 1 A range	INT16	RW	16384
80pF:03	Calibration phase offset	reserved	INT16	RW	0
80pF:04	Calibration offset range 1	reserved	INT16	RW	0
80pF:05	Calibration gain range 1	reserved	INT16	RW	16384
80pF:06	Calibration phase offset range 1	reserved	INT16	RW	0
80pF:07	Calibration offset range 2	Offset Range 2 (Herstellerabgleich)	INT16	RW	0
80pF:08	Calibration gain range 2	Gain Range 2 (Herstellerabgleich)	INT16	RW	16384
80pF:09	Calibration phase offset range 2	reserved	INT16	RW	0

5.6.3 Input data

Index 60p0 AI Inputs (for p = 0...5, corresponding to channel 1...6)

Index (hex)	Name	Meaning	Data type	Flags	Default
60p0:0	Status	Maximum subindex	UINT8	RO	> 16 <
60p0:01	Underrange	Value below measuring range.	BOOLEAN	RO	FALSE
60p0:02	Overrange	Measuring range exceeded.	BOOLEAN	RO	FALSE
60p0:03	Limit	Limit value monitoring Limit 0: not active 1: One or more values <= Limit 2: One or more values >= Limit 3: Cases 1 and 2 simultaneously	BIT2	RO	0x00 (0 _{dec})
60p0:07	Error	The error bit is set if the data is invalid (over-range, under-range)	BOOLEAN	RO	FALSE
60p0:0E	Sync error	The Sync error bit is only required for DC mode. It indicates whether a synchronization error has occurred during the previous cycle. This means a SYNC signal was triggered in the terminal, although no new process data were available (0=OK, 1=NOK).	BOOLEAN	RO	FALSE
60p0:0F	TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).	BOOLEAN	RO	FALSE
60p0:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	FALSE

Index 60p2 Samples (for p = 0...5, corresponding to channel 1...6)

Index (hex)	Name	Meaning	Data type	Flags	Default
60p2:0	Samples	Maximum subindex	UINT8	RO	> 100 <
60p2:01	Subindex 001	Sample 001	BOOLEAN	RO	0
...
60p2:64	Subindex 100	Sample 100	BOOLEAN	RO	0

5.6.4 Information and diagnostic data

Index 90p0 Analog Input Info data Ch.1 - 6 (for p = 0..5, corresponds to Ch.1..6)

Index (hex)	Name	Meaning	Data type	Flags	Default
90p0:0	Analog Input Info data	Maximum subindex	UINT8	RO	> 1 <
90p0:01	Saturation Time	Time, the channel has been run in overload range (unit:1 s)	UINT32	RO	0x00000000 (0 _{dec})

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Maximum subindex	UINT8	RO	> 21 <
10F3:01	Maximum Messages	Maximum number of stored messages A maximum of 50 messages can be stored	UINT8	RO	0x10 (16 _{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	FALSE
10F3:05	Flags	not used	UINT16	RW	0x0000 (0 _{dec})
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[28]	RO	see Diag Messages
...
10F3:15	Diagnosis Message 016	Message 16	OCTET-STRING[28]	RO	-

Index 10F8 Actual Time Stamp

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

Index F600 NextSync1 Time

Index (hex)	Name	Meaning	Data type	Flags	Default
6060:0	Next Sync1 Time	Max. subindex	UINT8	RO	> 1 <
6060:01	Lo	Time Stamp (Lo)	UINT64	RO	0x00000000 (0)
6060:01	Hi	Time Stamp (Hi)	UINT64	RO	0x00000000 (0)

Index F601 Sample Count

Index (hex)	Name	Meaning	Data type	Flags	Default
6061:0	Sample Count	Max. Subindex	UINT8	RO	> 1 <
6061:01	Sample Count	see Process data ▶ 133	UINT16	RO	0x0000 (0 _{dec})

Index F602 Synchron Oversampling Diagnosis

Index (hex)	Name	Meaning	Data type	Flags	Default
6062:0	Synchron Oversampling Diagnosis	Max. subindex	UINT8	RO	> 4 <
6062:02	Measured Cycle Time	Measured cycle time	DINT32	RO	-
6062:03	Internal Cycle Time	Internal cycle time	DINT32	RO	-
6062:04	Internal Buffer	Internal Buffer	INT16	RO	-

Index F603 Current Range

Index (hex)	Name	Meaning	Data type	Flags	Default
6063:0	Sample Count	Max. subindex	UINT8	RO	> 5 <
6063:01	High Current Range 1	Current range of the single sample (1)	UINT16	RO	0x00 (0)
6063:02	High Current Range 2	Current range of the single sample (2)	UINT16	RO	0x00 (0)
6063:03	High Current Range 3	Current range of the single sample (3)	UINT16	RO	0x00 (0)
6063:04	High Current Range 4	Current range of the single sample (4)	UINT16	RO	0x00 (0)
6063:05	High Current Range Active	TRUE: 5 A range active	UINT16	RO	FALSE

5.6.5 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	0x00001389 (5001 _{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL3783-0000

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	-

Index 100B Bootloader version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Bootloader version	Version of Bootloader	STRING	RO	-

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	> 4 <
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x0EBD3052 (247279698 _{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	-
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	-

Index 10F0 Backup parameter handling

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

Index 18pp L1/2/3 Voltage/Current TxPDO-Par Samples (for Ch.1 ... Ch.6; 01_{hex} ≤ pp ≤ 5F_{hex}, pp ≠ 10, 20, 30, 40, 50)

Index (hex)	Name	Meaning	Data type	Flags	Default
18pp:0	AI TxPDO-Par Sample	PDO Parameter TxPDO	UINT8	RO	> 6 <
18pp:06	Exclude TxPDOs	Specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with this PDO	OCTET-STRING[2]	RO	see Predefined PDO Assignment

Index 1Ap0 L1/2/3 Voltage/Current TxPDO-Map Status (for p = 0...5; dependent on the number of channels)

Index (hex)	Name	Meaning	Data type	Flags	Default
1Ap0:0	AI TxPDO-Map Status	PDO Mapping TxPDO	UINT8	RO	> 10 <
1Ap0:01	SubIndex 001	1. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x01 (Underrange))	UINT32	RO	0x60p0:01, 1
1Ap0:02	SubIndex 002	2. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x02 (Overrange))	UINT32	RO	0x60p0:02, 1
1Ap0:03	SubIndex 003	3. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x03 (Limit))	UINT32	RO	0x60p0:03, 2
1Ap0:04	SubIndex 004	4. PDO Mapping entry (2 bit align)	UINT32	RO	0x0000:00, 2
1Ap0:05	SubIndex 005	5. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x07 (Error))	UINT32	RO	0x60p0:07, 1
1Ap0:06	SubIndex 006	6. PDO Mapping entry (1 bit align)	UINT32	RO	0x0000:00, 1
1Ap0:07	SubIndex 007	7. PDO Mapping entry (5 bits align)	UINT32	RO	0x0000:00, 5
1Ap0:08	SubIndex 008	8. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x0E (Sync error))	UINT32	RO	0x60p0:0E, 1
1Ap0:09	SubIndex 009	9. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x0F (TxPDO State))	UINT32	RO	0x60p0:0F, 1
1Ap0:0A	SubIndex 010	10. PDO Mapping entry (object 0x60p0 (AI Inputs), entry 0x10 (TxPDO Toggle))	UINT32	RO	0x60p0:10, 1

Index 1Apq L1/2/3 Voltage/Current TxPDO-Map Samples

Index (hex)	Name	Meaning	Data type	Flags	Default																																																																								
1Apq:0	AI TxPDO-Map Samples	<table border="1"> <thead> <tr> <th>n</th> <th>p</th> <th>q</th> </tr> </thead> <tbody> <tr> <td>1_{dec} = Max. Subindex: 001</td> <td>0x0 = L1 Voltage Samples</td> <td>0x1 = 1 Sample</td> </tr> <tr> <td>2_{dec} = Max. Subindex: 002</td> <td>0x1 = L1 Current Samples</td> <td>0x2 = 2 Samples,</td> </tr> <tr> <td>4_{dec} = Max. Subindex: 004</td> <td>0x2 = L2 Voltage Samples</td> <td>0x60p2:01, 16</td> </tr> <tr> <td>5_{dec} = Max. Subindex: 005</td> <td>0x3 = L2 Current Samples</td> <td>0x60p2:02, 16</td> </tr> <tr> <td>8_{dec} = Max. Subindex: 008</td> <td>0x4 = L3 Voltage Samples</td> <td>0x3 = 4 Samples,</td> </tr> <tr> <td>10_{dec} = Max. Subindex: 010</td> <td>0x5 = L3 Current Samples</td> <td>0x60p2:01, 16</td> </tr> <tr> <td>16_{dec} = Max. Subindex: 016</td> <td>0x6 = L3 Voltage Samples</td> <td>....</td> </tr> <tr> <td>20_{dec} = Max. Subindex: 020</td> <td>0x7 = L3 Current Samples</td> <td>0x60p2:04, 16</td> </tr> <tr> <td>25_{dec} = Max. Subindex: 025</td> <td>0x8 = L3 Voltage Samples</td> <td>0x4 = 5 Samples,....</td> </tr> <tr> <td>32_{dec} = Max. Subindex: 032</td> <td>0x9 = L3 Current Samples</td> <td>0x5 = 8 Samples,....</td> </tr> <tr> <td>40_{dec} = Max. Subindex: 040</td> <td>0xA = L3 Voltage Samples</td> <td>0x6 = 10 Samples,....</td> </tr> <tr> <td>50_{dec} = Max. Subindex: 050</td> <td>0xB = L3 Current Samples</td> <td>0x7 = 16 Samples,....</td> </tr> <tr> <td>64_{dec} = Max. Subindex: 064</td> <td>0xC = L3 Voltage Samples</td> <td>0x8 = 20 Samples,....</td> </tr> <tr> <td>80_{dec} = Max. Subindex: 080</td> <td>0xD = L3 Current Samples</td> <td>0x9 = 25 Samples,....</td> </tr> <tr> <td>100_{dec} = Max. Subindex: 100</td> <td>0xE = L3 Voltage Samples</td> <td>0xA = 32 Samples,....</td> </tr> <tr> <td></td> <td></td> <td>0xB = 40 Samples,....</td> </tr> <tr> <td></td> <td></td> <td>0xC = 50 Samples,....</td> </tr> <tr> <td></td> <td></td> <td>0xD = 64 Samples,....</td> </tr> <tr> <td></td> <td></td> <td>0xE = 80 Samples,....</td> </tr> <tr> <td></td> <td></td> <td>0xF = 100 Samples,</td> </tr> <tr> <td></td> <td></td> <td>0x60p2:01, 16</td> </tr> <tr> <td></td> <td></td> <td>....</td> </tr> <tr> <td></td> <td></td> <td>0x60p2:64, 16</td> </tr> </tbody> </table>	n	p	q	1 _{dec} = Max. Subindex: 001	0x0 = L1 Voltage Samples	0x1 = 1 Sample	2 _{dec} = Max. Subindex: 002	0x1 = L1 Current Samples	0x2 = 2 Samples,	4 _{dec} = Max. Subindex: 004	0x2 = L2 Voltage Samples	0x60p2:01, 16	5 _{dec} = Max. Subindex: 005	0x3 = L2 Current Samples	0x60p2:02, 16	8 _{dec} = Max. Subindex: 008	0x4 = L3 Voltage Samples	0x3 = 4 Samples,	10 _{dec} = Max. Subindex: 010	0x5 = L3 Current Samples	0x60p2:01, 16	16 _{dec} = Max. Subindex: 016	0x6 = L3 Voltage Samples	20 _{dec} = Max. Subindex: 020	0x7 = L3 Current Samples	0x60p2:04, 16	25 _{dec} = Max. Subindex: 025	0x8 = L3 Voltage Samples	0x4 = 5 Samples,....	32 _{dec} = Max. Subindex: 032	0x9 = L3 Current Samples	0x5 = 8 Samples,....	40 _{dec} = Max. Subindex: 040	0xA = L3 Voltage Samples	0x6 = 10 Samples,....	50 _{dec} = Max. Subindex: 050	0xB = L3 Current Samples	0x7 = 16 Samples,....	64 _{dec} = Max. Subindex: 064	0xC = L3 Voltage Samples	0x8 = 20 Samples,....	80 _{dec} = Max. Subindex: 080	0xD = L3 Current Samples	0x9 = 25 Samples,....	100 _{dec} = Max. Subindex: 100	0xE = L3 Voltage Samples	0xA = 32 Samples,....			0xB = 40 Samples,....			0xC = 50 Samples,....			0xD = 64 Samples,....			0xE = 80 Samples,....			0xF = 100 Samples,			0x60p2:01, 16					0x60p2:64, 16	UINT8	RO	> n <
n	p	q																																																																											
1 _{dec} = Max. Subindex: 001	0x0 = L1 Voltage Samples	0x1 = 1 Sample																																																																											
2 _{dec} = Max. Subindex: 002	0x1 = L1 Current Samples	0x2 = 2 Samples,																																																																											
4 _{dec} = Max. Subindex: 004	0x2 = L2 Voltage Samples	0x60p2:01, 16																																																																											
5 _{dec} = Max. Subindex: 005	0x3 = L2 Current Samples	0x60p2:02, 16																																																																											
8 _{dec} = Max. Subindex: 008	0x4 = L3 Voltage Samples	0x3 = 4 Samples,																																																																											
10 _{dec} = Max. Subindex: 010	0x5 = L3 Current Samples	0x60p2:01, 16																																																																											
16 _{dec} = Max. Subindex: 016	0x6 = L3 Voltage Samples																																																																											
20 _{dec} = Max. Subindex: 020	0x7 = L3 Current Samples	0x60p2:04, 16																																																																											
25 _{dec} = Max. Subindex: 025	0x8 = L3 Voltage Samples	0x4 = 5 Samples,....																																																																											
32 _{dec} = Max. Subindex: 032	0x9 = L3 Current Samples	0x5 = 8 Samples,....																																																																											
40 _{dec} = Max. Subindex: 040	0xA = L3 Voltage Samples	0x6 = 10 Samples,....																																																																											
50 _{dec} = Max. Subindex: 050	0xB = L3 Current Samples	0x7 = 16 Samples,....																																																																											
64 _{dec} = Max. Subindex: 064	0xC = L3 Voltage Samples	0x8 = 20 Samples,....																																																																											
80 _{dec} = Max. Subindex: 080	0xD = L3 Current Samples	0x9 = 25 Samples,....																																																																											
100 _{dec} = Max. Subindex: 100	0xE = L3 Voltage Samples	0xA = 32 Samples,....																																																																											
		0xB = 40 Samples,....																																																																											
		0xC = 50 Samples,....																																																																											
		0xD = 64 Samples,....																																																																											
		0xE = 80 Samples,....																																																																											
		0xF = 100 Samples,																																																																											
		0x60p2:01, 16																																																																											
																																																																												
		0x60p2:64, 16																																																																											
1Apq:hh (for 01 _{hex} ≤ hh ≤ 64 _{hex})	Subindex n	Voltage Samples or Current Samples	INT16	RO	-																																																																								

Index 1A60 AI TxPDO-Map NextSyncTime

Index (hex)	Name	Meaning	Data type	Flags	Default
1A60:0	AI TxPDO-Map NextSyncTime	Max. subindex	UINT8	RO	> 1 <
1A60:01	Subindex 001	1. PDO Mapping entry (object 0x6060 (NextSync1 Time), entry 0x01 (Lo))	UINT32	RO	0x6060:01, 32
1A60:02	Subindex 002	2. PDO Mapping entry (object 0x6060 (NextSync1 Time), entry 0x02 (Hi))	UINT32	RO	0x6060:02, 32

Index 1A61 AI TxPDO-Map Sample Count

Index (hex)	Name	Meaning	Data type	Flags	Default
1A60:0	AI Sample Count	Max. subindex	UINT8	RO	> 1 <
1A61:01	Subindex 001	1. PDO Mapping entry (object 0x6061 (AI Inputs), entry 0x01 (SampleCount))	UINT16	RO	0x6061:01, 16

Index 1A62 AI TxPDO-Map Synchron Oversampling Diagnosis

Index (hex)	Name	Meaning	Data type	Flags	Default
1A62:0	AI TxPDO-Map Synchron Oversampling Diagnosis	Max. subindex	UINT8	RO	> 3 <
1A62:01	Subindex 001	1. PDO Mapping entry (object 0x6062 (Synchron Oversampling Diagnosis), entry 0x02 (Measured Cycle Time))	UINT32	RO	0x6062:02, 32
1A62:02	Subindex 002	2. PDO Mapping entry (object 0x6062 (Synchron Oversampling Diagnosis), entry 0x03 (Internal Cycle Time))	UINT32	RO	0x6062:03, 32
1A62:03	Subindex 003	3. PDO Mapping entry (object 0x6062 (Synchron Oversampling Diagnosis), entry 0x04 (Internal Buffer))	UINT32	RO	0x6062:04, 16

Index 1A63 AI TxPDO-Map Current Range

Index (hex)	Name	Meaning	Data type	Flags	Default
1A63:0	AI TxPDO-Map Current Range	Max. Subindex	UINT8	RO	> 6 <
1A63:01	Subindex 001	1. PDO Mapping entry (object 0x6063 (Current Range), entry 0x01 (High Current Range 1))	UINT32	RO	0x6063:01, 8
1A63:02	Subindex 002	2. PDO Mapping entry (object 0x6063 (Current Range), entry 0x02 (High Current Range 2))	UINT32	RO	0x6063:02, 8
1A63:03	Subindex 003	3. PDO Mapping entry (object 0x6063 (Current Range), entry 0x03 (High Current Range 2))	UINT32	RO	0x6063:03, 8
1A63:04	Subindex 004	4. PDO Mapping entry (object 0x6063 (Current Range), entry 0x04 (High Current Range 4))	UINT32	RO	0x6063:04, 8
1A63:05	Subindex 005	5. PDO Mapping entry (object 0x6063 (Current Range), entry 0x05 (High Current Range Active))	UINT32	RO	0x6063:05, 1
1A63:06	Subindex 006	6. PDO Mapping entry (object 0x6063 (Current Range), entry 0x00 (Align))	UINT32	RO	0x0000:00, 15

Index 1C00 0 Sync manager type

Index (hex)	Name	Meaning	Data type	Flags	Default
1C00:0	Sync manager type	Using the sync managers	UINT8	RO	> 4 <
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	> 0 <

Index 1C13 TxPDO assign

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

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	> 13 <
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	0x1A06 (6662 _{dec})
1C13:03	SubIndex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A10 (6672 _{dec})
1C13:04	SubIndex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A16 (6678 _{dec})
1C13:05	SubIndex 005	5. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A20 (6688 _{dec})
1C13:06	SubIndex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A26 (6694 _{dec})
1C13:07	SubIndex 007	7. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A30 (6704 _{dec})
1C13:08	SubIndex 008	8. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A36 (6710 _{dec})
1C13:09	SubIndex 009	9. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A40 (6720 _{dec})
1C13:0A	SubIndex 010	10. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A46 (6726 _{dec})
1C13:0B	SubIndex 011	11. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A50 (6736 _{dec})
1C13:0C	SubIndex 012	12. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A56 (6742 _{dec})
1C13:0D	SubIndex 013	13. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A61 (6753 _{dec})
..
1C13:13	SubIndex 019	-	-	-	-

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	> 32 <
1C33:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> • Bit 0 = 0: Free Run • Bit 0 = 1: Synchron with SM 2 Event • Bit 0-1 = 11: DC with SYNC1 event • Bit 15 = 1: Fast Mode 	UINT16	RW	0x0002 (2 _{dec})
1C33:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC-Mode: SYNC0/SYNC1 Cycle Time 	UINT32	RW	0x003D0900 (4000000 _{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 = 1: Free Run • Bit 1 = 1: Synchron with SM 2 Event (Outputs available) • Bit 2 = 1: DC-Mode (SYNC0) • Bit 3 = 1: DC-Mode (SYNC1) • Bit 12 = 1: Legacy Synchron • Bit 13 = 1: SM event • Bit 14 = 1: dynamic times (measure by writing 0x1C33:08 [▶ 146]) • Bit 15 = 1: Fast Mode 	UINT16	RO	0x0806 (2054 _{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	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	With this entry the real required process data provision time can be measured. <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 0x1C33:03 [▶ 146], 0x1C33:06 [▶ 146], 0x1C33:09 [▶ 146] are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p>	UINT16	RW	0x0000 (0 _{dec})
1C33:09	Maximum Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 _{dec})
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0 _{dec})
1C33:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0 _{dec})
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	FALSE

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	> 2 <
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	0x0010 (10 _{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	Maximum subindex	UINT8	RW	> 10 <
F010:01	SubIndex 001	Analog input profile (300)	UINT32	RW	0x0000012C (300 _{dec})
F010:02	SubIndex 002	Analog input profile (300)	UINT32	RW	0x0000012C (300 _{dec})
F010:03	SubIndex 003	Analog input profile (300)	UINT32	RW	0x0000012C (300 _{dec})
F010:04	SubIndex 004	Analog input profile (300)	UINT32	RW	0x0000012C (300 _{dec})
F010:05	SubIndex 005	Analog input profile (300)	UINT32	RW	0x0000012C (300 _{dec})
F010:06	SubIndex 006	Analog input profile (300)	UINT32	RW	0x0000012C (300 _{dec})
F010:07	SubIndex 007	Analog input profile (300)	UINT32	RW	0x0000012C (300 _{dec})
F010:08	SubIndex 008	Analog input profile (342)	UINT32	RW	0x00000156 (342 _{dec})
F010:09	SubIndex 009	Analog input profile (342)	UINT32	RW	0x00000156 (342 _{dec})
F010:10	SubIndex 010	Analog input profile (342)	UINT32	RW	0x00000156 (342 _{dec})

Index F081 Download revision

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

5.7 Notices on analog specifications

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

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

5.7.1 Full scale value (FSV)

An I/O device with an analog input measures over a nominal measuring range that is limited by an upper and a lower limit (initial value and end value); these can usually be taken from the device designation.

The range between the two limits is called the measuring span and corresponds to the equation (end value - initial value). Analogous to pointing devices this is the measuring scale (see IEC 61131) or also the dynamic range.

For analog I/O devices from Beckhoff the rule is that the limit with the largest value is chosen as the full scale value of the respective product (also called the reference value) and is given a positive sign. This applies to both symmetrical and asymmetrical measuring spans.

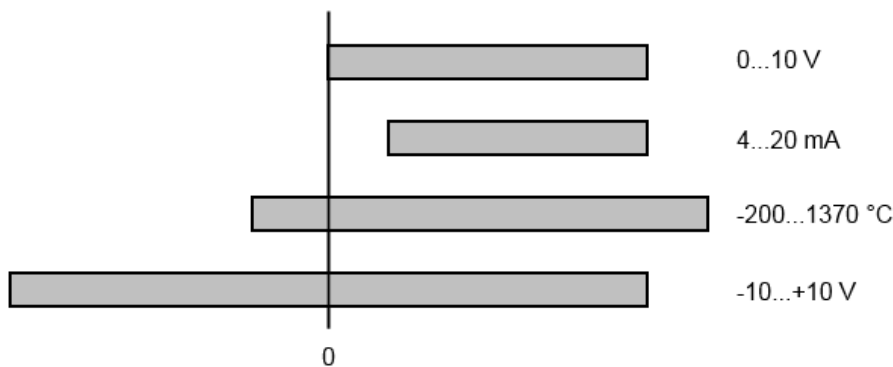


Fig. 178: Full scale value, measuring span

For the above **examples** this means:

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

This applies to analog output terminals/ boxes (and related Beckhoff product groups).

5.7.2 Measuring error/ measurement deviation

The relative measuring error (% of the full scale value) is referenced to the full scale value and is calculated as the quotient of the largest numerical deviation from the true value ('measuring error') referenced to the full scale value.

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

The measuring error is generally valid for the entire permitted operating temperature range, also called the 'usage error limit' and contains random and systematic portions of the referred device (i.e. 'all' influences such as temperature, inherent noise, aging, etc.).

It is always to be regarded as a positive/negative span with \pm , even if it is specified without \pm in some cases.

The maximum deviation can also be specified directly.

Example: Measuring range 0...10 V and measuring error $< \pm 0.3$ % full scale value \rightarrow maximum deviation ± 30 mV in the permissible operating temperature range.

i Lower measuring error

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

This applies to analog output devices.

5.7.3 Temperature coefficient tK [ppm/K]

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

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

The temperature coefficient, when indicated, specified by Beckhoff allows the user to calculate the expected measuring error outside the basic accuracy at 23 °C.

Due to the extensive uncertainty considerations that are incorporated in the determination of the basic accuracy (at 23 °C), Beckhoff recommends a quadratic summation.

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

$$G_{35} = \sqrt{(0.01\%)^2 + (12K \cdot 20 \frac{\text{ppm}}{K})^2} = 0.026\% \text{ full scale value, typ}$$

Remarks: ppm $\triangleq 10^{-6}$ % $\triangleq 10^{-2}$

5.7.4 Single-ended/differential typification

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

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

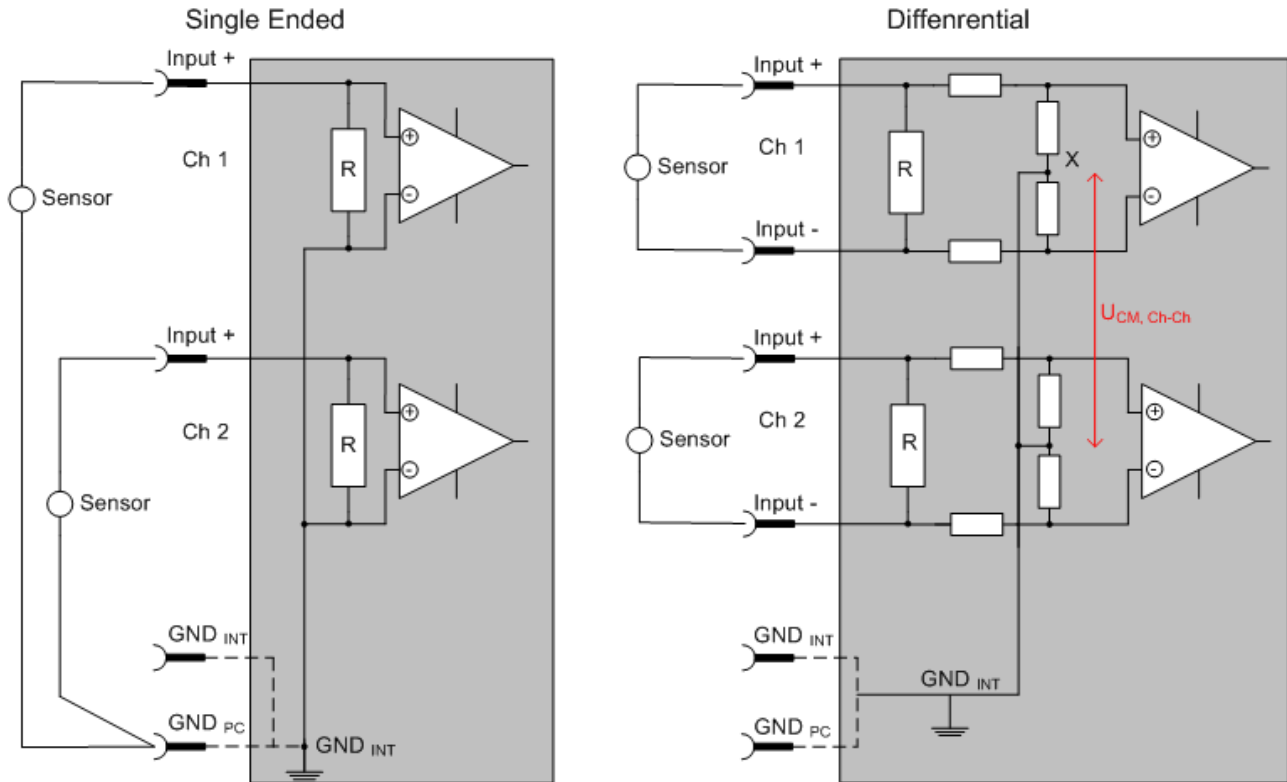


Fig. 179: SE and DIFF module as 2-channel version

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

The basic rule:

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

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

- Beckhoff terminals/ boxes (and related product groups) always feature electrical isolation between the field/analog side and the bus/EtherCAT side. In other words, if two analog terminals/ boxes are not connected via the power contacts (cable), the modules are effectively electrically isolated.
- If channels within a module are electrically isolated, or if a single-channel module has no power contacts, the channels are effectively always differential. See also explanatory notes below. Differential channels are not necessarily electrically isolated.
- Analog measuring channels are subject to technical limits, both in terms of the recommended operating range (continuous operation) and the destruction limit. Please refer to the respective terminal/ box documentation for further details.

Explanation

• differential (DIFF)

- Differential measurement is the most flexible concept. The user can freely choose both connection points, input+/signal potential and input-/reference potential, within the framework of the technical specification.
- A differential channel can also be operated as SE, if the reference potential of several sensors is linked. This interconnection may take place via the system GND.
- Since a differential channel is configured symmetrically internally (cf. Fig. SE and DIFF module as 2-channel variant), there will be a mid-potential (X) between the two supplied potentials that is the same as the internal ground/reference ground for this channel. If several DIFF channels are used in a module without electrical isolation, the technical property V_{CM} (common-mode voltage) indicates the degree to which the mean voltage of the channels may differ.
- The internal reference ground may be accessible as connection point at the terminal/ box, in order to stabilize a defined GND potential in the terminal/ box. In this case it is particularly important to pay attention to the quality of this potential (noiselessness, voltage stability). At this GND point a wire may be connected to make sure that $V_{CM,max}$ is not exceeded in the differential sensor cable. If differential channels are not electrically isolated, usually only one $V_{CM,max}$ is permitted. If the channels are electrically isolated this limit should not apply, and the channels voltages may differ up to the specified separation limit.
- Differential measurement in combination with correct sensor wiring has the special advantage that any interference affecting the sensor cable (ideally the feed and return line are arranged side by side, so that interference signals have the same effect on both wires) has very little effect on the measurement, since the potential of both lines varies jointly (hence the term common mode). In simple terms: Common-mode interference has the same effect on both wires in terms of amplitude and phasing.
- Nevertheless, the suppression of common-mode interference within a channel or between channels is subject to technical limits, which are specified in the technical data.
- Further helpfully information on this topic can be found on the documentation page *Configuration of 0/4..20 mA differential inputs* (see documentation for the EL30xx terminals, for example).

• Single Ended (SE)

- If the analog circuit is designed as SE, the input/reference wire is internally fixed to a certain potential that cannot be changed. This potential must be accessible from outside on at least one point for connecting the reference potential, e.g. via the power contacts (cable).
- In other words, in situations with several channels SE offers users the option to avoid returning at least one of the two sensor cables to the terminal/ box (in contrast to DIFF). Instead, the reference wire can be consolidated at the sensors, e.g. in the system GND.
- A disadvantage of this approach is that the separate feed and return line can result in voltage/ current variations, which a SE channel may no longer be able to handle. See common-mode interference. A V_{CM} effect cannot occur, since the module channels are internally always 'hard-wired' through the input/reference potential.

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

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

Self-supplied sensors

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

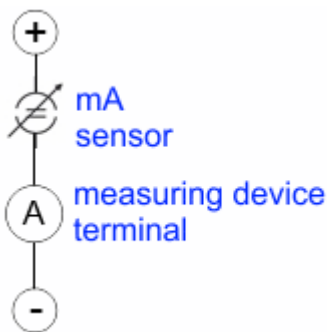


Fig. 180: 2-wire connection

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

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

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

Externally supplied sensors

- 3- and 4-wire connection see Fig. *Connection of externally supplied sensors*, cf. IEC60381-1
 - the sensor draws the energy/operating voltage for its own operation from 2 supply cables of its own. One or two further sensor cables are used for the signal transmission of the current loop:
 - 1 sensor cable: according to the Beckhoff terminology such sensors are to be connected to **'single-ended' inputs** in 3 cables with +/-Signal lines and if necessary FE/shield
 - 2 sensor cables: for sensors with 4-wire connection based on +supply/-supply/+signal/-signal, check whether +signal can be connected to +supply or –signal to –supply.
 - Yes: then you can connect accordingly to a Beckhoff **'single-ended' input**.
 - No: the Beckhoff **'differential' input** for +Signal and –Signal is to be selected; +Supply and –Supply are to be connected via additional cables.
- It is important to refer to the information page *Configuration of 0/4..20 mA differential inputs* (see documentation for the EL30xx terminals, for example)!

Note: expert organizations such as NAMUR demand a usable measuring range <4 mA/>20 mA for error detection and adjustment, see also NAMUR NE043.

The Beckhoff device documentation must be consulted in order to see whether the respective device supports such an extended signal range.

Usually there is an internal diode existing within unipolar terminals/ boxes (and related product groups), in this case the polarity/direction of current have to be observed.

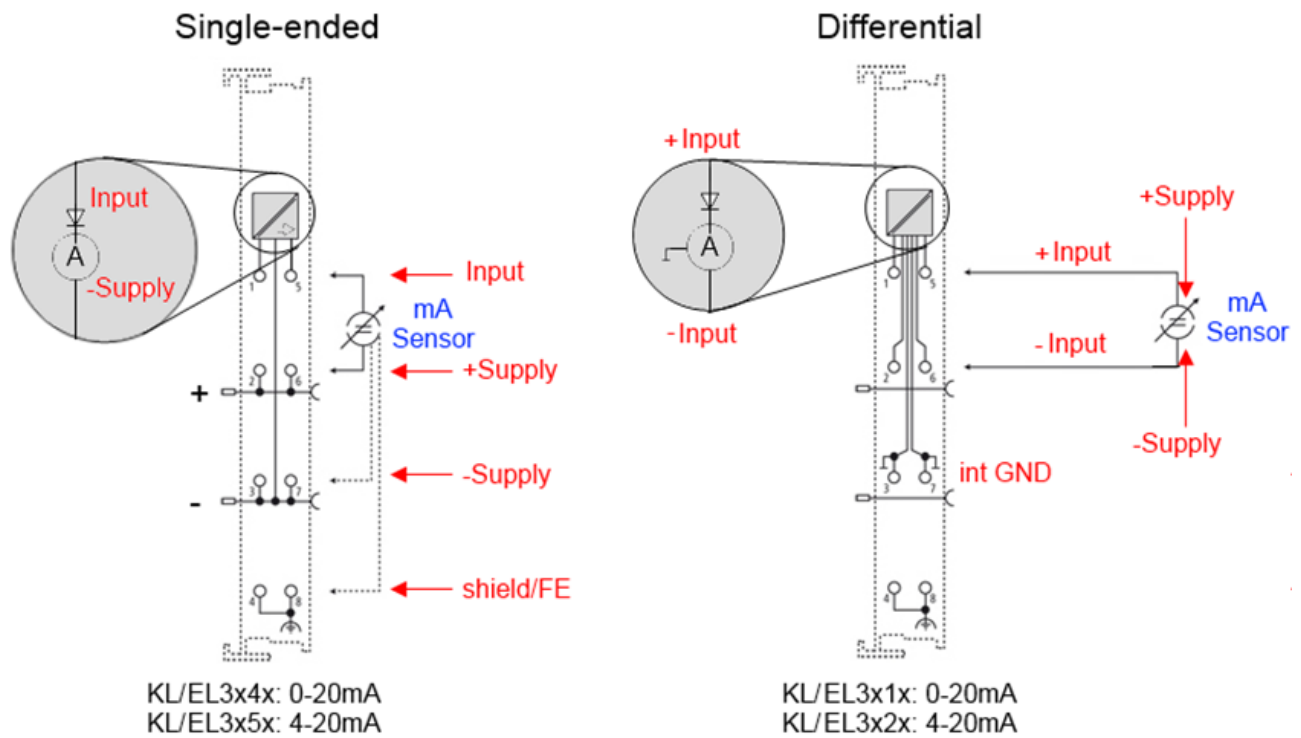


Fig. 181: Connection of externally supplied sensors

Classification of the Beckhoff terminals/ boxes - Beckhoff 0/4-20 mA terminals/ boxes (and related product groups) are available as **differential** and **single-ended** terminals/ boxes (and related product groups):

Single-ended

EL3x4x: 0-20 mA, EL3x5x: 4-20 mA; KL and related product groups exactly the same
Preferred current direction because of internal diode
Designed for the connection of externally-supplied sensors with a 3/4-wire connection
Designed for the connection of self-supplied sensors with a 2-wire connection

Differential

EL3x1x: 0-20 mA, EL3x2x: 4-20 mA; KL and related product groups exactly the same
Preferred current direction because of internal diode
The terminal/ box is a passive differential current measuring device; passive means that the sensor is not supplied with power.

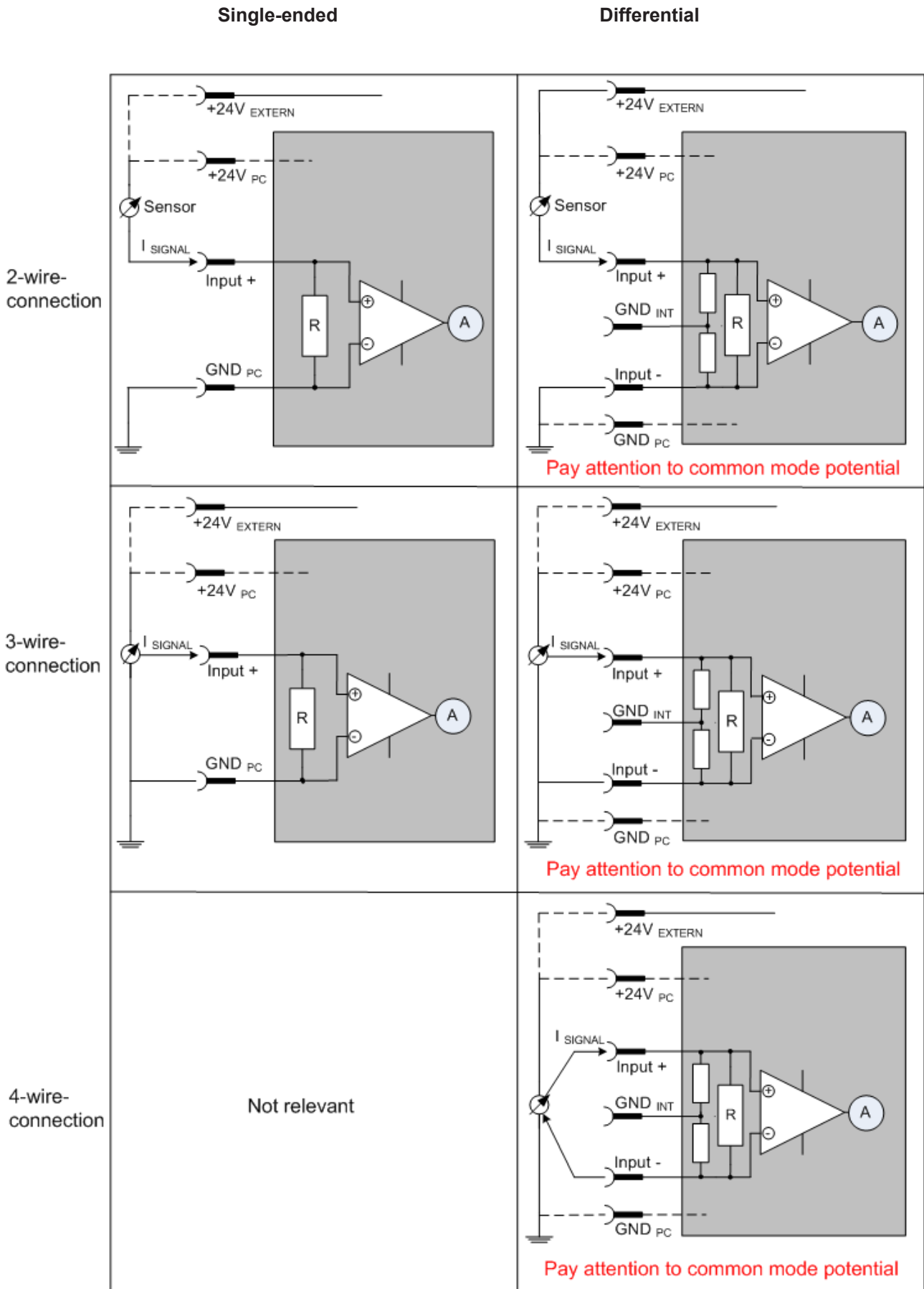


Fig. 182: 2-, 3- and 4-wire connection at single-ended and differential inputs

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

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

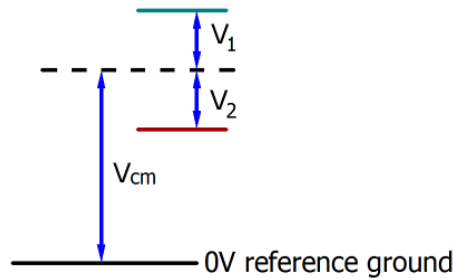


Fig. 183: Common-mode voltage (V_{cm})

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

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

The reference ground is usually accessible at or near the terminal/ box, e.g. at the terminal contacts, power contacts (cable) or a mounting rail. Please refer to the documentation regarding positioning. The reference ground should be specified for the device under consideration.

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

Reference ground samples for Beckhoff IO devices:

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

5.7.6 Dielectric strength

A distinction should be made between:

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

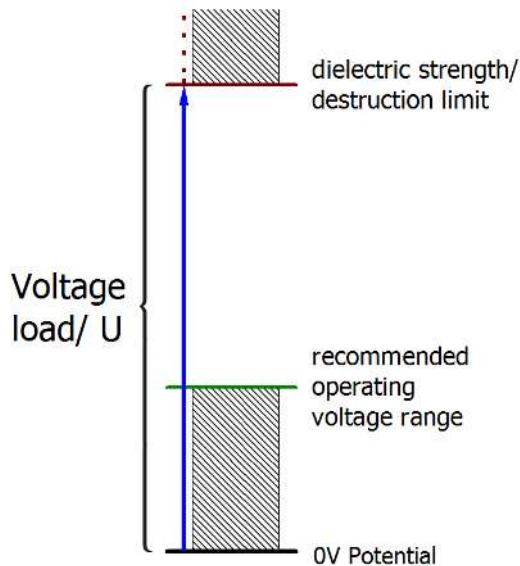


Fig. 184: Recommended operating voltage range

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

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

5.7.7 Temporal aspects of analog/digital conversion

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

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

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

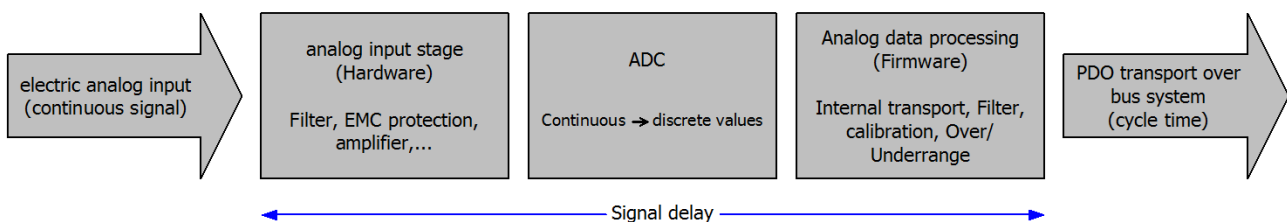


Fig. 185: Signal processing analog input

Two aspects are crucial from a user perspective:

- “How often do I receive new values?”, i.e. a sampling rate in terms of speed with regard to the device/channel
- What delay does the (whole) AD conversion of the device/channel cause?
I.e. the hardware and firmware components in its entirety. For technological reasons, the signal characteristics must be taken into account when determining this information: the run times through the system differ, depending on the signal frequency.

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

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

1. Minimum conversion time [ms, µs]

This is the reciprocal value of the maximum **sampling rate** [sps, samples per second]:

Indicates how often the analog channel makes a newly detected process data value available for collection by the fieldbus. Whether the fieldbus (EtherCAT, K-bus) fetches the value with the same speed (i.e. synchronous), or more quickly (if the AI channel operates in slow FreeRun mode) or more slowly (e.g. with oversampling), is then a question of the fieldbus setting and which modes the AI device supports.

For EtherCAT devices the so-called toggle bit indicates (by toggling) for the diagnostic PDOs when a newly determined analog value is available.

Accordingly, a maximum conversion time, i.e. a smallest sampling rate supported by the AI device, can be specified.

Corresponds to IEC 61131-2, section 7.10.2 2, “Sampling repeat time”

2. Typical signal delay

Corresponds to IEC 61131-2, section 7.10.2 1, “Sampling duration”. From this perspective it includes all internal hardware and firmware components, but not “external” delay components from the fieldbus or the controller (TwinCAT).

This delay is particularly relevant for absolute time considerations, if AI channels also provide a time stamp that corresponds to the amplitude value – which can be assumed to match the physically prevailing amplitude value at the time.

Due to the frequency-dependent signal delay time, a dedicated value can only be specified for a given signal. The value also depends on potentially variable filter settings of the channel.

A typical characterization in the device documentation may be:

2.1 Signal delay (step response)

Keywords: Settling time

The square wave signal can be generated externally with a frequency generator (note impedance!)

The 90 % limit is used as detection threshold.

The signal delay [ms, µs] is then the time interval between the (ideal) electrical square wave signal and the time at which the analog process value has reached the 90 % amplitude.

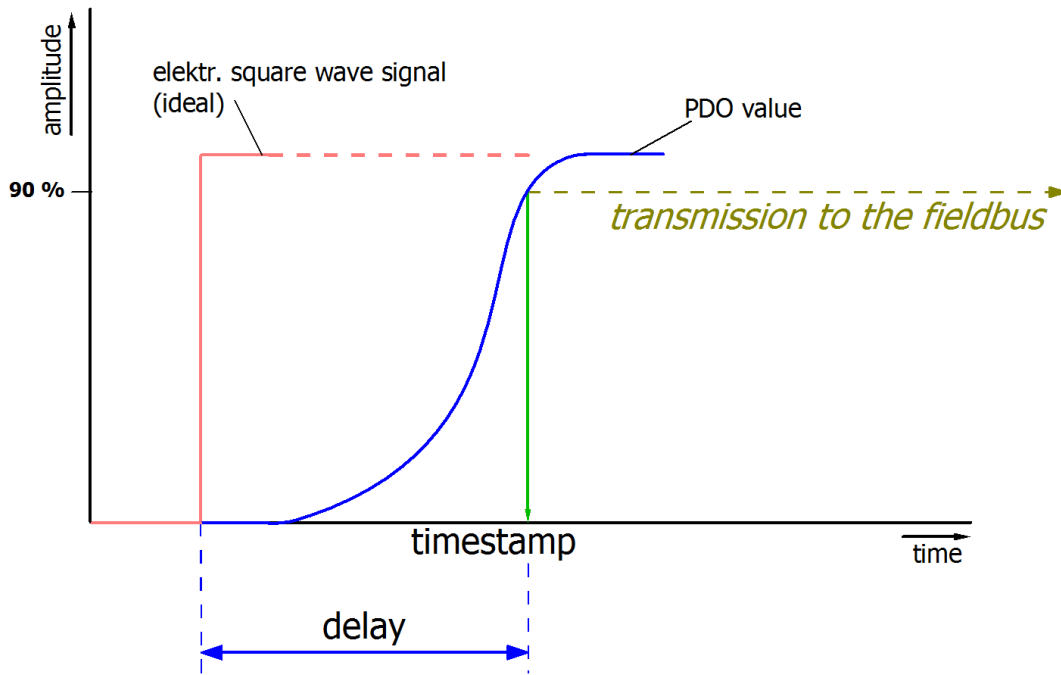


Fig. 186: Diagram signal delay (step response)

2.2 Signal delay (linear)

Keyword: Group delay

Describes the delay of a signal with constant frequency

A test signal can be generated externally with a frequency generator, e.g. as sawtooth or sine. A simultaneous square wave signal would be used as reference.

The signal delay [ms, μ s] is then the interval between the applied electrical signal with a particular amplitude and the moment at which the analog process value reaches the same value.

A meaningful range must be selected for the test frequency, e.g. 1/20 of the maximum sampling rate.

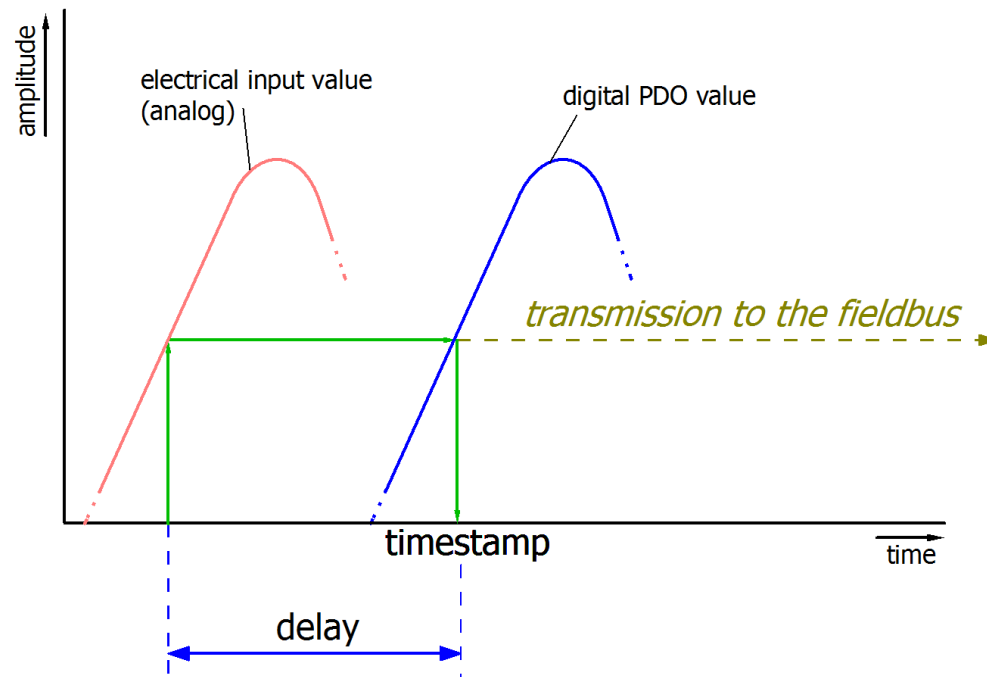


Fig. 187: Diagram signal delay (linear)

3. Additional Information

May be provided in the specification, e.g.

- Actual sampling rate of the ADC (if different from the channel sampling rate)
- Time correction values for run times with different filter settings
- etc.

6 Diagnosis

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:

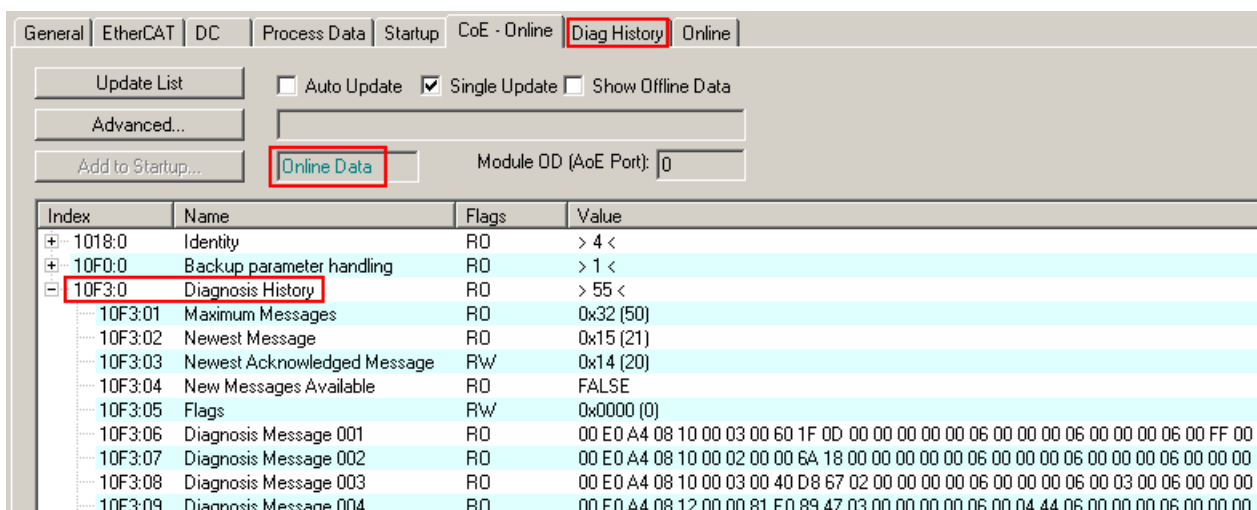


Fig. 188: *DiagMessages* in the CoE

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

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.

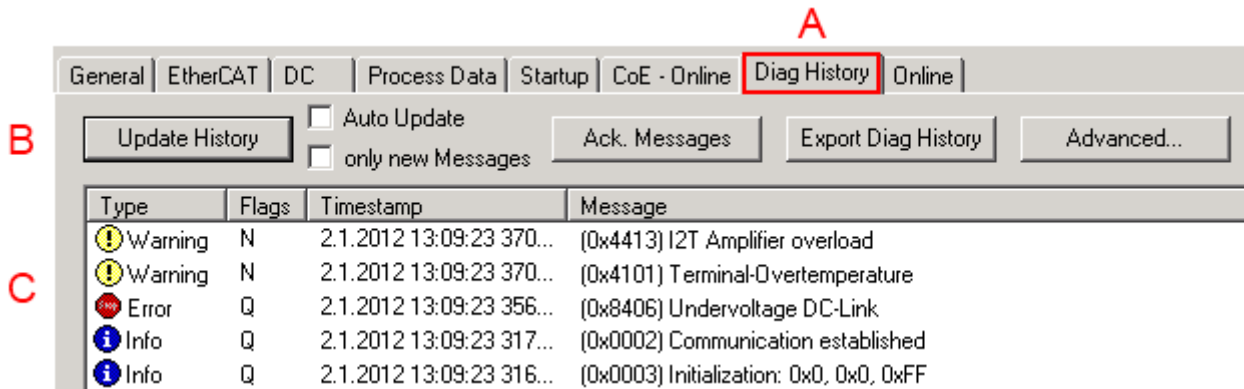


Fig. 189: Implementation of the DiagMessage system in the TwinCAT System Manager

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.

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 should be specified 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 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
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)	- The amplifier is being operated outside the specification - The I2T-model of the amplifier is incorrectly parameterized

Text ID	Type	Place	Text Message	Additional comment
0x4414	Warning	Drive	I2T-Model Motor over-load (Warning)	- The motor is being operated outside the parameterized rated values - The I2T-model of the motor is incorrectly parameterized
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. - 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)	- 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	

Text ID	Type	Place	Text Message	Additional comment
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
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 supply error	Encoder power supply unit error
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
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)	- 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)	- 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).

Text ID	Type	Place	Text Message	Additional comment
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
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

7 Appendix

7.1 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.

Storage locations

An EtherCAT slave stores operating data in up to 3 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 3 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

- Firmware downloads to an EtherCAT device must not be interrupted
- Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.
- 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.

7.1.1 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:

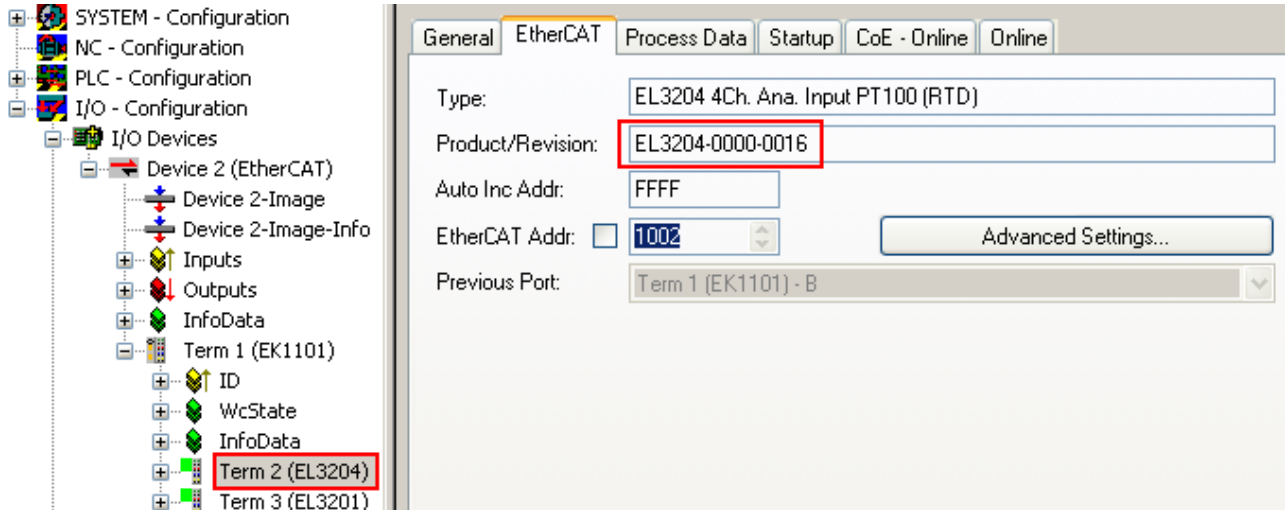


Fig. 190: Device identifier consisting of name EL3204-0000 and revision -0016

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:

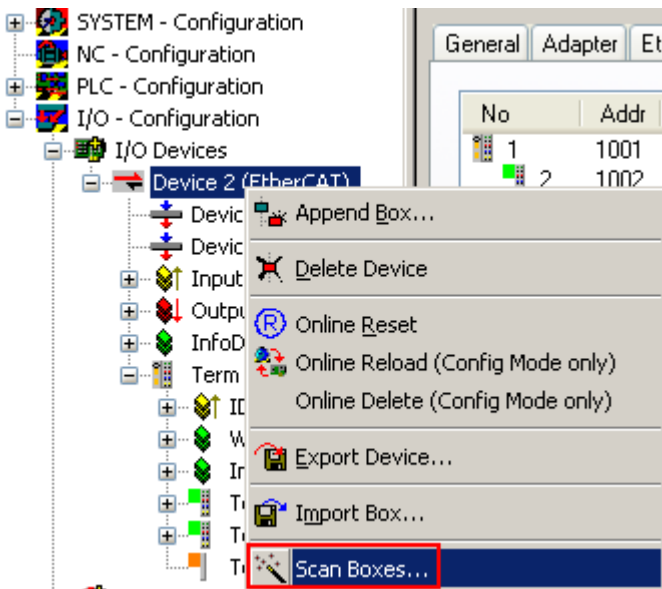


Fig. 191: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 192: Configuration is identical

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

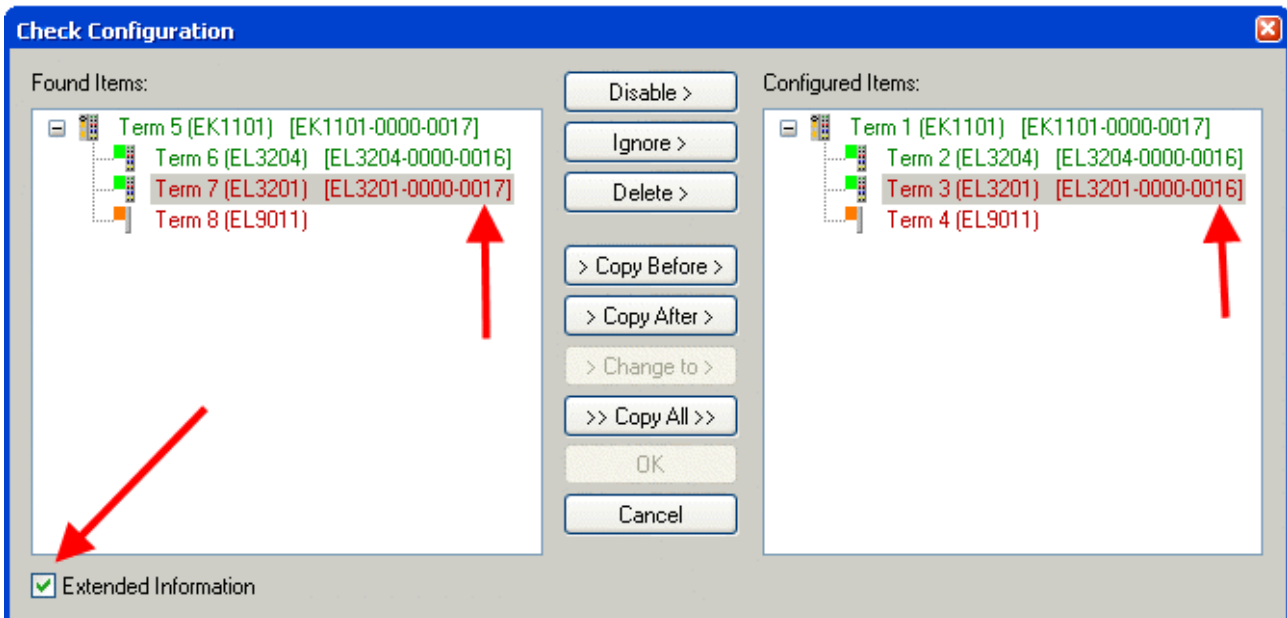


Fig. 193: Change dialog

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*

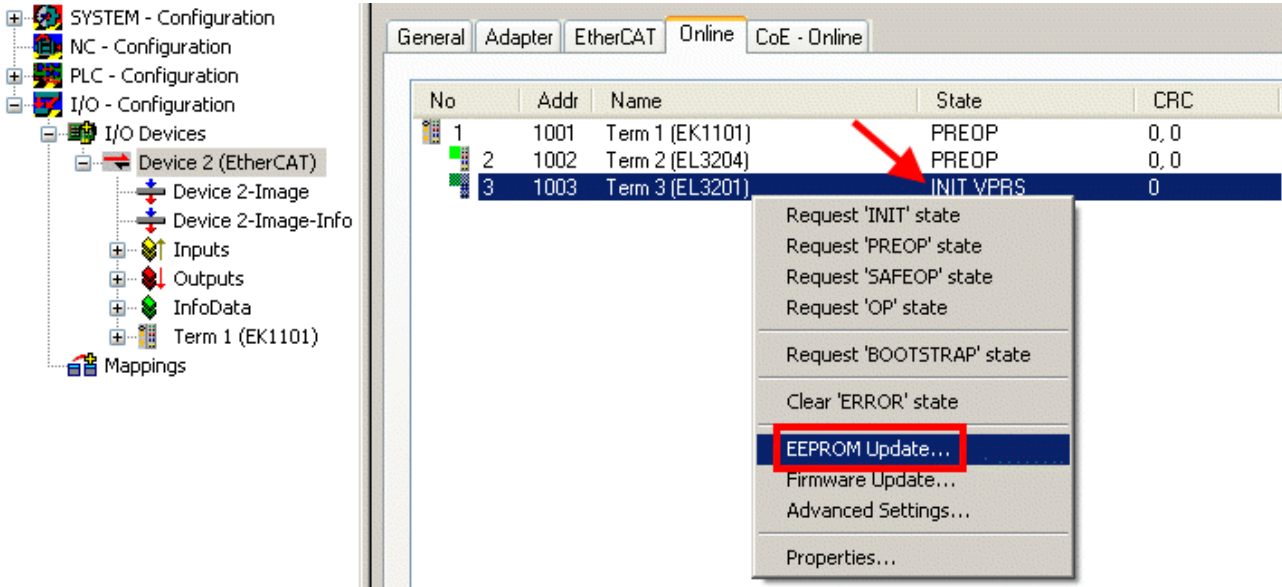


Fig. 194: EEPROM Update

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.

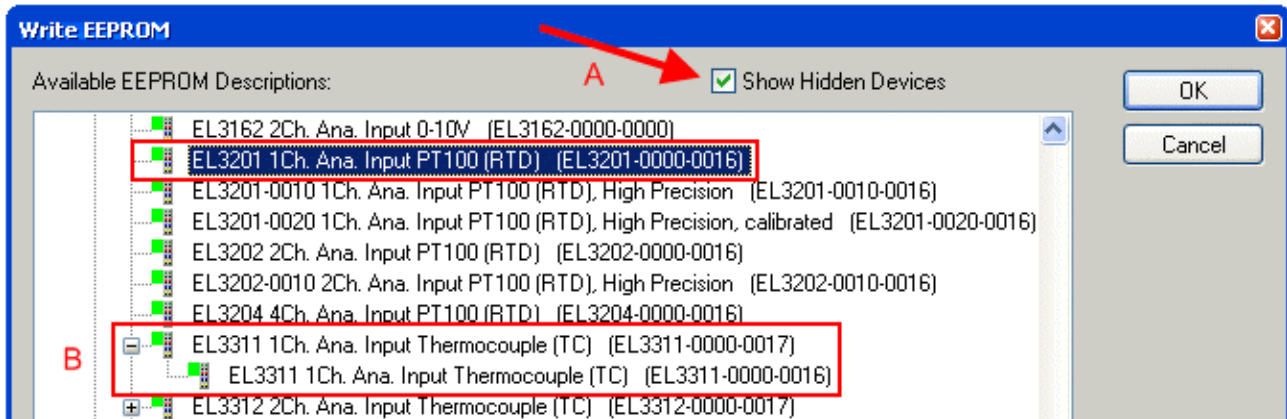


Fig. 195: Selecting the new ESI

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.

7.1.2 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.

Index	Name	Flags	Value
1000	Device type	RO	0x01401389 (20976521)
1008	Device name	RO	EL3204-0000
1009	Hardware version	RO	00
100A	Software version	RO	03
1011:0	Restore default parameters	RU	> 1 <

Fig. 196: 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 *All Objects*.

7.1.3 Updating controller firmware *.efw

i **CoE directory**

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

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

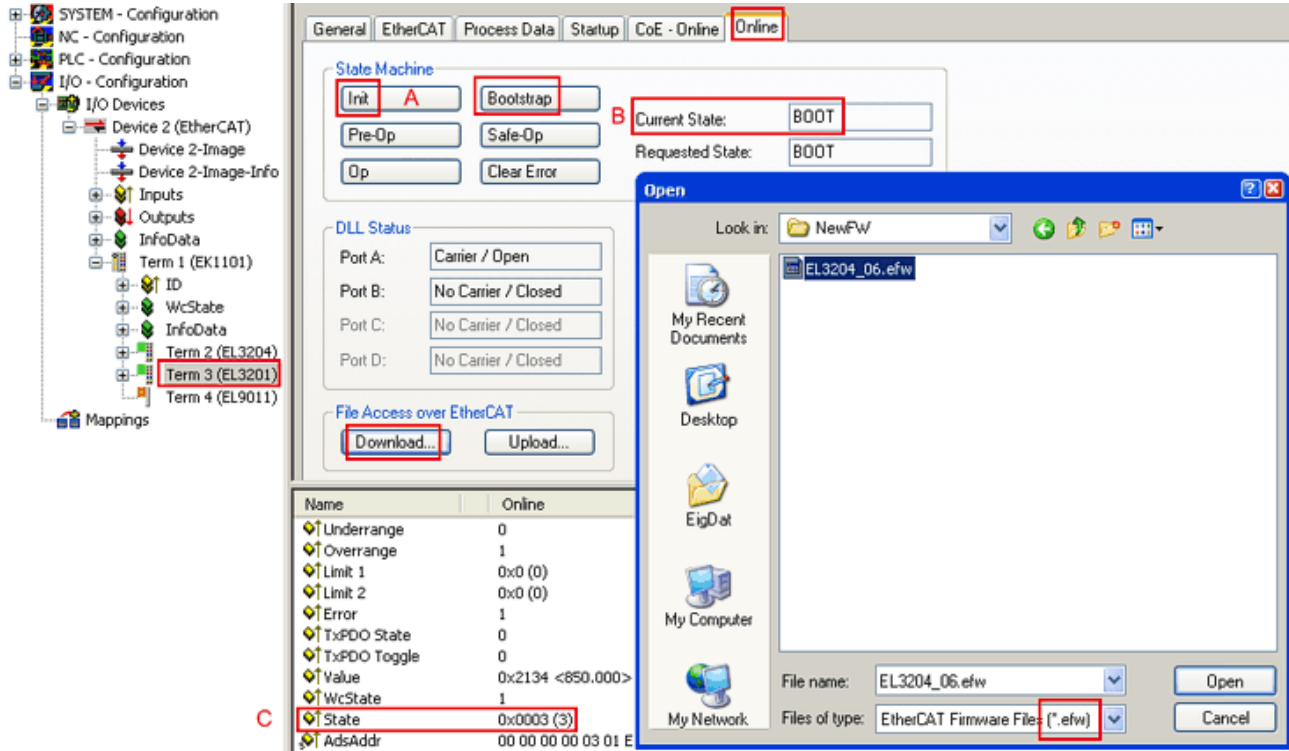
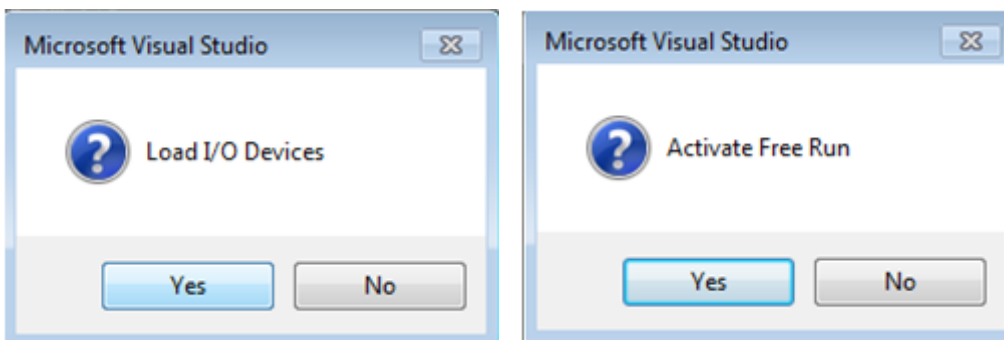


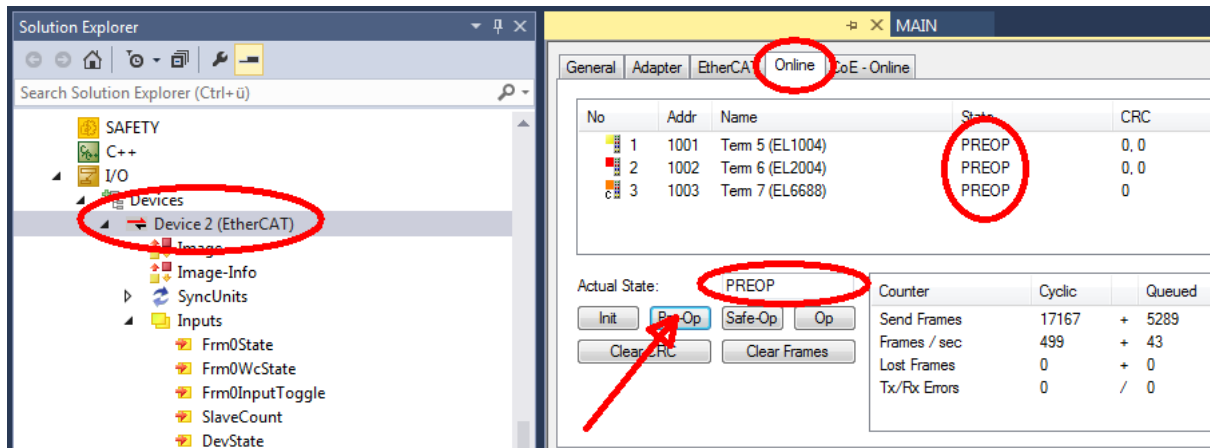
Fig. 197: Firmware Update

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

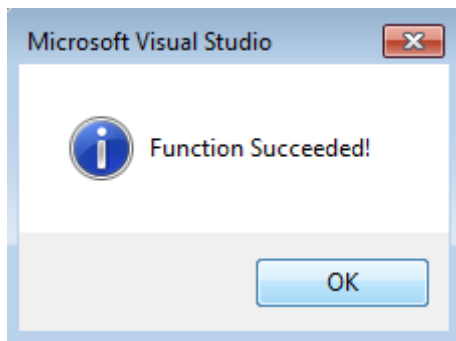
- Switch TwinCAT system to ConfigMode/FreeRun with cycle time ≥ 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.



- 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.

7.1.4 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.

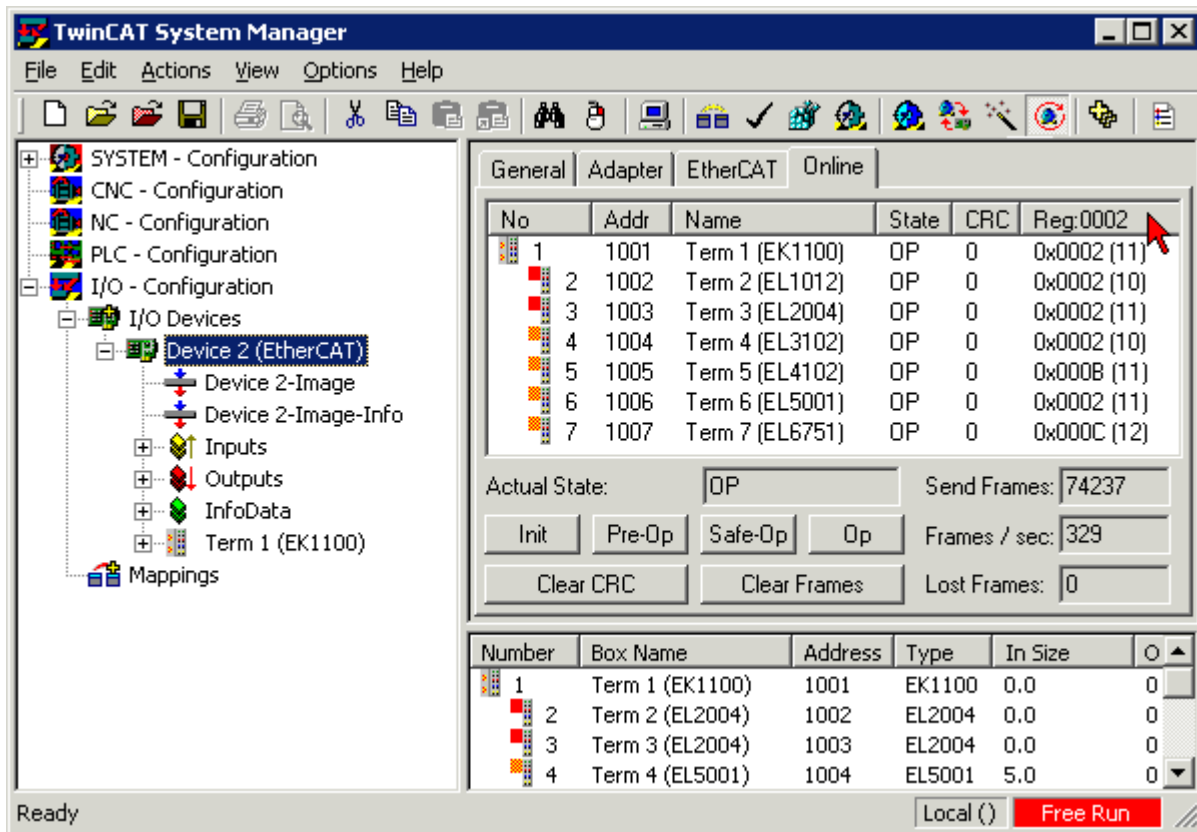


Fig. 198: FPGA firmware version definition

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

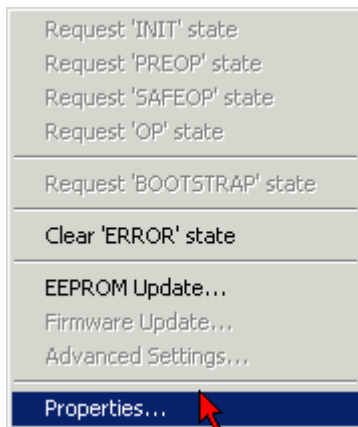


Fig. 199: Context menu *Properties*

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.

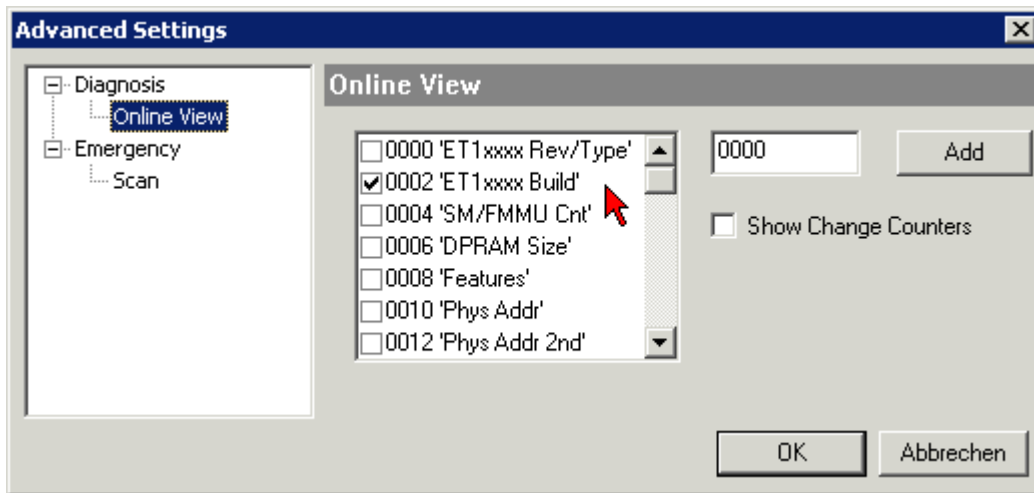


Fig. 200: Dialog *Advanced Settings*

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.

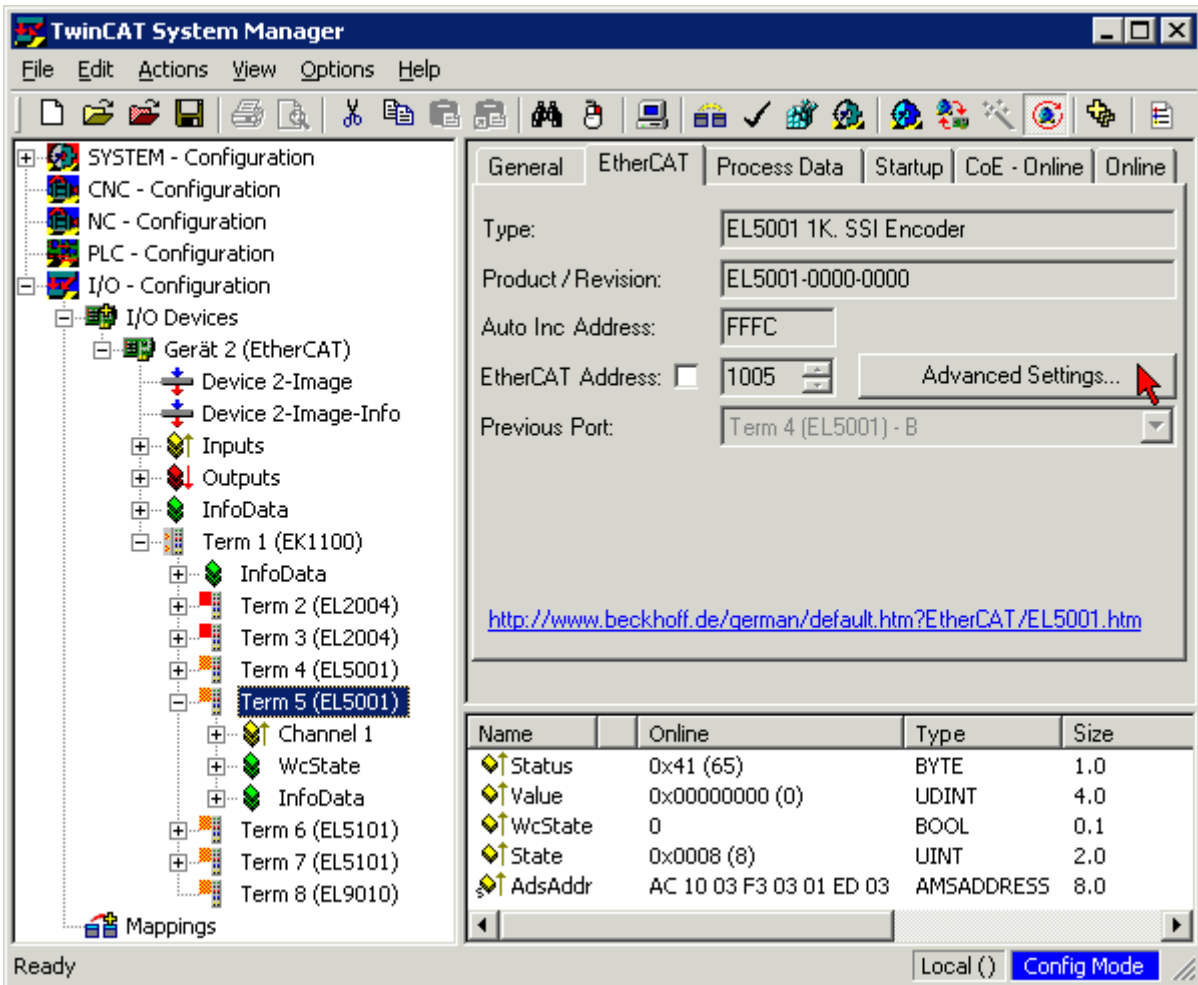
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

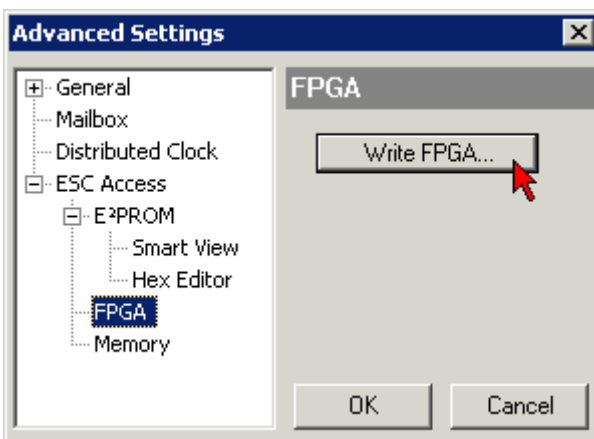
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- 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.

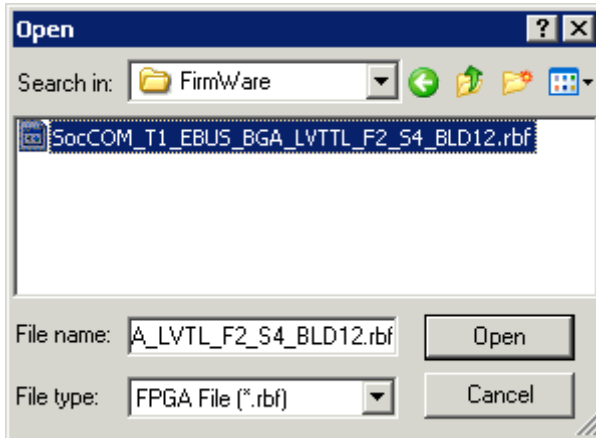
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



- 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!

7.1.5 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.

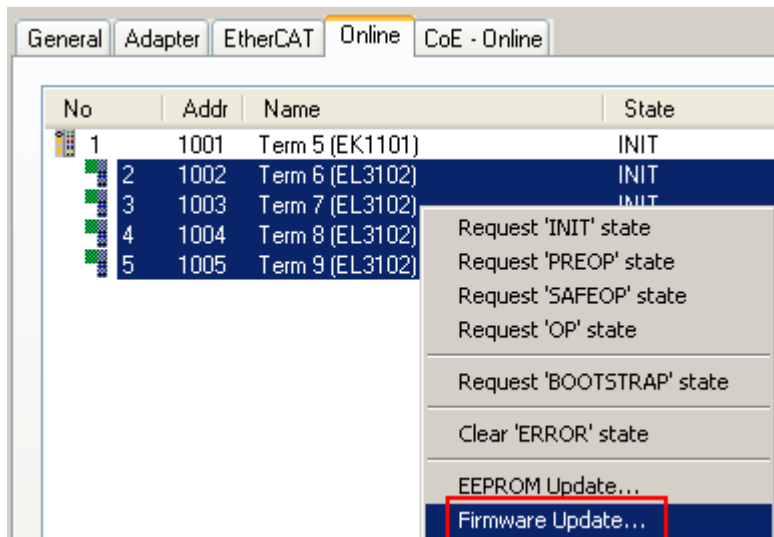


Fig. 201: Multiple selection and firmware update

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

7.2 Firmware compatibility

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

Note

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

NOTE			
Risk of damage to the device!			
Pay attention to the instructions for firmware updates on the separate page [▶ 170] . 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!			

EL3783			
Hardware (HW)	Firmware (FW)	Revision no.	Release date
00 - 01	01	EL3783-0000-0016	2016/02
	02		2016/04
02 - 03	03		2016/10
	04		
	05		2017/02
	06	EL3783-0000-0018	2017/02
03*	07*		2017/06

*) 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 on EL3783 firmware update



Secondary voltage necessary for firmware updates

In order to change the firmware of the EL3783, 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.

7.3 Restoring the delivery state

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

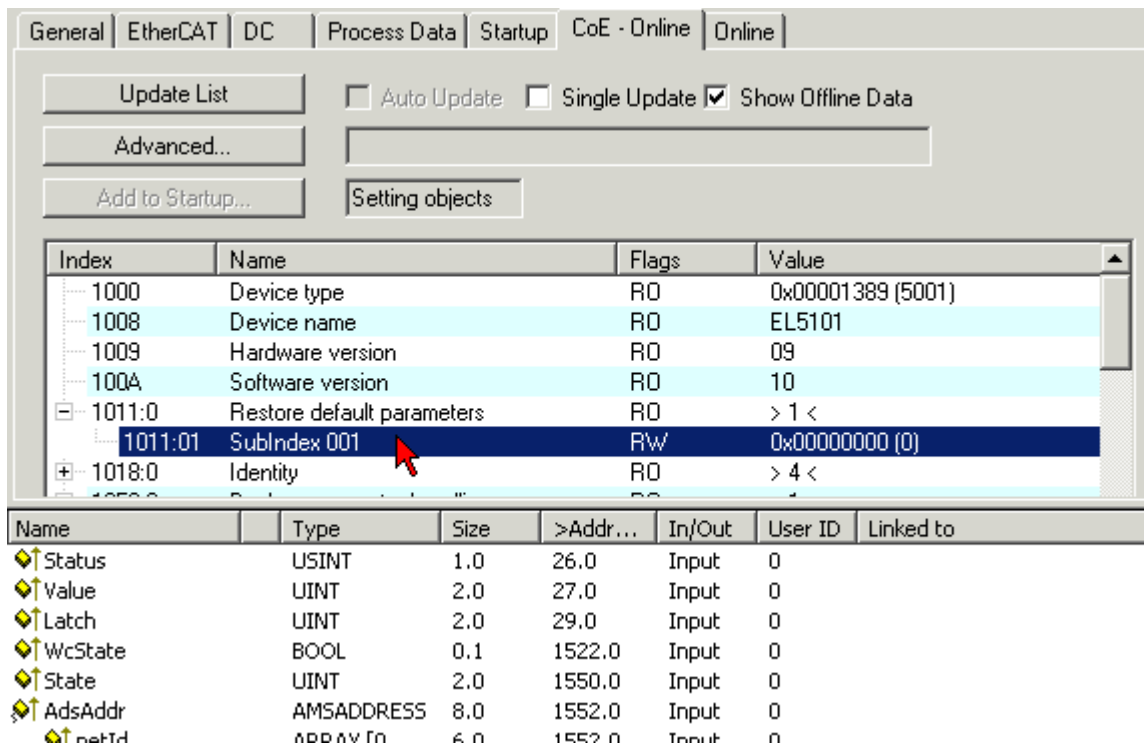


Fig. 202: Selecting the "Restore default parameters" PDO

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.

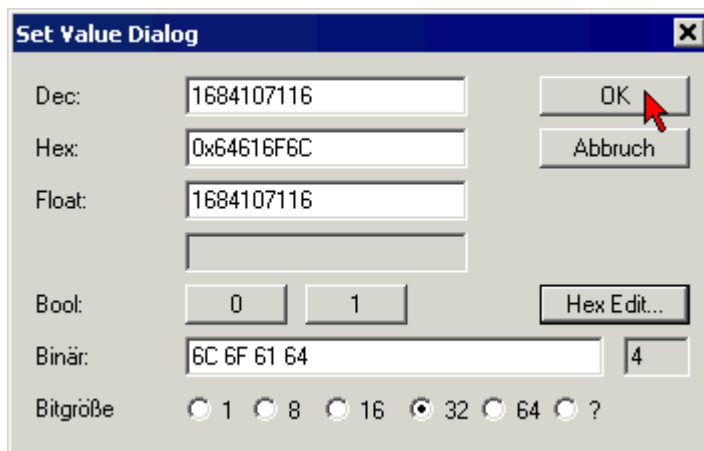


Fig. 203: Entering a restore value in the Set Value dialog

● 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:

<http://www.beckhoff.com>

You will also find further documentation for Beckhoff components there.

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

Phone:	+49(0)5246/963-0
Fax:	+49(0)5246/963-198
e-mail:	info@beckhoff.com

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(0)5246/963-157
Fax:	+49(0)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(0)5246/963-460
Fax:	+49(0)5246/963-479
e-mail:	service@beckhoff.com

List of illustrations

Fig. 1	EL5021 EL terminal, standard IP20 IO device with serial/ batch number and revision ID (since 2014/01).....	9
Fig. 2	EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number.....	9
Fig. 3	CU2016 switch with serial/ batch number.....	9
Fig. 4	EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418	10
Fig. 5	EP1258-00001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102.....	10
Fig. 6	EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070	10
Fig. 7	EL2904 IP20 safety terminal with batch number/ date code 50110302 and unique serial number 00331701.....	10
Fig. 8	ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201.....	11
Fig. 9	EL3783	13
Fig. 10	Voltages in three-phase mains supply of Central Europe.....	15
Fig. 11	System manager current calculation	18
Fig. 12	EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog	19
Fig. 13	States of the EtherCAT State Machine.....	21
Fig. 14	"CoE Online " tab	23
Fig. 15	Startup list in the TwinCAT System Manager	24
Fig. 16	Offline list.....	25
Fig. 17	Online list	25
Fig. 18	Spring contacts of the Beckhoff I/O components.....	28
Fig. 19	Attaching on mounting rail	29
Fig. 20	Disassembling of terminal.....	30
Fig. 21	Power contact on left side.....	31
Fig. 22	Standard wiring.....	32
Fig. 23	Pluggable wiring	32
Fig. 24	High Density Terminals.....	33
Fig. 25	Mounting a cable on a terminal connection	34
Fig. 26	Correct positioning	36
Fig. 27	Incorrect positioning.....	36
Fig. 28	Recommended minimum distances for standard installation position	37
Fig. 29	EL3783	38
Fig. 30	Measurement in a 3-phase, 4-conductor grid, asymmetrical load	40
Fig. 31	Measurement in a 3-phase, 4-conductor grid with 3 voltage transformers, asymmetrical load...	40
Fig. 32	Measurement in a 3-phase, 3-conductor grid, asymmetrical load	41
Fig. 33	Relationship between user side (commissioning) and installation.....	43
Fig. 34	Control configuration with Embedded PC, input (EL1004) and output (EL2008)	44
Fig. 35	Initial TwinCAT 2 user interface.....	45
Fig. 36	Selection of the target system	46
Fig. 37	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	46
Fig. 38	Select "Scan Devices..."	47
Fig. 39	Automatic detection of I/O devices: selection the devices to be integrated.....	47
Fig. 40	Mapping of the configuration in the TwinCAT 2 System Manager.....	48
Fig. 41	Reading of individual terminals connected to a device.....	48

Fig. 42	TwinCAT PLC Control after startup	49
Fig. 43	Sample program with variables after a compile process (without variable integration)	50
Fig. 44	Appending the TwinCAT PLC Control project	50
Fig. 45	PLC project integrated in the PLC configuration of the System Manager	51
Fig. 46	Creating the links between PLC variables and process objects	51
Fig. 47	Selecting PDO of type BOOL	52
Fig. 48	Selecting several PDOs simultaneously: activate "Continuous" and "All types"	52
Fig. 49	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample	53
Fig. 50	Choose target system (remote)	54
Fig. 51	PLC Control logged in, ready for program startup	55
Fig. 52	Initial TwinCAT 3 user interface	56
Fig. 53	Create new TwinCAT project	56
Fig. 54	New TwinCAT3 project in the project folder explorer	57
Fig. 55	Selection dialog: Choose the target system	57
Fig. 56	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	58
Fig. 57	Select "Scan"	58
Fig. 58	Automatic detection of I/O devices: selection the devices to be integrated	59
Fig. 59	Mapping of the configuration in VS shell of the TwinCAT3 environment	59
Fig. 60	Reading of individual terminals connected to a device	60
Fig. 61	Adding the programming environment in "PLC"	61
Fig. 62	Specifying the name and directory for the PLC programming environment	61
Fig. 63	Initial "Main" program of the standard PLC project	62
Fig. 64	Sample program with variables after a compile process (without variable integration)	63
Fig. 65	Start program compilation	63
Fig. 66	Creating the links between PLC variables and process objects	64
Fig. 67	Selecting PDO of type BOOL	64
Fig. 68	Selecting several PDOs simultaneously: activate "Continuous" and "All types"	65
Fig. 69	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample	65
Fig. 70	TwinCAT development environment (VS shell): logged-in, after program startup	66
Fig. 71	System Manager "Options" (TwinCAT 2)	67
Fig. 72	Call up under VS Shell (TwinCAT 3)	68
Fig. 73	Overview of network interfaces	68
Fig. 74	EtherCAT device properties(TwinCAT 2): click on „Compatible Devices...“ of tab “Adapter”	68
Fig. 75	Windows properties of the network interface	69
Fig. 76	Exemplary correct driver setting for the Ethernet port	69
Fig. 77	Incorrect driver settings for the Ethernet port	70
Fig. 78	TCP/IP setting for the Ethernet port	71
Fig. 79	Identifier structure	72
Fig. 80	OnlineDescription information window (TwinCAT 2)	73
Fig. 81	Information window OnlineDescription (TwinCAT 3)	73
Fig. 82	File OnlineDescription.xml created by the System Manager	74
Fig. 83	Indication of an online recorded ESI of EL2521 as an example	74
Fig. 84	Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)	74
Fig. 85	Using the ESI Updater (>= TwinCAT 2.11)	76
Fig. 86	Using the ESI Updater (TwinCAT 3)	76
Fig. 87	Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)	77

Fig. 88	Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3).....	77
Fig. 89	Selecting the Ethernet port	77
Fig. 90	EtherCAT device properties (TwinCAT 2)	78
Fig. 91	Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3).....	78
Fig. 92	Selection dialog for new EtherCAT device	79
Fig. 93	Display of device revision	79
Fig. 94	Display of previous revisions	80
Fig. 95	Name/revision of the terminal.....	80
Fig. 96	EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3).....	81
Fig. 97	Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3).....	82
Fig. 98	Scan Devices (left: TwinCAT 2; right: TwinCAT 3).....	82
Fig. 99	Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3).....	82
Fig. 100	Detected Ethernet devices	83
Fig. 101	Example default state	83
Fig. 102	Installing EtherCAT terminal with revision -1018	84
Fig. 103	Detection of EtherCAT terminal with revision -1019.....	84
Fig. 104	Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)	84
Fig. 105	Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3).....	85
Fig. 106	Scan progress exemplary by TwinCAT 2	85
Fig. 107	Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3).....	85
Fig. 108	Displaying of "Free Run" and "Config Mode" toggling right below in the status bar	85
Fig. 109	TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)	85
Fig. 110	Online display example	86
Fig. 111	Faulty identification	86
Fig. 112	Identical configuration (left: TwinCAT 2; right: TwinCAT 3).....	87
Fig. 113	Correction dialog	87
Fig. 114	Name/revision of the terminal.....	88
Fig. 115	Correction dialog with modifications	89
Fig. 116	Dialog "Change to Compatible Type..." (left: TwinCAT 2; right: TwinCAT 3).....	89
Fig. 117	TwinCAT 2 Dialog Change to Alternative Type	89
Fig. 118	Branch element as terminal EL3751.....	90
Fig. 119	"General" tab.....	90
Fig. 120	„EtherCAT“ tab.....	91
Fig. 121	"Process Data" tab.....	92
Fig. 122	Configuring the process data.....	93
Fig. 123	„Startup“ tab.....	94
Fig. 124	"CoE – Online" tab.....	95
Fig. 125	Dialog "Advanced settings"	96
Fig. 126	„Online“ tab.....	96
Fig. 127	"DC" tab (Distributed Clocks).....	97
Fig. 128	Selection of the diagnostic information of an EtherCAT Slave	99
Fig. 129	Basic EtherCAT Slave Diagnosis in the PLC.....	100
Fig. 130	EL3102, CoE directory	102
Fig. 131	Example of commissioning aid for a EL3204	103

Fig. 132	Default behaviour of the System Manager	104
Fig. 133	Default target state in the Slave	104
Fig. 134	PLC function blocks	105
Fig. 135	Illegally exceeding the E-Bus current	106
Fig. 136	Warning message for exceeding E-Bus current	106
Fig. 137	Opening the *. tzip archive.....	108
Fig. 138	Search of the existing HW configuration for the EtherCAT configuration of the example	108
Fig. 139	Switchover characteristic, DYN0 - DYN3.....	110
Fig. 140	Oversampling PDO of the EL37xx series and in the comparison with EL31xx	112
Fig. 141	Adding a Scope project into an already existing project.....	113
Fig. 142	Representation of a created PLC array variable („aUI_Samples“) to link with oversampling PDOs of EL3773.....	114
Fig. 143	Insertion of a free task	115
Fig. 144	Task property "Create symbols" must be activated.....	115
Fig. 145	Insertion of variable "Var 1" fitting to the oversampling (-factor).....	116
Fig. 146	Set up the link of the PLC array variable (left: for the last preceding paragraph Step 2a, right: for the last preceding paragraph Step 2b)	116
Fig. 147	Select the EL3773 PDO "L1 Voltage Samples" to create a link to the PLC array variable „aUI_Samples“.....	116
Fig. 148	Selection of the oversampling variable with the target browser.....	117
Fig. 149	Appending the variable "aUI_Samples“ below “axis” within the scope project of the solution explorer.....	117
Fig. 150	Example of recording a sine signal with 10 x oversampling at 1 ms measurement cycle time ...	118
Fig. 151	Activation of the ADS server of the EtherCAT device (TwinCAT 3)	119
Fig. 152	Direct access to PDOs of the terminal by ScopeView.....	119
Fig. 153	PLC declaration	121
Fig. 154	PLC in the System Manager.....	121
Fig. 155	Add Variable Type	121
Fig. 156	Definition of the variable type	122
Fig. 157	Overview of declared types	122
Fig. 158	Linking	123
Fig. 159	Array variables of an oversampling terminal.....	123
Fig. 160	Settings in the additional task.....	123
Fig. 161	Variable browser up to the array VAR152	124
Fig. 162	AddSymbol on the array	124
Fig. 163	Channel settings	124
Fig. 164	Activation of the marks	125
Fig. 165	Illustration of a 10 x oversampling variable of the EL3751 by the Scope2	126
Fig. 166	Activation of the ADS server of the EtherCAT Device (TwinCAT 2).....	127
Fig. 167	Direct access of the Scope2 to the terminal's PDOs	127
Fig. 168	Automatically calculated array variable (red) in the Scope2.....	128
Fig. 169	Process data flow EL3783	129
Fig. 170	Linking of 2-bit variable to additional task.....	130
Fig. 171	User settings and vendor calibration in the CoE online directory	131
Fig. 172	EL3783 Selection dialog "Predefined PDO Assignment"	133
Fig. 173	Process image of the EL3783 in the TwinCAT System Manager.....	134
Fig. 174	EL3783 Selection dialog "Predefined PDO Assignment"	136

Fig. 175	Setting of the oversampling rates via System Manager, "DC" tab.....	137
Fig. 176	Deactivation of the Distributed Clock in the "DC" tab	137
Fig. 177	EL3783 Selection dialog "Predefined PDO Assignment"	137
Fig. 178	Full scale value, measuring span	148
Fig. 179	SE and DIFF module as 2-channel version	150
Fig. 180	2-wire connection.....	152
Fig. 181	Connection of externally supplied sensors	153
Fig. 182	2-, 3- and 4-wire connection at single-ended and differential inputs	154
Fig. 183	Common-mode voltage (Vcm).....	155
Fig. 184	Recommended operating voltage range.....	156
Fig. 185	Signal processing analog input.....	156
Fig. 186	Diagram signal delay (step response)	158
Fig. 187	Diagram signal delay (linear).....	158
Fig. 188	DiagMessages in the CoE	160
Fig. 189	Implementation of the DiagMessage system in the TwinCAT System Manager	161
Fig. 190	Device identifier consisting of name EL3204-0000 and revision -0016	171
Fig. 191	Scan the subordinate field by right-clicking on the EtherCAT device	172
Fig. 192	Configuration is identical	172
Fig. 193	Change dialog	172
Fig. 194	EEPROM Update	173
Fig. 195	Selecting the new ESI.....	173
Fig. 196	Display of EL3204 firmware version	174
Fig. 197	Firmware Update	175
Fig. 198	FPGA firmware version definition	177
Fig. 199	Context menu Properties	177
Fig. 200	Dialog Advanced Settings	178
Fig. 201	Multiple selection and firmware update	180
Fig. 202	Selecting the "Restore default parameters" PDO	182
Fig. 203	Entering a restore value in the Set Value dialog.....	182