



Documentation

EL37x2

2 channel Analog Input Terminals with oversampling

Version: 3.6
Date: 2018-10-30

BECKHOFF

1 Product overview analog input terminals with oversampling

[EL3702 \[► 14\]](#)

Two-channel analog input terminal, -10 V ... +10 V with oversampling

[EL3702-0015 \[► 14\]](#)

Two-channel analog input terminal, -150 mV ... +150 mV with oversampling

[EL3742 \[► 16\]](#)

Two-channel analog input terminal, 0 ... 20 mA with oversampling

Table of contents

1	Product overview analog input terminals with oversampling	3
2	Foreword	7
2.1	Notes on the documentation	7
2.2	Safety instructions	8
2.3	Documentation issue status	9
2.4	Version identification of EtherCAT devices	9
3	Product overview	14
3.1	EL3702 - Introduction	14
3.2	EL3702 - Technical data	15
3.3	EL3742 - Introduction	16
3.4	EL3742 - Technical data	17
3.5	Basic function principles	18
3.6	Sample programs	22
4	Basics communication	27
4.1	EtherCAT basics	27
4.2	EtherCAT cabling – wire-bound	27
4.3	General notes for setting the watchdog	28
4.4	EtherCAT State Machine	30
4.5	CoE - Interface: notes	32
4.6	Distributed Clock	33
5	Mounting and wiring	34
5.1	Instructions for ESD protection	34
5.2	Installation on mounting rails	34
5.3	Installation instructions for enhanced mechanical load capacity	37
5.4	Connection system	38
5.5	Installation positions	41
5.6	Positioning of passive Terminals	43
5.7	ATEX - Special conditions (extended temperature range)	44
5.8	ATEX Documentation	45
5.9	UL notice	45
5.10	Configuration of 0/4..20 mA differential inputs	47
5.11	EL37x2 - LEDs and pin assignment	51
5.11.1	EL37x2 - LEDs	51
5.11.2	EL37x2 - Pin assignment	52
6	Commissioning	55
6.1	TwinCAT Quick Start	55
6.1.1	TwinCAT 2	57
6.1.2	TwinCAT 3	67
6.2	TwinCAT Development Environment	79
6.2.1	Installation of the TwinCAT real-time driver	79
6.2.2	Notes regarding ESI device description	85
6.2.3	TwinCAT ESI Updater	89
6.2.4	Distinction between Online and Offline	89

6.2.5	OFFLINE configuration creation	90
6.2.6	ONLINE configuration creation	95
6.2.7	EtherCAT subscriber configuration	103
6.3	General Notes - EtherCAT Slave Application	112
6.4	Oversampling terminals and TwinCAT Scope	120
6.4.1	TwinCAT 3 procedure	121
6.4.2	TwinCAT 2 procedure	129
6.5	Process data and configuration	138
6.5.1	TwinCAT tree	138
6.5.2	Extent of process data in delivery state	140
6.5.3	Oversampling settings, distributed clocks (DC)	144
6.6	Notices on analog specifications	150
6.6.1	Full scale value (FSV)	151
6.6.2	Measuring error/ measurement deviation	151
6.6.3	Temperature coefficient tK [ppm/K]	152
6.6.4	Single-ended/differential typification	153
6.6.5	Common-mode voltage and reference ground (based on differential inputs)	158
6.6.6	Dielectric strength	158
6.6.7	Temporal aspects of analog/digital conversion	159
7	Appendix	162
7.1	EtherCAT AL Status Codes	162
7.2	Firmware information for EL37xx/EL47xx	162
7.3	Firmware compatibility	163
7.4	Firmware Update EL/ES/EM/EPxxxx	163
7.4.1	Device description ESI file/XML	164
7.4.2	Firmware explanation	167
7.4.3	Updating controller firmware *.efw	168
7.4.4	FPGA firmware *.rbf	170
7.4.5	Simultaneous updating of several EtherCAT devices	174
7.5	Support and Service	175

2 Foreword

2.1 Notes on the documentation

Intended audience

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

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

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.

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

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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

2.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

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.

2.3 Documentation issue status

Version	Comment
3.6	<ul style="list-style-type: none"> • EL3702-0015 added • Update structure • Update revision status
3.5	<ul style="list-style-type: none"> • Update chapter "Sample programs" • Update structure
3.4	<ul style="list-style-type: none"> • Update chapter "Technical data" • Chapter "ATEX - Special conditions" replaced with chapter "ATEX - Special conditions (extended temperature range)" • Addenda chapter "Instructions for ESD protection" • Update chapter "Notices on Analog specification" • Update revision status
3.3	<ul style="list-style-type: none"> • Update chapter "Product overview" • Addenda chapter "Notices on analog specifications"
3.2	<ul style="list-style-type: none"> • Update chapter "Notes on the documentation" • Update of Technical data • Addenda chapter "TwinCAT Quick Start" • Update revision status
3.1	<ul style="list-style-type: none"> • Section "Oversampling terminals and TwinCAT Scope" added
3.0	<ul style="list-style-type: none"> • Migration
2.0	<ul style="list-style-type: none"> • Update chapter "Technical data" • Addenda chapter "Installation instructions for enhanced mechanical load capacity" • Update structure • Update revision status
1.9	<ul style="list-style-type: none"> • Update structure • "Technical data" section updated
1.8	<ul style="list-style-type: none"> • Structural update • Technical data updated • Section "Configuration of 0/4..20 mA differential inputs" amended
1.7	<ul style="list-style-type: none"> • Technical notes added
1.6	<ul style="list-style-type: none"> • Technical notes example program amended
1.5	<ul style="list-style-type: none"> • Example program amended
1.4	<ul style="list-style-type: none"> • EL3742 amended
1.3	<ul style="list-style-type: none"> • Examples amended
1.2	<ul style="list-style-type: none"> • Technical data and safety instructions amended
1.1	<ul style="list-style-type: none"> • Technical data amended
1.0	<ul style="list-style-type: none"> • Technical data amended
0.1	<ul style="list-style-type: none"> • Provisional documentation for EL37x2

2.4 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

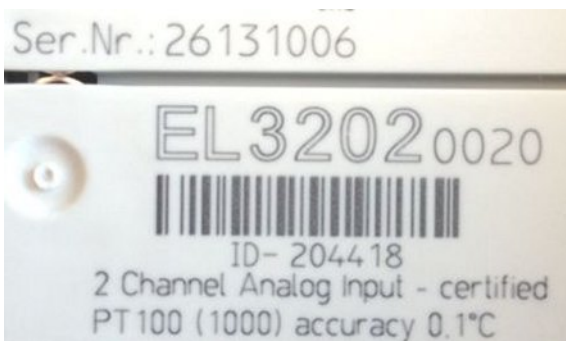


Fig. 4: EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418

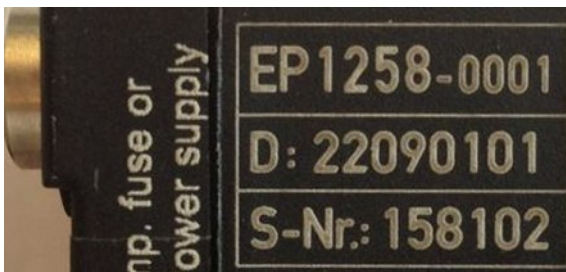


Fig. 5: EP1258-0001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102

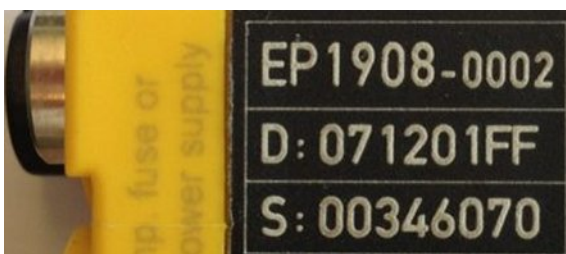


Fig. 6: EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070



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

3 Product overview

3.1 EL3702 - Introduction

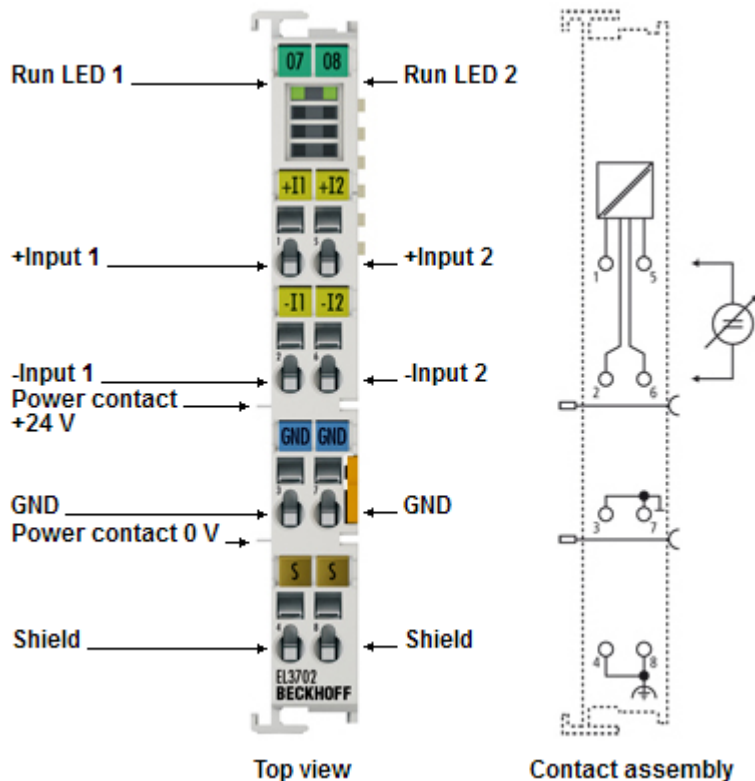


Fig. 9: EL3702

Two-channel analog input terminal, differential inputs, -10 V ... +10 V with oversampling

The EL3702 analog input terminal handles signals in the range from -10 V to +10 V. The voltage is digitized to a resolution of 16 bits, and is transmitted, electrically isolated, to the controller. The input channels of the EtherCAT Terminals have differential inputs and possess a common, internal ground potential.

The signals are sampled with a configurable, integer multiple (oversampling factor: n) of the bus cycle time (n microcycles per EtherCAT bus cycle). The EtherCAT Terminal generates a set of n process data per channel for each EtherCAT bus cycle, which is accumulated and transferred in the next bus cycle. The timebase of the terminal can be synchronized precisely with other EtherCAT devices via distributed clocks. This procedure enables the temporal resolution of the analog input signals to be increased to n times the bus cycle time.

Responses at equidistant intervals are made possible in conjunction with the EL4702/EL4732 (analog output terminal with oversampling) or any other distributed clocks-capable EtherCAT device.

The maximum sampling rate of the EL3702 is 100 ksp/s (10 μ s sampling) at a recommended input signal bandwidth of 0..30 kHz.

Information on the distributed clocks can be found in the separate System description.

The EL3702-0015 analog input terminal is also available for signals in the range from -150 mV to + 150 mV.

Quick links

- [EtherCAT basics](#)
- [Basic function principles \[► 18\]](#)
- [Process data and configuration \[► 138\]](#)

3.2 EL3702 - Technical data

Technical data	EL3702	EL3702-0015
Number of inputs	2	
Signal voltage	-10 V ... +10 V	-150 mV ... +150 mV
Oversampling factor	n = integer multiple of the EtherCAT cycle time, 1..100	
Distributed Clocks precision	< 100 ns	
Internal resistance	> 200 kΩ	
Common-mode voltage U _{CM}	max. 35 V	
Resolution	16 bit	
Conversion time	~ 10 μs	
Measuring error	< ± 0.3 % (at 25°C ... +55°C, based on the full-scale value, at < 10 Hz input signal) < ± 0.5 % (when the extended temperature range is used)	< ± 0.3 % (based on the full-scale value, at < 10 Hz input signal)
Input signal bandwidth	0 ... 30 kHz (recommended)	
Electrical isolation	500 V (E-bus/field voltage)	
Supply voltage for electronics	via the E-bus	
Current consumption via E-bus	typ. 200 mA	
Bit width in process image	Input: n x 2 x 16 bit data, 2 x 16 bit CycleCounter if applicable, 4 bytes StartNextLatchTime if applicable	
Configuration	via TwinCAT System Manager	
Weight	approx. 60 g	
Permissible ambient temperature range during operation	-25°C ... +60°C (extended temperature range)	0°C ... +55°C
Permissible ambient temperature range during storage	-40°C ... +85°C	-25°C ... +85°C
Permissible relative humidity	95 %, no condensation	
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)	
Mounting [▶ 34]	on 35 mm mounting rail conforms to EN 60715	
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [▶ 37] for enhanced mechanical load capacity	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	variable	
Approval	CE ATEX [▶ 44] IECEX cULus [▶ 45]	CE

3.3 EL3742 - Introduction

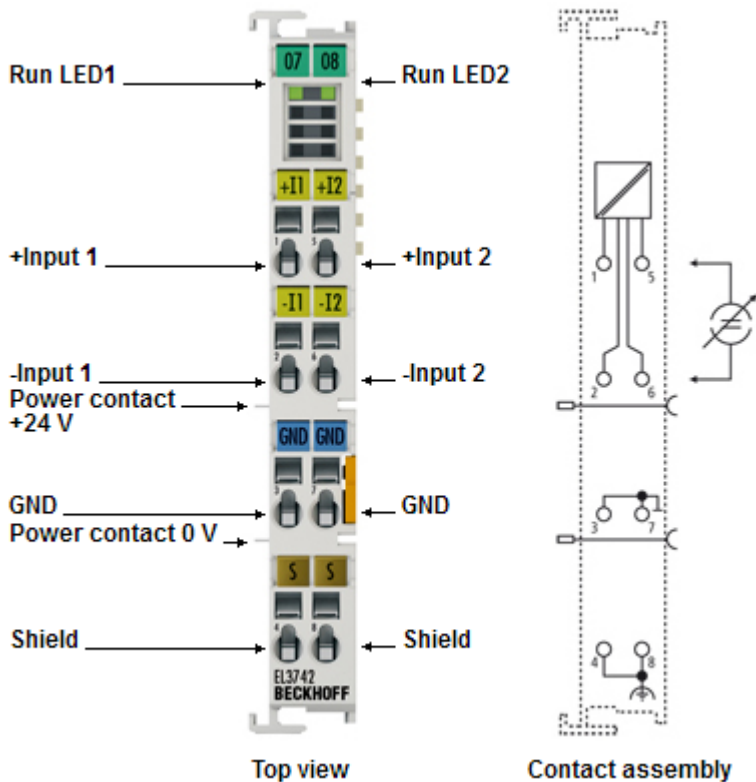


Fig. 10: EL3742

Two-channel analog input terminal, differential inputs, 0 ... 20 mA with oversampling

The EL3742 analog input terminal handles signals in the range from 0 to 20 mA. The voltage is digitized to a resolution of 16 bits, and is transmitted, electrically isolated, to the controller. The input channels of the EtherCAT Terminals have differential inputs and possess a common, internal ground potential.

The signals are sampled with a configurable, integer multiple (oversampling factor: n) of the bus cycle time (n microcycles per EtherCAT bus cycle). The EtherCAT Terminal generates a set of n process data per channel for each EtherCAT bus cycle, which is accumulated and transferred in the next bus cycle. The timebase of the terminal can be synchronized precisely with other EtherCAT devices via distributed clocks. This procedure enables the temporal resolution of the analog input signals to be increased to n times the bus cycle time.

Responses at equidistant intervals are made possible in conjunction with the EL4702/EL4732 (analog output terminal with oversampling) or any other distributed clocks-capable EtherCAT device.

The maximum sampling rate of the EL3742 is 100 kps (10 μ s sampling) at a recommended input signal bandwidth of 0..30 kHz.

Information on the distributed clocks can be found in the separate System description.

Quick links

- [EtherCAT basics](#)
- [Basic function principles \[►_18\]](#)
- [Process data and configuration \[►_138\]](#)

3.4 EL3742 - Technical data

Technical data	EL3742
Number of inputs	2
Signal voltage	0 mA...20 mA
Oversampling factor	n = integer multiple of the EtherCAT cycle time, 1..100
Distributed Clocks precision	< 100 ns
Internal resistance	85 Ω typ. + diode voltage
Resolution	16 bit
Conversion time	~ 10 μs
Input signal bandwidth	0 ... 30 kHz (recommended)
Measuring error	< ± 0.3 % (at 25 °C ... +55 °C, based on the full-scale value, at < 10 Hz input signal) < ± 0.5 % (when the extended temperature range is used)
Electrical isolation	500 V (E-bus/field voltage)
Supply voltage for electronics	via the E-bus
Current consumption via E-bus	typ. 200 mA
Bit width in process image	Input: n x 2 x 16 bit data, 2 x 16 bit CycleCounter if applicable, 4 bytes StartNextLatchTime if applicable
Configuration	via TwinCAT System Manager
Weight	approx. 60 g
Permissible ambient temperature range during operation	-25 °C ... +60 °C (extended temperature range)
Permissible ambient temperature range during storage	-40 °C ... +85 °C
Permissible relative humidity	95 %, no condensation
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)
Mounting [▶ 34]	on 35 mm mounting rail conforms to EN 60715
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [▶ 37] for enhanced mechanical load capacity
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE ATEX [▶ 44] cULus [▶ 45]

3.5 Basic function principles

The analog EL3702/ EL3742 input terminal enables measurement of two voltages/currents, which can be displayed with a resolution of 16 bit (65535 steps).

The oversampling feature enables the terminal to sample analog input values several times during each bus cycle on each channel.

Oversampling

A conventional analog input terminal samples one analog input value ("sample") during each bus cycle and transfers it to the higher-level control system during the next fieldbus cycle. The EL37x2 samples the voltage several times between two fieldbus communication cycles configurable and at equidistant intervals. A 16-bit packet of x analog measured values is transferred to the higher-level control system during the next fieldbus communication cycle. This procedure is referred to as oversampling.

Distributed Clock

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

The distributed clock represents a local clock in the 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 $< 100\text{ ns}$.

The EL37x2 only offers 32-bit support.

Sample:

The fieldbus/EtherCAT master is operated with a cycle time of 1 ms to match the higher-level PLC cycle time of 1 ms , for example. This means that every 1 ms an EtherCAT frame is sent to collect the process data from the EL37x2. The local terminal clock therefore triggers an interrupt in the ESC every 1 ms (1 kHz), in order to make the process data available in time for collection by the EtherCAT frame. This first interrupt is called SYNC1.

The EL37x2 may be set to oversampling $n = 10$ in the TwinCAT System Manager. This causes the ESC to generate a second interrupt in the terminal with an n -times higher frequency, in this case 10 kHz or $100\ \mu\text{s}$ period. This interrupt is called SYNC0. At each SYNC0 signal the analog-to-digital converter (ADC) starts a data sampling event, and the sampled analog values are sequentially stored in a buffer.

● Determining the input voltages/input currents

I Both input voltages / input currents (channels 1 and 2) are always sampled simultaneously. This is ensured by the ADC type that is triggered by the SYNC0 pulse. No other operation mode is possible.

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

The maximum oversampling factor depends on the memory size of the used ESC and in the KKYY0000 version of the EL37x2, it is $n = 100$.

● **Maximum sampling frequency**

i A smaller period than 10 μs is not permitted for the EL37x2. The maximum sampling frequency for the EL37x2 is therefore 100 kSps (samples per second).

Regarding the calculation of SYNC0 from the SYNC1 pulse based on manual specification of an oversampling factor, please note that for SYNC0 only integer values are calculated at nanosecond intervals.

Sample: 187,500 μs is permitted, 333.3 is not.

Sample:

For SYNC1 = 1 ms oversampling factors such as 1,2,5 or 100 are permitted, but not 3! If implausible values are used the terminal will reach the OP state but will not supply any process data. This may result in a working counter error.

The 16-bit measured values accumulated in the buffer are sent as a packet to the higher-level control system. For two channels and $n = 100$, $2 \times 2 \times 100 = 400$ bytes of process data are transferred during each EtherCAT cycle.

Time-related cooperation with other terminals

ADC data sampling in the EL37x2 is triggered by an interrupt that is generated by the local clock in the terminal. All local clocks in the supporting EtherCAT slaves are synchronized. This enables EtherCAT slaves (here: terminals) to sample measured values simultaneously (simultaneous interrupt generation), *independent of the distance between them*. This simultaneity is within the distributed clock precision range of < 100 ns.

Sample:

Matching between two EL37x2:

The EtherCAT master, e.g. Beckhoff TwinCAT, configures both EL37x2 such that their SYNC1 signals occur at the same time. Assumption: The EtherCAT bus cycle time is 500 μs . SYNC1 is therefore triggered every 500 μs in all EL37x2. If both terminals operate with a corresponding oversampling factor (e.g. 20), the SYNC0 pulse correlating to SYNC1 will occur simultaneously in all EL37x2, in this example every 25 μs . One application option would be "distributed" oscilloscope with a sampling frequency of 40 kSps, for example.

If the EL37x2 use different oversampling factors, their SYNC0 pulses no longer occur simultaneously. The higher-level SYNC1 pulse is retained.

If a value is entered under "Shift time (μs)" in the TwinCAT System Manager (DC tab, Advanced Settings) for the SYNC0 pulse in an EL37x2, the EL37x2 manipulated in this way will start output sooner or later, according to the set value. This can be useful in cases where a wanted signal has to be sampled at a higher frequency than the 100 kSps permitted for the EL37x2, and several interconnected EL37x2 are therefore used.

Sample:

For sampling a wanted signal at 200 kSps two EL37x2 are used side-by-side and subjected to the same wanted signal (e.g. channel 1 in both terminals). Assume a bus cycle of 1 ms, an oversampling factor in both EL37x2 of $n = 100$, and therefore maximum speed at 10 μs interval or 100 kSps. In the second EL37x2 an *additional* Shift Time of 5 μs is entered manually for this terminal in the System Manager (DC tab, Advanced Settings, SYNC0, User Defined). This means all SYNC0 pulses for this EL37x2 will occur 5 μs later than for the first EL37x2.

● **Synchronization and provision of process data**

i Since the SYNC1 pulse is derived from the SYNC0 pulse, each SYNC1 pulse of the second EL37x2 will occur 5 μs after the SYNC1 pulse of the first EL37x2. Please note that this may influence the timing of the process data allocation for the EtherCAT frame, since this is controlled by the SYNC1 pulse.

Each terminal now transfers a 400 byte process data packet to the higher-level control system. If these two data sets (which are offset by 5 μs) with a length of 1 ms each are sorted in the right chronological order in the control system, the wanted signal is transparently sampled at 200 kSps.

The application of these functions using the Beckhoff TwinCAT System Manager is described in section [Process data and configuration \[▶ 138\]](#).

● SYNC0 and SYNC1 pulse with several EtherCAT slaves

i This approach of matching the SYNC0 and SYNC1 pulses of several EtherCAT slaves is not limited to EL37x2. All EtherCAT slaves supporting the distributed clock function can be correlated relatively freely in this way.

NOTE

Attention! Risk of device damage!

The above notes and information should be used advisedly. The EtherCAT master automatically allocates SYNC0 and SYNC1 settings that support reliable and timely process data acquisition. User intervention at this point may lead to undesired behavior. If these settings are changed in the System Manager, no plausibility checks are carried out on the software side. Correct function of the terminal with all conceivable setting options cannot be guaranteed.

Timestamp of the process data

The EL37xx can provide a "timestamp" for each process data block, if required. This process record can be activated as *StartTimeNextLatch*, a 32-bit value, by activating 0x1B10 in the *Process Data* tab, see also [Process Data \[▶ 138\]](#) page.

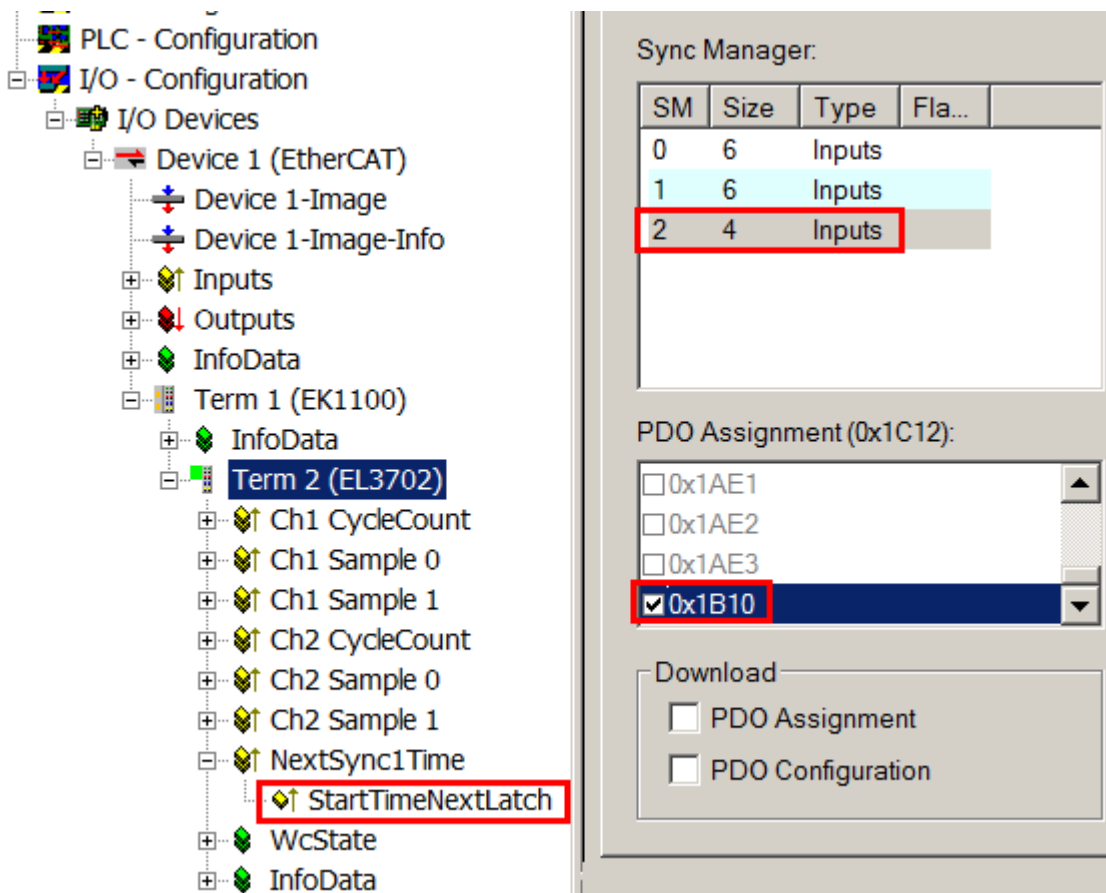


Fig. 11: Optional process record StartTimeNextLatch

As the name suggests, the data block consisting of sample value+timestamp, which is transferred in each cycle, is not related. The relationship is shown in Fig. *Temporal relationship between SYNC signals and SyncManager interrupt*. To explain in more detail:

- the example is based on an oversampling factor of 5
- The SYNC0 signal in the terminal triggers the AD conversion and fills the internal buffer with 5 measured values (A).
- SYNC1, which triggers the filled buffer to be made available as process record and at the same time fetches the *StartTimeNextLatch* from the local distributed clock (B), runs synchronous with the cycle time.
- The data block is linked with the next but one LatchTime.
- The next EtherCAT cycle fetches this data (C).

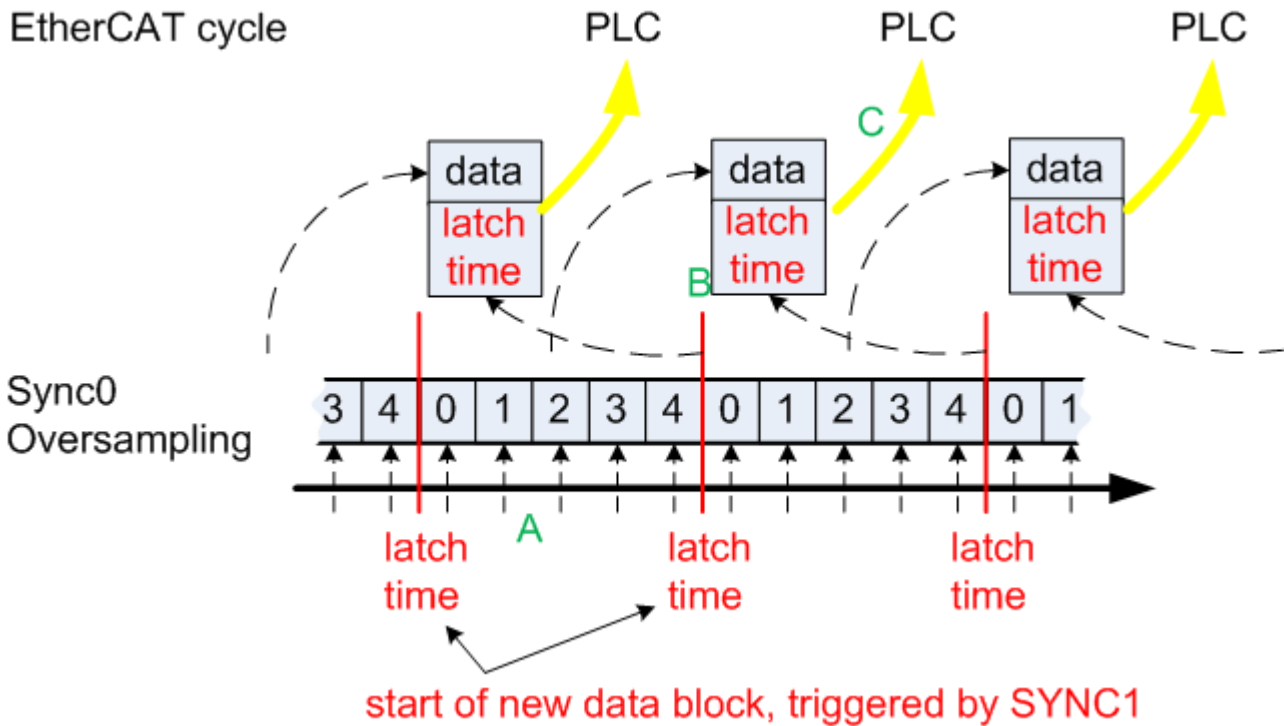


Fig. 12: Temporal relationship between SYNC signals and SyncManager interrupt

Process data

Analog values are represented as follows:

Input signal	Value	
EL3702	Decimal	Hexadecimal
-10 V	-32767	0x8001
+10 V	+32767	0x7FFF

Input signal	Value	
EL3742	Decimal	Hexadecimal
0 mA	0	0x0000
20 mA	+32767	0x7FFF

The terminal is adjusted during production. No further user intervention is required.

Input characteristics

The input circuit of this terminal is optimized for higher-frequency signals up to around 30 kHz, i.e. the recommended bandwidth of the wanted signal is 0 Hz to 30 kHz in the range -10 V to +10 V or 0 mA to 20 mA. In this frequency range the typical measuring accuracy is as follows:

< 10 Hz < 0.3 % of full-scale value

< 10 kHz < 1 % of full-scale value

< 30 kHz < 4 % of full-scale value

For wanted signals with higher frequencies the signal transducer must have adequately low impedance, in order to prevent amplitude variations (e.g. attenuation in association with simple signal generators) leading to incorrect measurements.

Interference from equipment

This fast analog EtherCAT Terminal may pick up high-frequency superimposed interference signals from other equipment (e.g. proportional valves, stepper motor or DC motor output stages). In order to ensure trouble-free operation, we recommend using separate power supply units for the terminals and the equipment causing interference. The cables should be screened.

3.6 Sample programs

i Using the example programs

This document contains sample applications of our products for certain areas of application. The application notices provided here are based on typical features of our products and only serve as samples. The notices 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.

Diagnostics and time-stamping of analog input data

 Download (<https://infosys.beckhoff.com/content/1033/el37x2/Resources/zip/2445351051.zip>)

In this example the input data of an EL3702 will be checked for validity and processed:

- 1 ms cycle time, 10-fold oversampling, 2 channels
- WC, State, EtherCAT Master DevState and CycleCounter are checked cyclically; the input data is only passed on if it is valid
- Starting from the time stamp delivered with sample 0, all other samples will be given 64-bit DC time stamps
- Default values will be passed on if the input data is invalid; time stamping continues
- The data from each cycle is placed in a FIFO buffer so that a superordinated process, e.g. an evaluation, can take place.

Connection diagram:

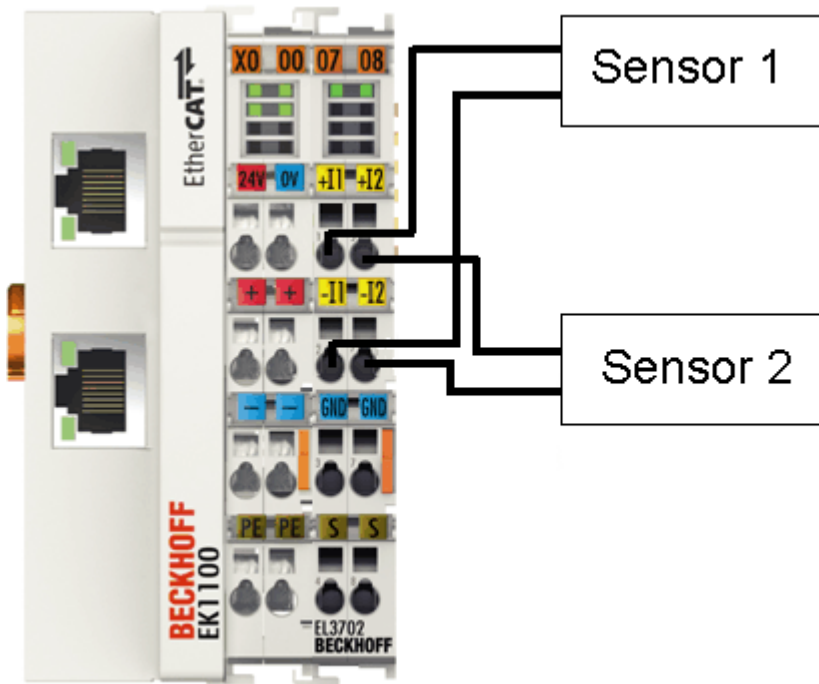


Fig. 13: Connection for sample program

Starting the example program

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

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

- TwinCAT master PC with Windows XP Professional SP 3, TwinCAT version 2.10 (Build 1330) and INTEL PRO/100 VE Ethernet adapter
- Beckhoff EK1100 EtherCAT coupler, EL3702 and EL9011 terminals
- 2 x optical proximity limit switch 0 - 10 V with two-wire technology

Procedure for starting the program

- After clicking the Download button, save the zip file locally on your hard disk, and unzip the *.TSM (configuration) and the *.PRO (PLC program) files into a temporary working folder.
- The *.pro file can be opened by double click or by the TwinCAT PLC Control application with menu selection "File/ Open". The *.tsm file is provided for the TwinCAT System Manager (to review or overtake configurations).
- Connect the hardware in accordance with fig. [Connection for sample program \[► 22\]](#) and connect the Ethernet adapter of your PC to the EtherCAT coupler (further information on this can be found in the corresponding coupler manuals)
- Select the local Ethernet adapter (with real-time driver, if applicable) under System configuration, I/O configuration, I/O devices, Device (EtherCAT); then on the "Adapter" tab choose "Search...", select the appropriate adapter and confirm (see Fig. [Searching the Ethernet adapter + Selection and confirmation of the Ethernet adapter](#)).

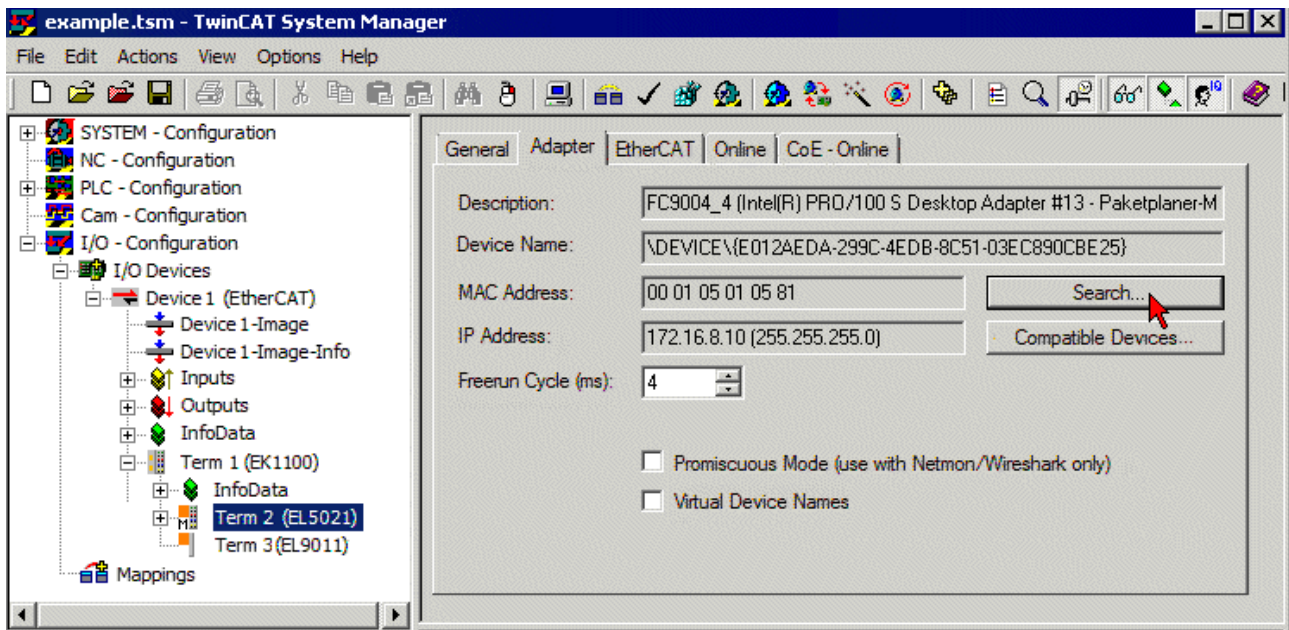


Fig. 14: Searching the Ethernet adapter

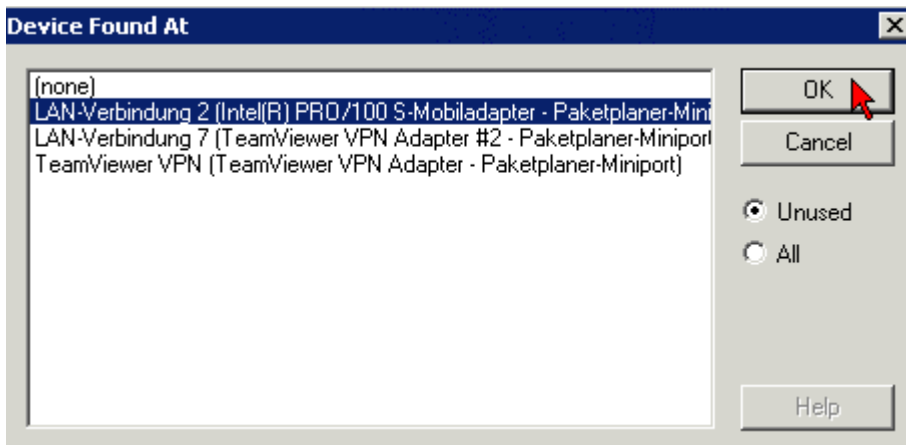


Fig. 15: Selection and confirmation of the Ethernet adapter

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



Fig. 16: Activation of the configuration

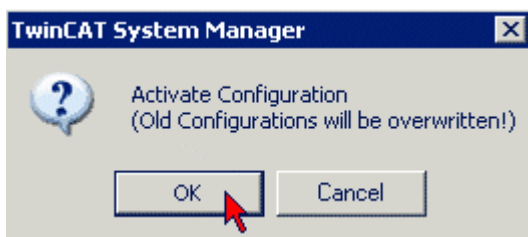


Fig. 17: Confirming the activation of the configuration

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

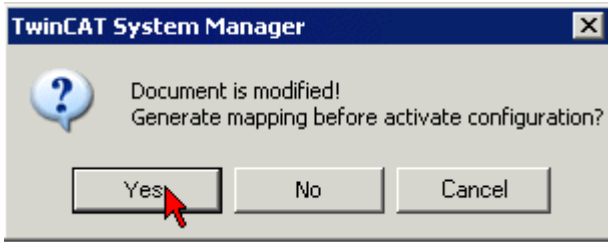


Fig. 18: Generating variable mapping



Fig. 19: Restarting TwinCAT in RUN mode

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

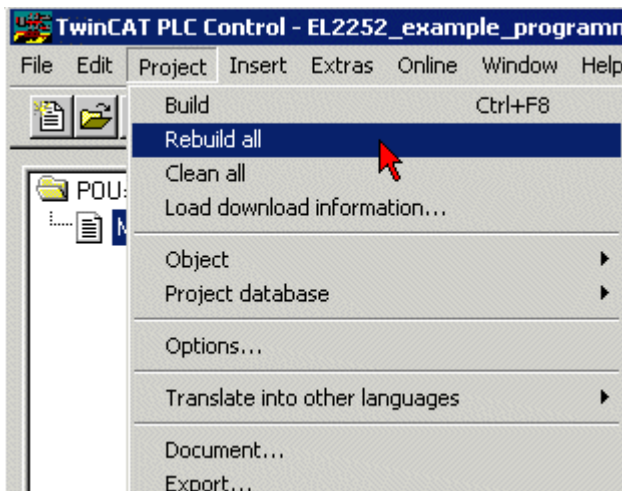


Fig. 20: Compile project

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

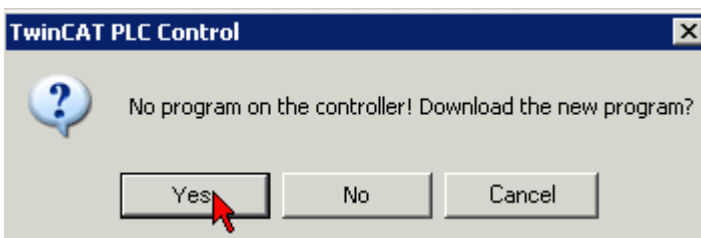


Fig. 21: Confirming program start

Working with DC times in the controller

From the perspective of the controller the distributed clock time has the following characteristics:

- Unit *1 ns*

- Universal zero point *1.1.2000 00:00*, i.e. for variable evaluations an offset of 2000 years has to be added
- Scope up to *64 bit* (sufficient for 584 years). However, some EtherCAT slaves only support a 32 bit scope, i.e. the register overflows locally after approx. 4.2 seconds and starts again at 0.

The following 3 data types are recommended for handling DC times

- **T_DCTIME** from TcEtherCAT.lib
This is based on T_ULARGE_INTEGER and is therefore unsigned. It can be used for linking with suitable hardware variables
- **T_ULARGE_INTEGER** from TcUtilities.lib
Unsigned 64-bit data type
- **T_LARGE_INTEGER** from TcUtilities.lib
Signed 64-bit data type, negative numbers are represented in two's complement notation (underflow below 0 --> 0xFFFF FFFF FFFF FFFF etc.)
TcUtilities.lib (section INT64) provides numerous relevant functions. Of particular significance are the cast functions LARGE_TO_ULARGE and vice versa.
This type should be used when working with time differences that may be negative.
If TwinCAT is used for external synchronization, negative times will inevitably occur in the offset values.

● 64- vs. 32-bit representation



Some EtherCAT slaves can only handle 32 bit values for representing the DC time or handle it as a process data. In order to prevent problems caused by overflow (every 4.2 seconds), we strongly recommend using 64-bit times in the controller.

- 32-bit times supplied to the PLC must be complemented with the current High part
- In this case only the Low part (lower 32 bit) should be transferred to the hardware

This sample project



(<https://infosys.beckhoff.com/content/1033/el37x2/Resources/zip/2469155979.zip>) contains a function block that cyclically adds the high part to a 32-bit DC time to make 64 bits.

4 Basics communication

4.1 EtherCAT basics

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

4.2 EtherCAT cabling – wire-bound

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

Cables and connectors

For connecting EtherCAT devices only Ethernet connections (cables + plugs) that meet the requirements of at least category 5 (Cat5) according to EN 50173 or ISO/IEC 11801 should be used. EtherCAT uses 4 wires for signal transfer.

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

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

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



Recommended cables

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

E-Bus supply

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

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

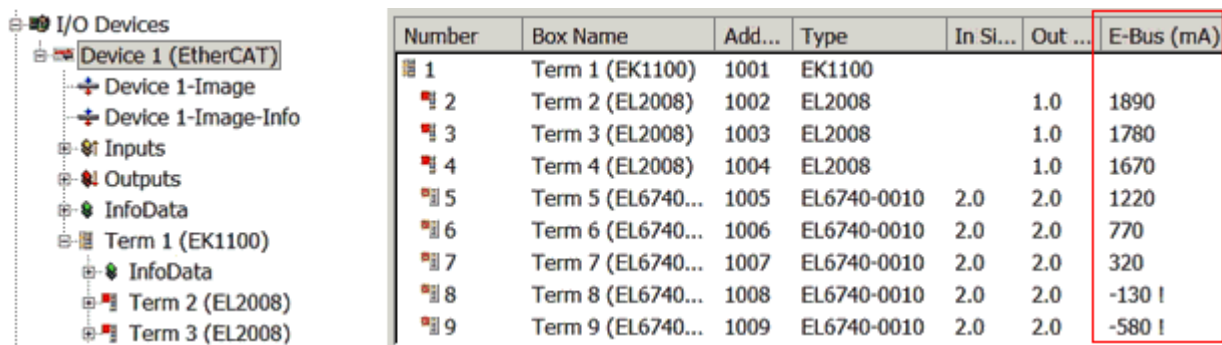


Fig. 22: System manager current calculation

NOTE

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

4.3 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.

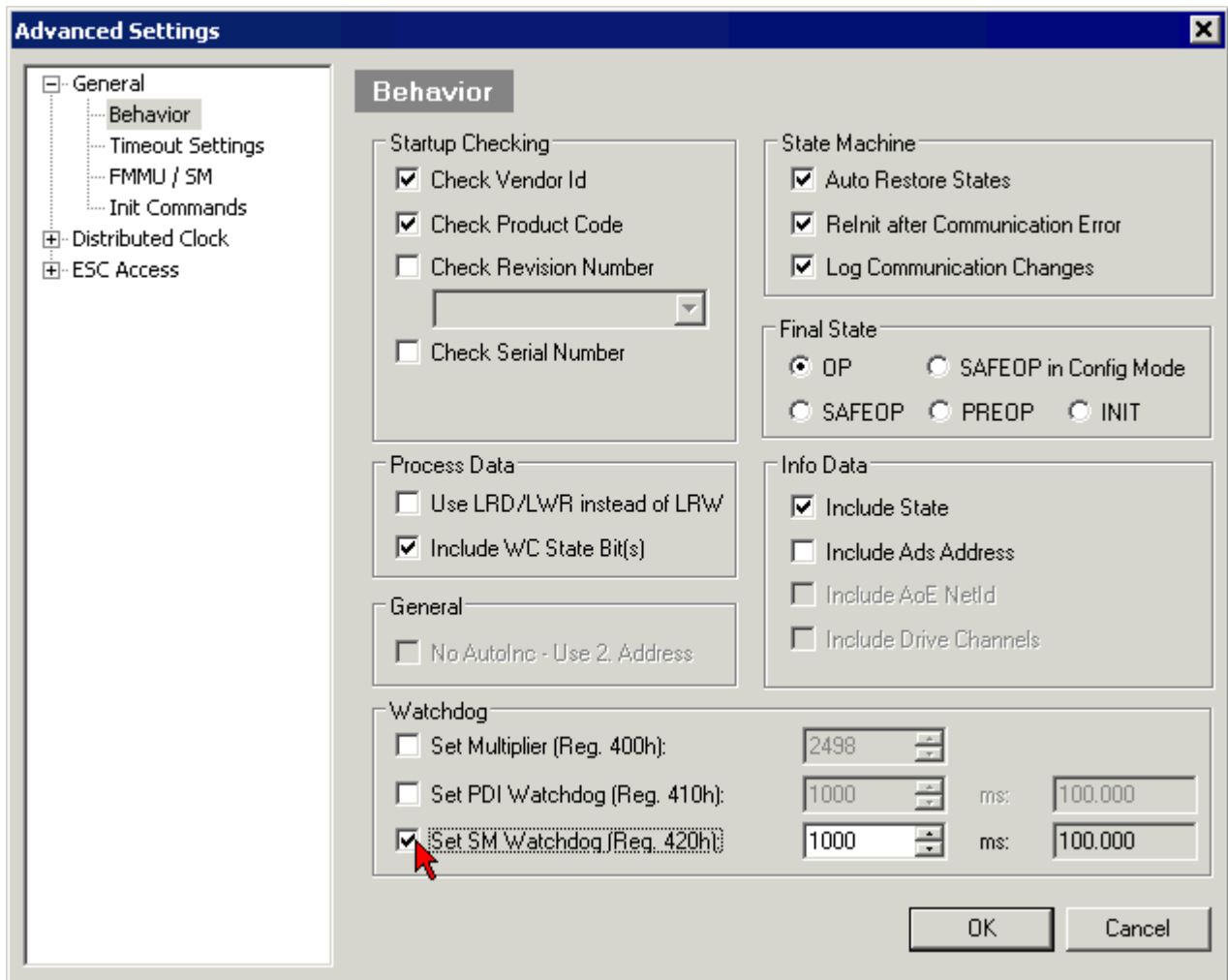


Fig. 23: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

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:

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

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.

4.4 EtherCAT State Machine

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

A distinction is made between the following states:

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

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

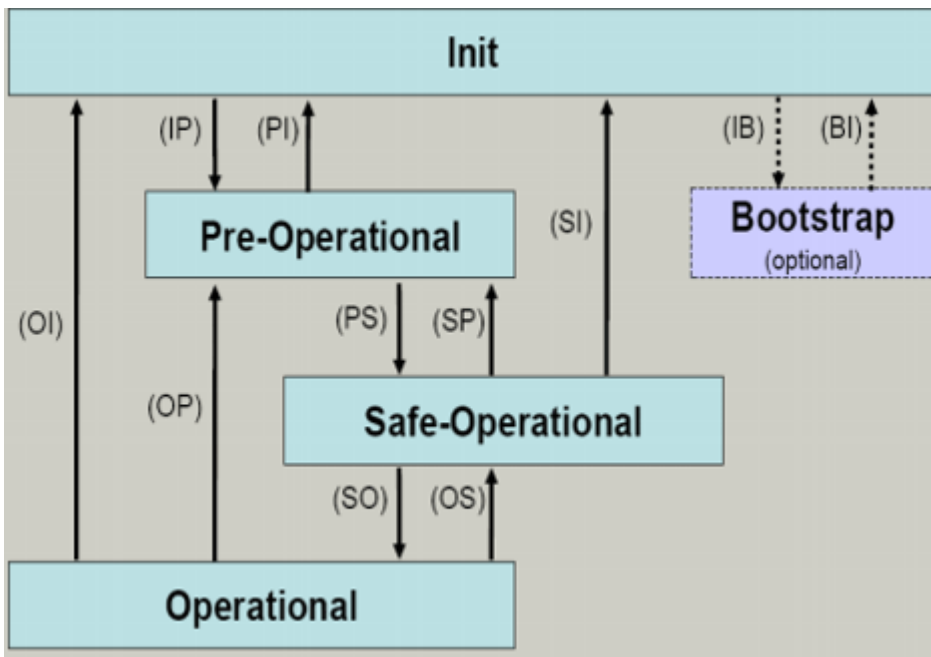


Fig. 24: States of the EtherCAT State Machine

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 [P_28]` 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.

4.5 CoE - Interface: notes

This device has no CoE.

Detailed information on the CoE interface can be found in the [EtherCAT system documentation](#) on the Beckhoff website.

4.6 Distributed Clock

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

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

For detailed information please refer to the [EtherCAT system description](#).

5 Mounting and wiring

5.1 Instructions for ESD protection

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.

a) Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.).

b) Surroundings (working place, packaging and personnel) should be grounded probably, when handling with the devices.

c) 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.

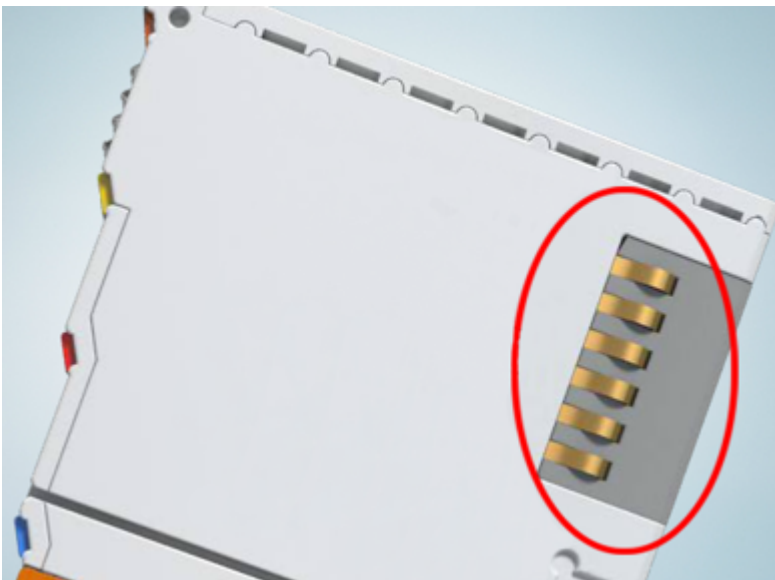


Fig. 25: Spring contacts of the Beckhoff I/O components

5.2 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

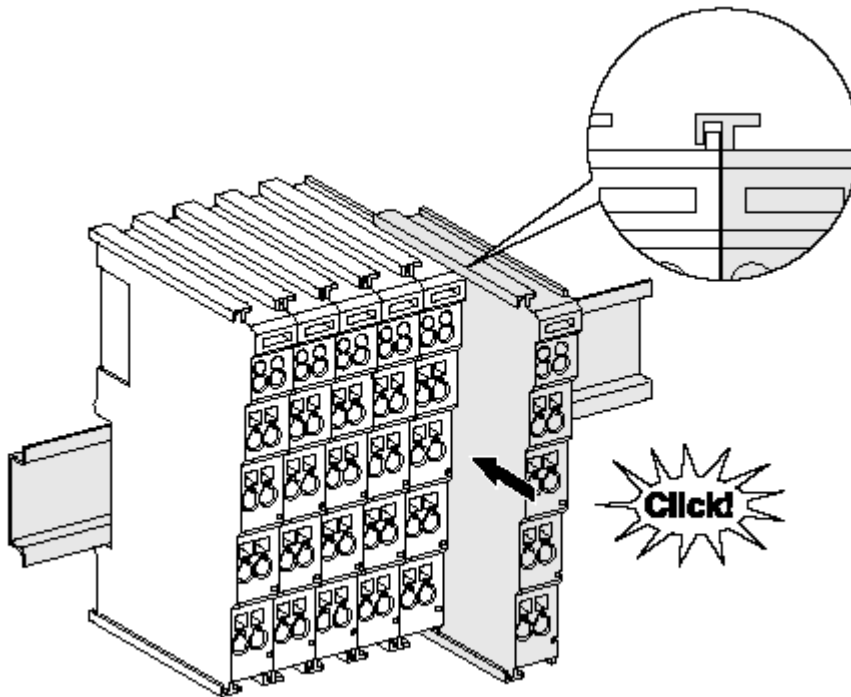


Fig. 26: Attaching on mounting rail

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

1. First attach the fieldbus coupler to the mounting rail.
2. The bus terminals are now attached on the right-hand side of the fieldbus coupler. Join the components with tongue and groove and push the terminals against the mounting rail, until the lock clicks onto the mounting rail.

If the terminals are clipped onto the mounting rail first and then pushed together without tongue and groove, the connection will not be operational! When correctly assembled, no significant gap should be visible between the housings.

● Fixing of mounting rails

i 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

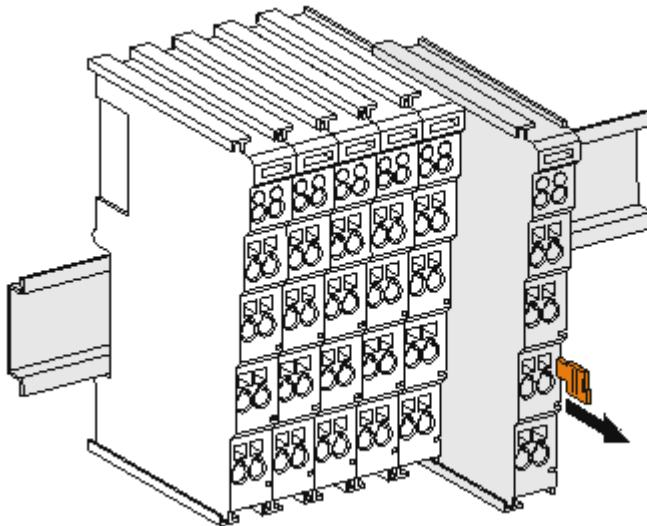


Fig. 27: Disassembling of terminal

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.

● Power Contacts

i 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.

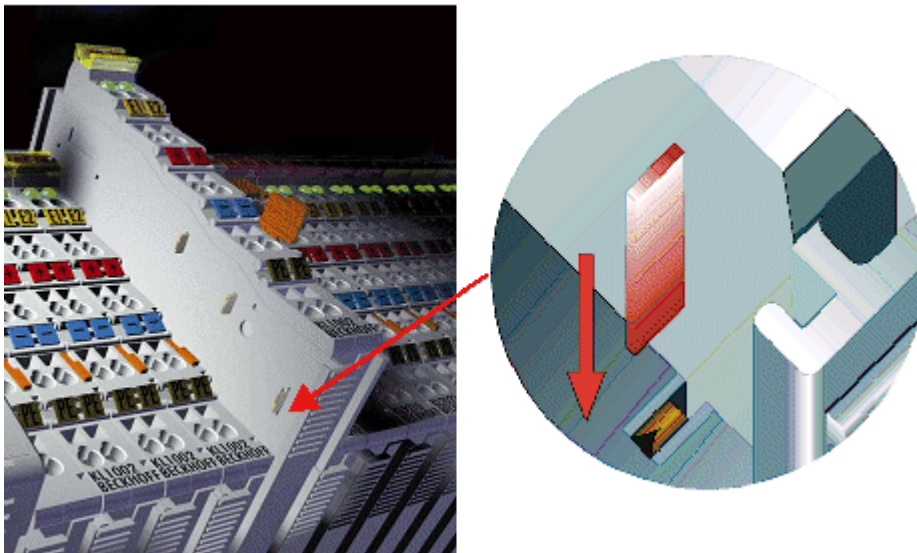


Fig. 28: Power contact on left side

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!

5.3 Installation instructions for enhanced mechanical load capacity

⚠ WARNING

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

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

Additional checks

The terminals have undergone the following additional tests:

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

Additional installation instructions

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

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

5.4 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. 29: 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

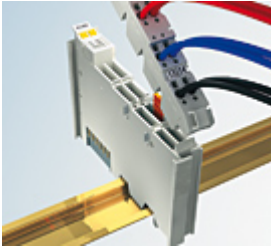


Fig. 30: Pluggable wiring

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. 31: 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.

● Wiring HD Terminals

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

Ultrasonically "bonded" (ultrasonically welded) conductors

● Ultrasonically "bonded" conductors

i 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 [[▶_40](#)] below!

Wiring

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

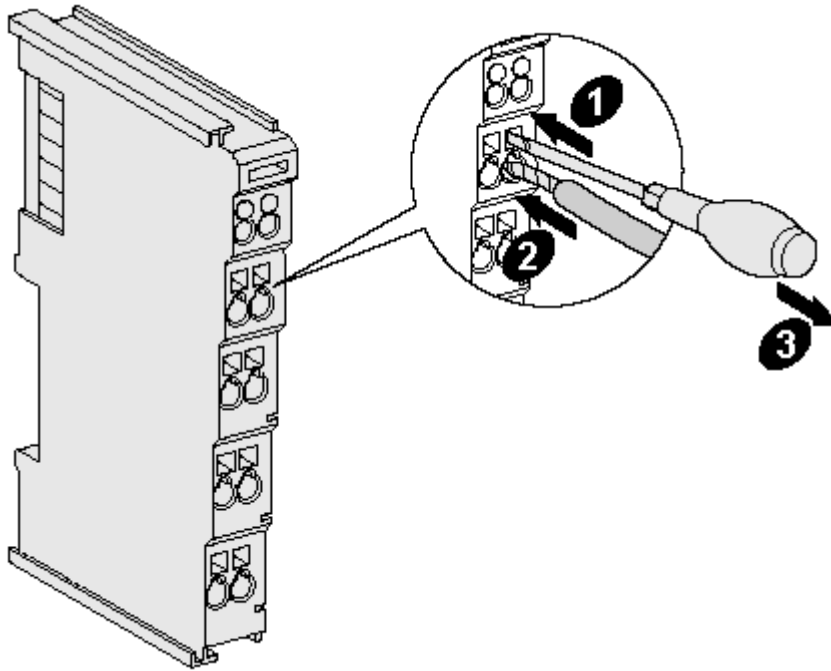


Fig. 32: Mounting a cable on a terminal connection

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 [► 39]!)
Wire stripping length	8 ... 9 mm

Shielding



Shielding

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

5.5 Installation positions

NOTE

Constraints regarding installation position and operating temperature range

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

Optimum installation position (standard)

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

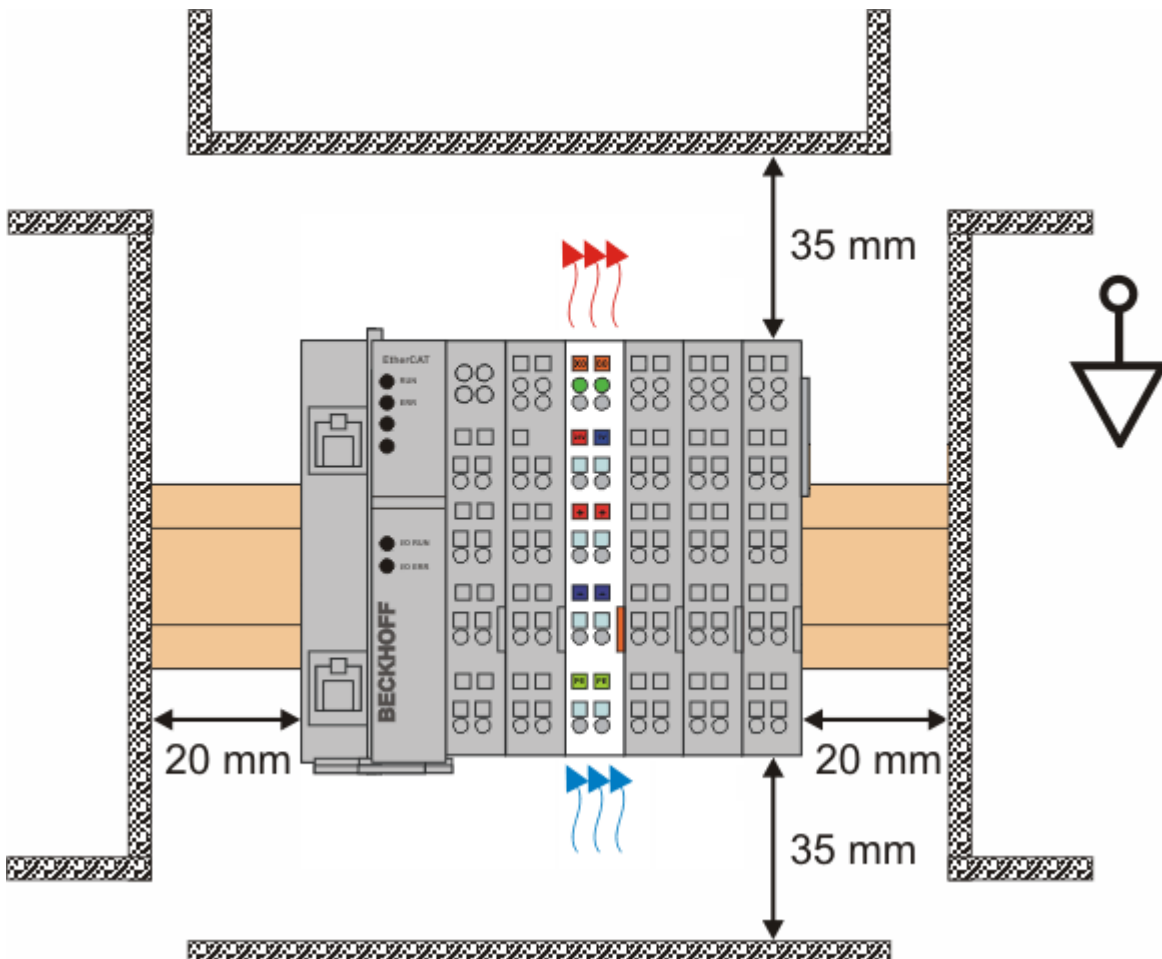


Fig. 33: Recommended distances for standard installation position

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

Other installation positions

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

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

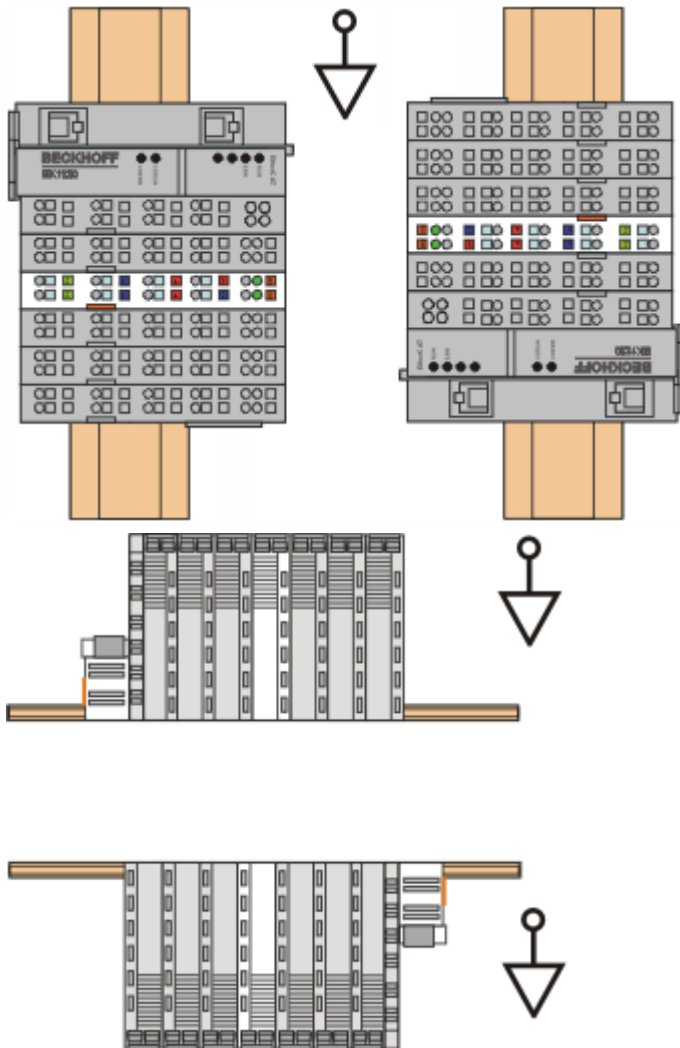


Fig. 34: Other installation positions

5.6 Positioning of passive Terminals

i **Hint for positioning of passive terminals in the bus terminal block**

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

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

Examples for positioning of passive terminals (highlighted)

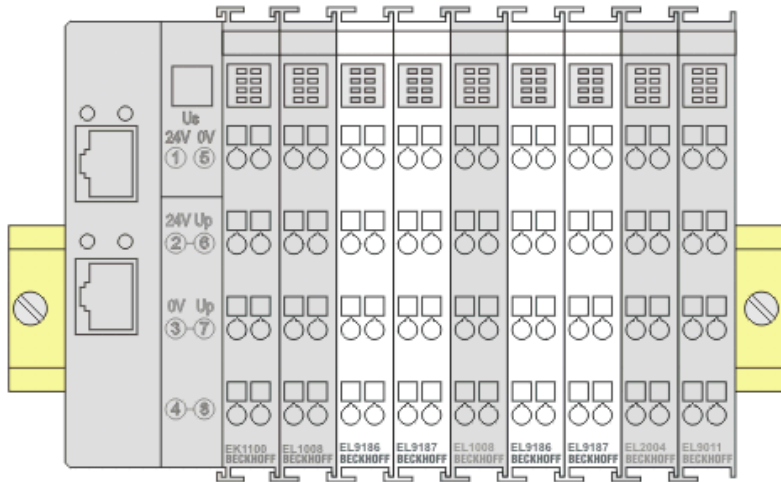


Fig. 35: Correct positioning

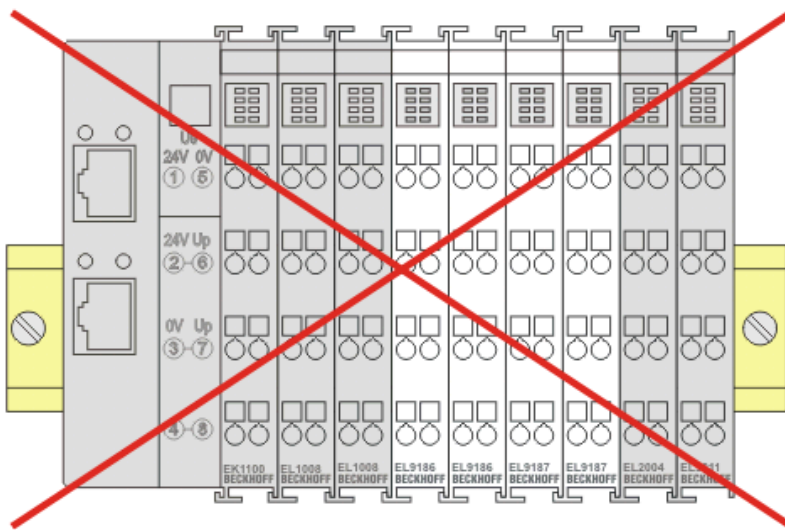


Fig. 36: Incorrect positioning

5.7 ATEX - Special conditions (extended temperature range)

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components with extended temperature range (ET) in potentially explosive areas (directive 94/9/EU)!

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60529! The environmental conditions during use are thereby to be taken into account!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of -25 to 60°C for the use of Beckhoff fieldbus components with extended temperature range (ET) in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

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

- EN 60079-0:2012+A11:2013
- EN 60079-15:2010

Marking

The Beckhoff fieldbus components with extended temperature range (ET) certified for potentially explosive areas bear the following marking:



II 3G KEMA 10ATEX0075 X Ex nA IIC T4 Gc Ta: -25 ... 60°C

or



II 3G KEMA 10ATEX0075 X Ex nC IIC T4 Gc Ta: -25 ... 60°C

5.8 ATEX Documentation






Notes about operation of the Beckhoff terminal systems in potentially explosive areas (ATEX)

Pay also attention to the continuative documentation

Notes about operation of the Beckhoff terminal systems in potentially explosive areas (ATEX)

that is available in the download area of the Beckhoff homepage <http://www.beckhoff.com!>

5.9 UL notice

	<p>Application Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.</p>
	<p>Examination For cULus examination, the Beckhoff I/O System has only been investigated for risk of fire and electrical shock (in accordance with UL508 and CSA C22.2 No. 142).</p>
	<p>For devices with Ethernet connectors Not for connection to telecommunication circuits.</p>

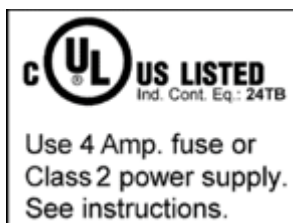
Basic principles

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

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



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



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

Application

If terminals certified *with restrictions* are used, then the current consumption at 24 V_{DC} must be limited accordingly by means of supply

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

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

5.10 Configuration of 0/4..20 mA differential inputs

This section describes the 0/4..20 mA differential inputs for terminal series EL301x, EL302x, EL311x, EL312x and terminals EL3174, EL3612, EL3742 and EL3751.

For the single-ended 20 mA inputs the terminal series EL304x, EL305x, EL314x, EL315x, EL317x, EL318x and EL375x they only apply with regard to technical transferability and also for devices whose analogue input channels have a common related ground potential (and therefore the channels are not to each other and/or not to power supply electrically isolated). Herewith an example for an electrically isolated device is the terminal EL3174-0002.

Technical background

The internal input electronics of the terminals referred to above have the following characteristic (see [Fig. \[▶ 47\]](#) *Internal connection diagram for 0/4..20 mA inputs*):

- Differential current measurement, i.e. concrete potential reference is primarily not required. The system limit applies is the individual terminal EL30xx/EL31xx.
- Current measurement via a 33 Ω shunt per channel, resulting in a maximum voltage drop of 660 mV via the shunt
- Internal resistor configuration with GND point (A) central to the shunt
The configuration of the resistors is symmetric, such that the potential of (A) is central relative to the voltage drop via the shunt.
- All channels within the terminal have this GND_{int} potential in common.
- the common GND_{int} potential (A)
 - is connected for 1 and 2 channel terminals to a terminal point and not with GND_{PC} (power contact).
 - is connected for 4 channel terminals with GND_{PC}
- The center point of the voltage drop over the 33 Ω shunt is referred to common mode point (CMP). According to the technical product data, the maximum permitted U_{CM} voltage (common mode) refers to the potential between the CMP of a channel and the internal GND or the potential between the CMP of 2 channels within a terminal.

It must not exceed the specified limit (typically ±10 or ±35 V).

Accordingly, for multi-channel measurements U_{CM} specifications must be followed.

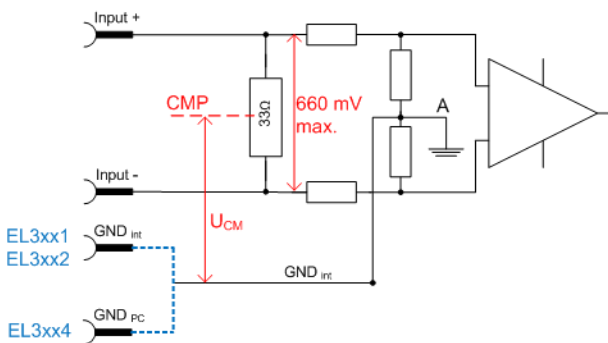


Fig. 37: Internal connection diagram 0/4...20 mA inputs

The block diagram for a 2 channel terminal shows the linked GND points within the terminal ([Fig. \[▶ 48\]](#) *Internal connection for 0/4..20 mA inputs of a EL3xx2*):

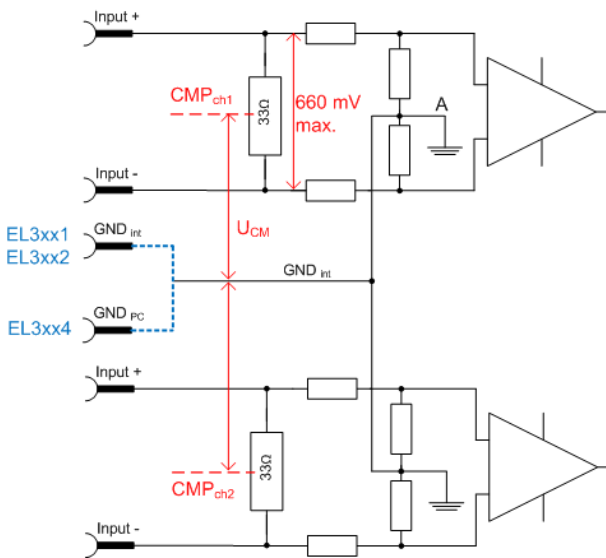


Fig. 38: Internal connection diagram for 0/4..20 mA inputs of a EL3xx2

For all channels within the terminal U_{CM-max} must not be exceeded.

i U_{CM} for 0/4..20 mA inputs

If U_{CM} of an analog input channel is exceeded, internal equalizing currents result in erroneous measurements.

For 1 and 2 channel terminals the internal GND is therefore fed out to a terminal point, so that the U_{CM} specification can be met through application-specific configuration of this GND point, even in cases of atypical sensor configuration.

Example 1

The 2-channel EL3012 is connected to 2 sensors, which are supplied with 5 and 24 V. Both current measurements are executed as low-side measurements. This connection type is permitted, because at I_{max} CMP_{ch1} and CMP_{ch2} are approx. 330 mV above 0 V, which means that U_{CM} is always < 0.5 V. The requirement of $U_{CM} < 10$ V (applicable to EL30xx) is therefore adhered to.

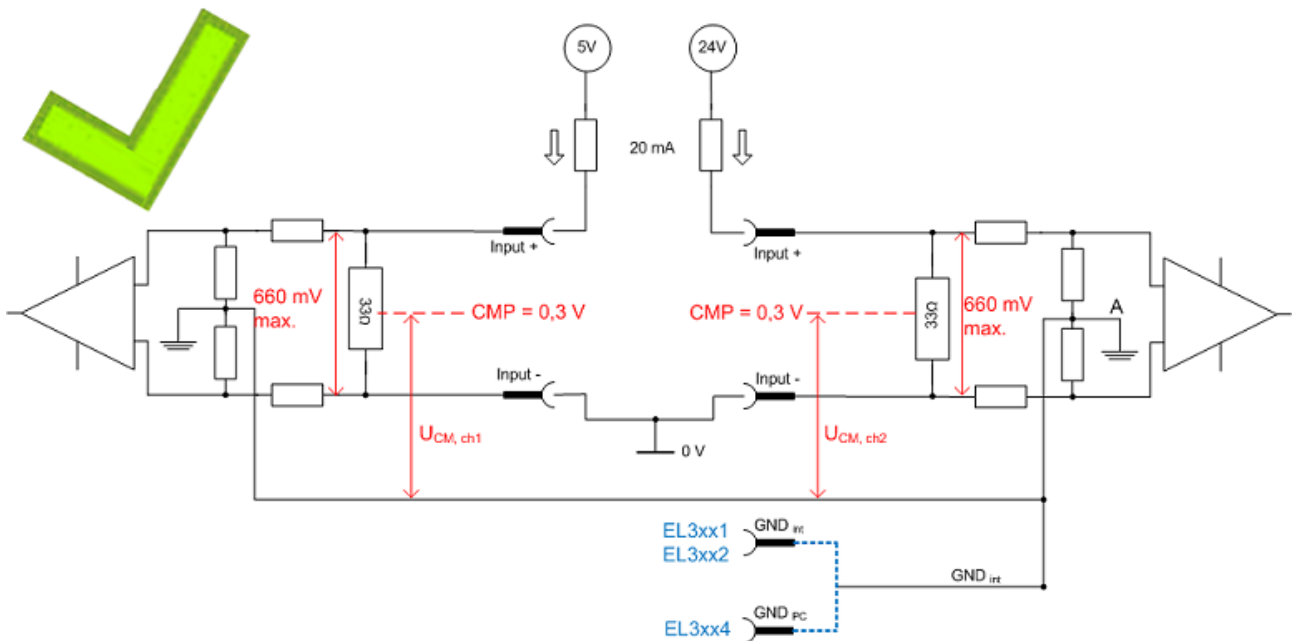


Fig. 39: Example 1: low-side measurement

If the EL30x1/EL30x2 or EL31x1/EL31x2 terminals have no external GND_{int} connection, the GND_{int} potential can adjust itself as required (referred to as "floating"). Please note that for this mode reduced measuring accuracy is to be expected.

Example 1a

Accordingly, this also applies if the floating point GND_{INT} is connected to another potential.

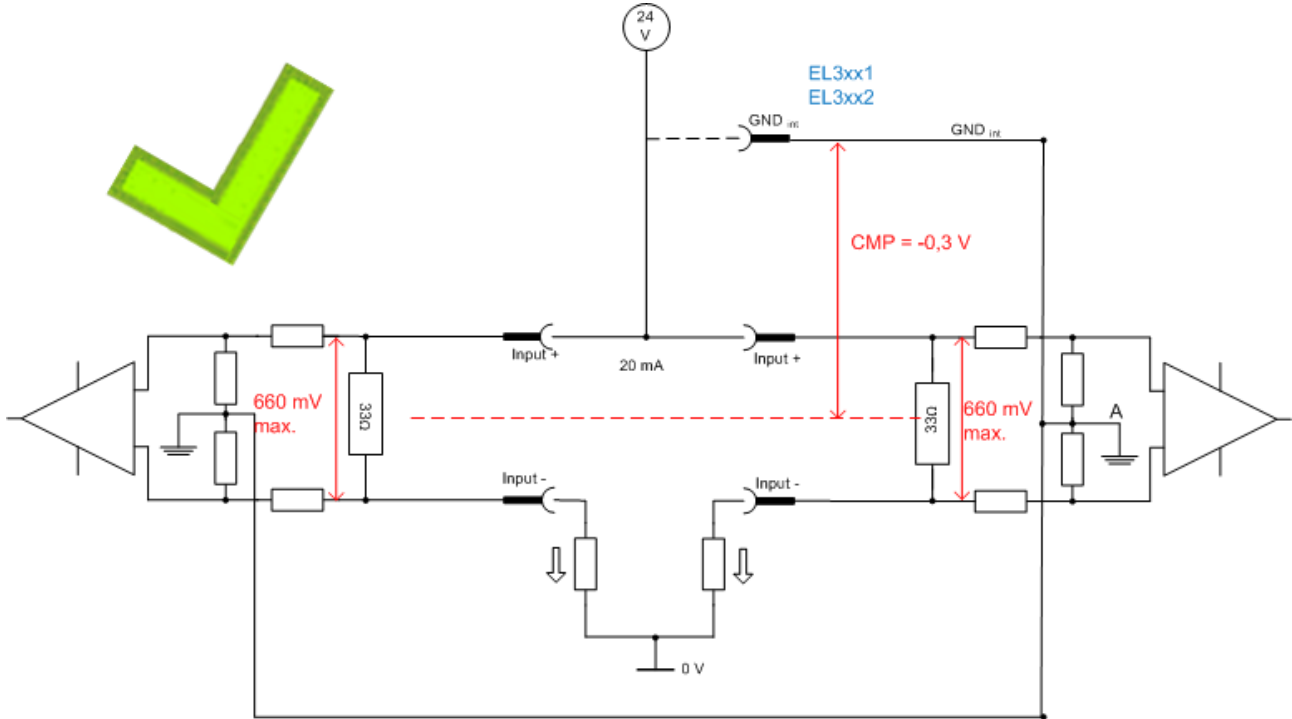


Fig. 40: Example 1a, high-side measurement

Example 2

The same EL3012 is now again connected with the two 20 mA sensors, although this time with one low-side measurement at 5 V and one high-side measurement at 12 V. This results in significant potential differences $U_{CM} > 10 V$ (applicable to EL30xx) between the two channels, which is not permitted.

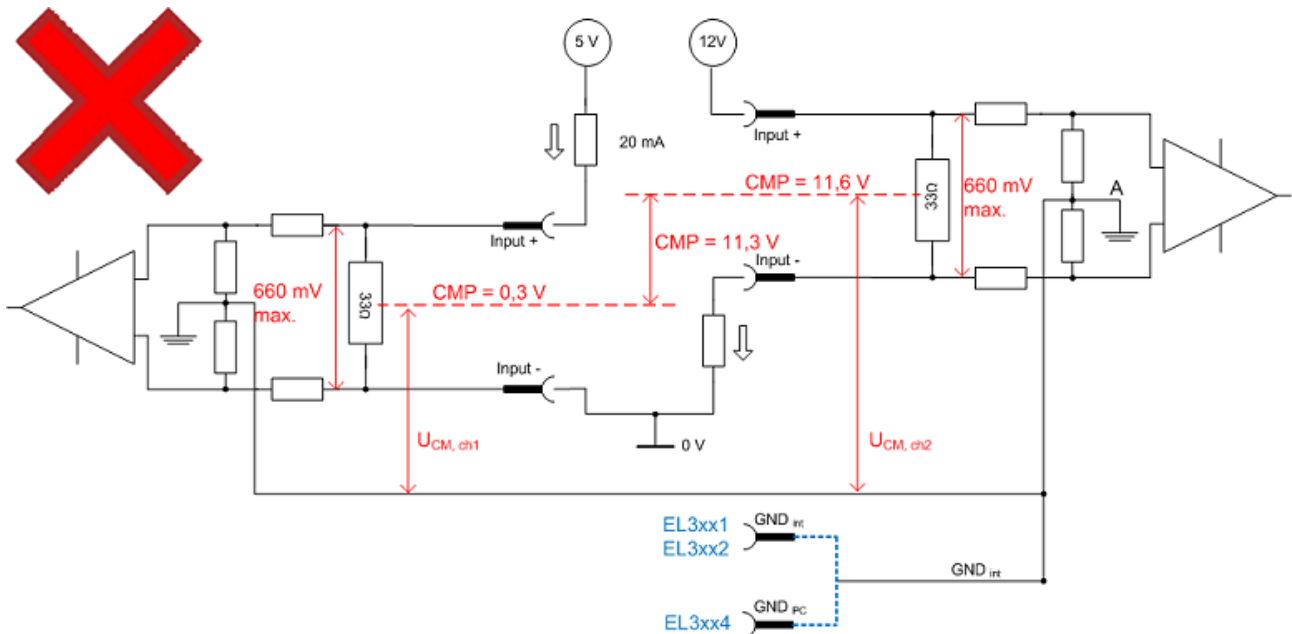


Fig. 41: Example 2, high-side/low-side measurement

To rectify this, GND_{int} can in this case be connected externally with an auxiliary potential of 6 V relative to "0 V". The resulting A/GND_{int} will be in the middle, i.e. approx. 0.3 V or 11.6 V.

Example 3

In the EL3xx4 terminals GND_{int} is internally connected with the negative power contact. The choice of potential is therefore limited.

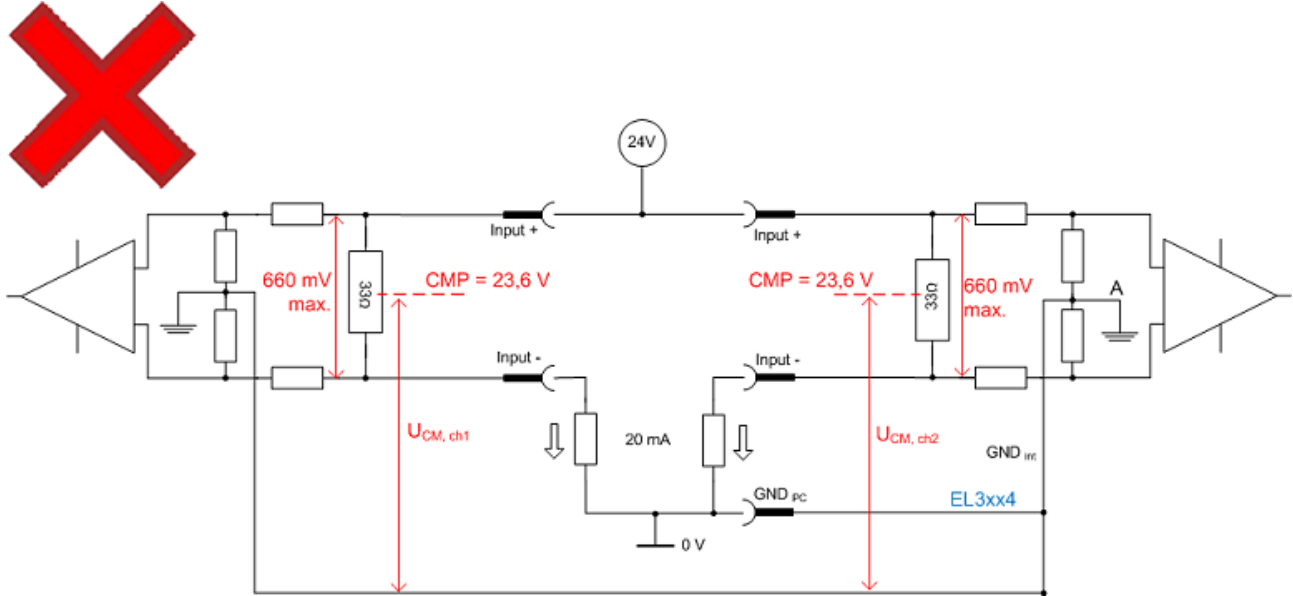


Fig. 42: Invalid EL3xx4 configuration

The resulting CMP is 23.6 V, i.e. $\gg 10$ V (applicable to EL30xx). The EL30xx4/EL31xx4 terminals should therefore be configured such that CMP is always less than $U_{CM,max}$.

Summary

This results in certain concrete specifications for external connection with 0/4..20 mA sensors:

- We recommend connecting GND_{int} with a low-impedance potential, because this significantly improves the measuring accuracy of the EL30xx/31xx. Please note the instructions relating to the U_{CM} potential reference.
- The U_{CM} potential reference must be adhered to between $CMP \leftrightarrow GND_{int}$ and $CMP_{ch(x)} \leftrightarrow CMP_{ch(y)}$. If this cannot be guaranteed, the single-channel version should be used.
- Terminal configuration:
 - EL3xx1/EL3xx2: GND_{int} is connected to terminal point for external connection. GND_{int} should be connected externally such that condition 2 is met.
 - EL3xx4: GND is connected with the negative power contact. The external connection should be such that condition 2 is met.

If the sensor cable is shielded, the shield should not be connected with the GND_{int} terminal point but with a dedicated low-impedance shield point.

- If terminal points of several EL30xx/EL31xx terminals are connected with each other, ensure that condition 2 is met.

● Connection of GND_{int}

i In the EL30xx1/EL30xx2 and EL31xx1/EL31xx2 terminals the internal GND, GND_{int} connection is fed out to terminal contacts.

To achieve a precise measurement result GND_{int} should be connected to a suitable external low-impedance potential, taking account the specifications for U_{CM} .

In the EL30xx4/EL31xx4 terminals GND_{int} is already connected with the negative power contact. Here too the specifications for U_{CM} must be followed.

5.11 EL37x2 - LEDs and pin assignment

5.11.1 EL37x2 - LEDs

LEDs (from hardware version 03)



Fig. 43: EL37x2 LEDs

LED	Color	Meaning	
RUN	green	These LEDs indicate the terminal's operating state:	
		off	State of the EtherCAT State Machine [▶ 103]: INIT = initialization of the terminal
		flashing uniformly	State of the EtherCAT State Machine: PREOP = function for mailbox communication and different standard-settings set
		flashing slowly	State of the EtherCAT State Machine: SAFEOP = verification of the Sync Manager [▶ 104] 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
		flashing rapidly*	State of the EtherCAT State Machine: BOOTSTRAP = function for firmware updates [▶ 163] of the terminal

* only RUN 1 LED

5.11.2 EL37x2 - Pin assignment

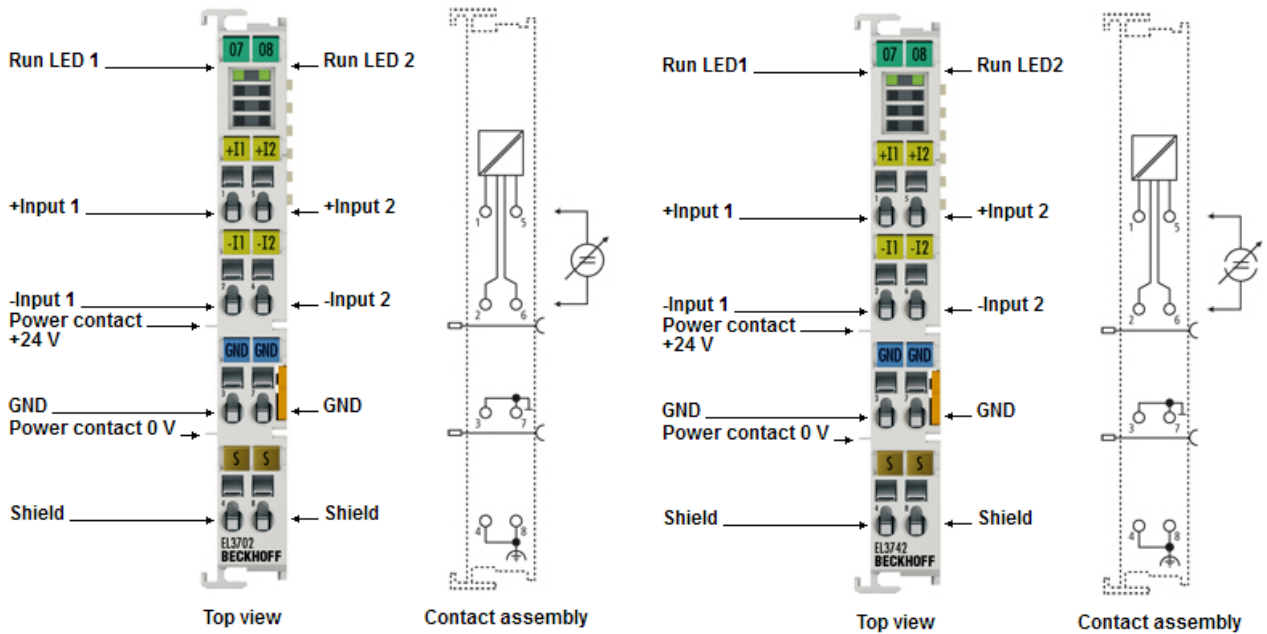


Fig. 44: Pin assignment EL3702, EL3742

Pin assignment and wiring

Terminal point		Description
Name	No.	
+ Input 1	1	+ Input 1
- Input 1	2	- Input 1
GND	3	Signal ground for input 1 (internally connected to terminal point 7)
Shield	4	Shield
+ Input 2	5	+ Input 2
- Input 2	6	- Input 2
GND	7	Signal ground for input 2 (internally connected to terminal point 3)
Shield	8	Shield

Electrical isolation of the inputs

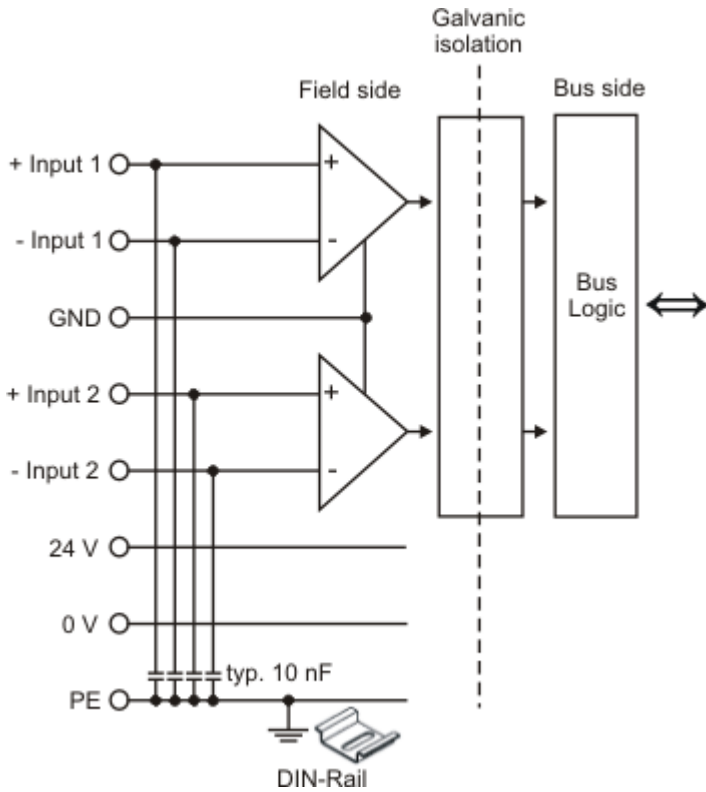


Fig. 45: Electrical isolation of the inputs

Sample: EL3602 wiring

In figure 2, the circuit diagram with a variable voltage source (channel 2) should be regarded as equivalent to the configuration of channel 1 with a sensor and measuring amplifier.

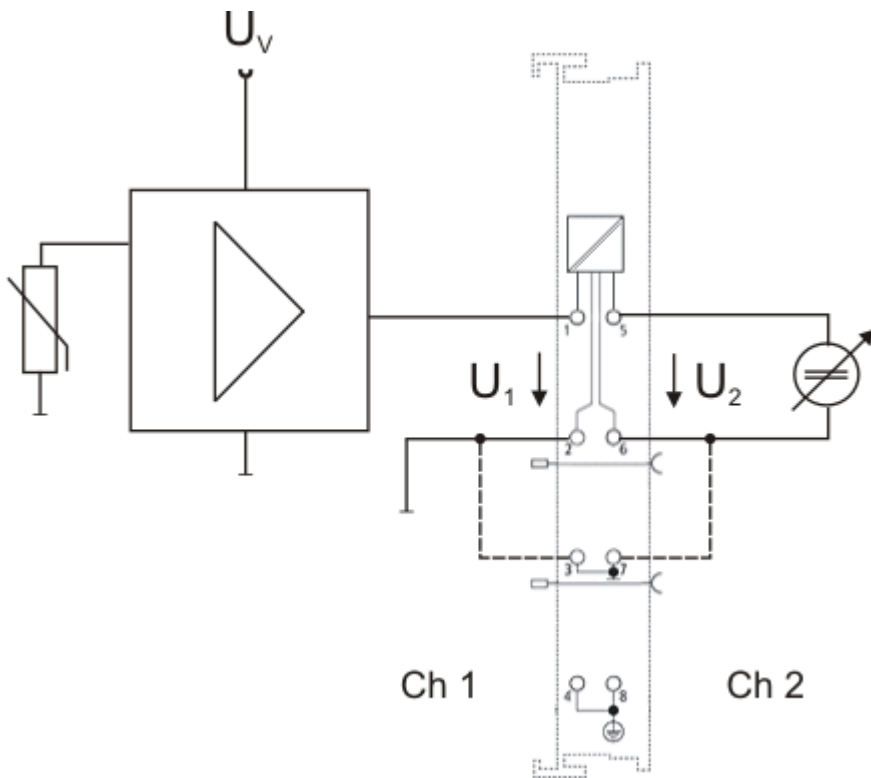


Fig. 46: EL3702 - equivalent circuit diagram with variable voltage source

Sample: EL3742 wiring

In figure 2, the circuit diagram with a variable current source (channel 2) should be regarded as equivalent to the configuration of channel 1 with a sensor.

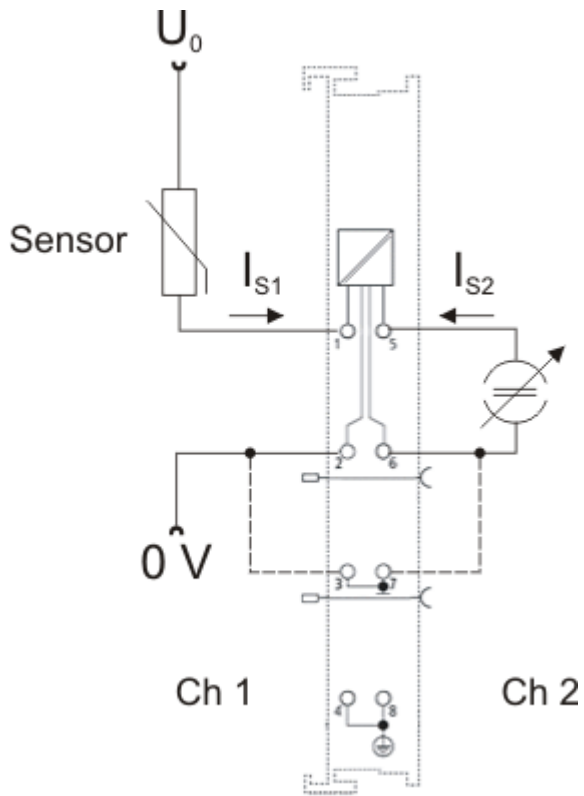


Fig. 47: EL3742 - equivalent circuit diagram with variable current source

6 Commissioning

6.1 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:

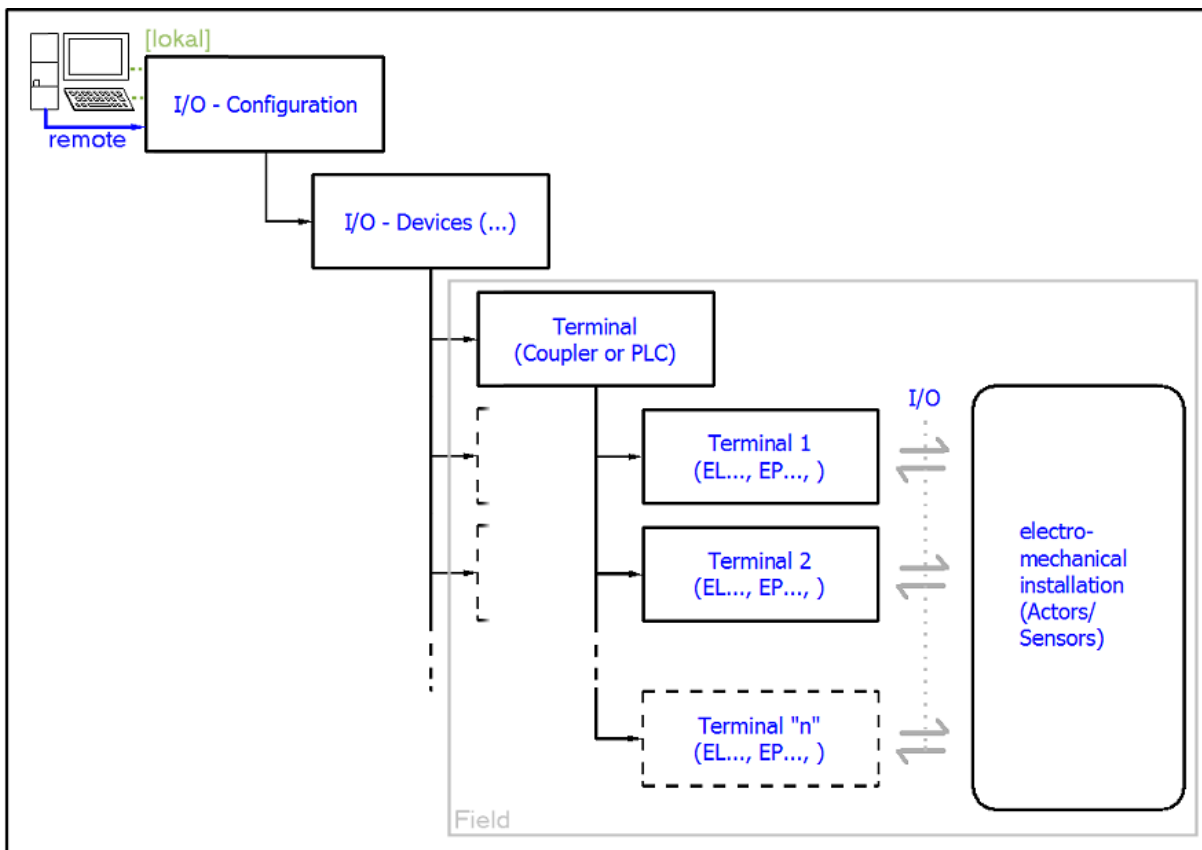


Fig. 48: Relationship between user side (commissioning) and installation

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)

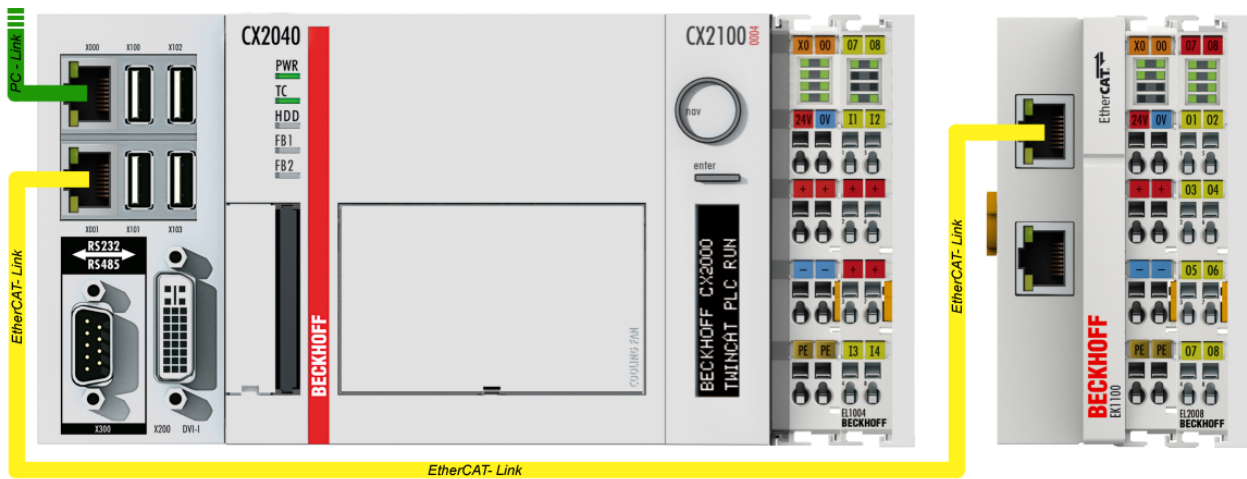


Fig. 49: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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

6.1.1 TwinCAT 2

Startup

TwinCAT 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:

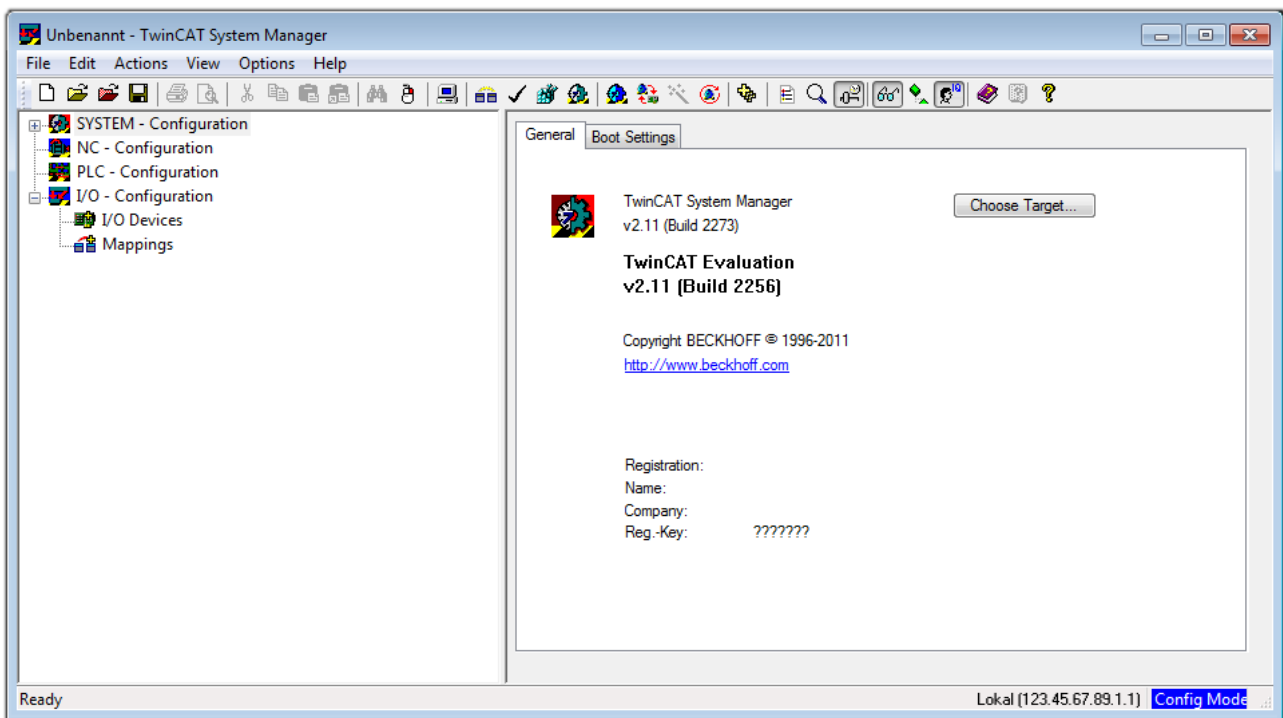



Fig. 50: Initial TwinCAT 2 user interface

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 [▶ 59]".

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:

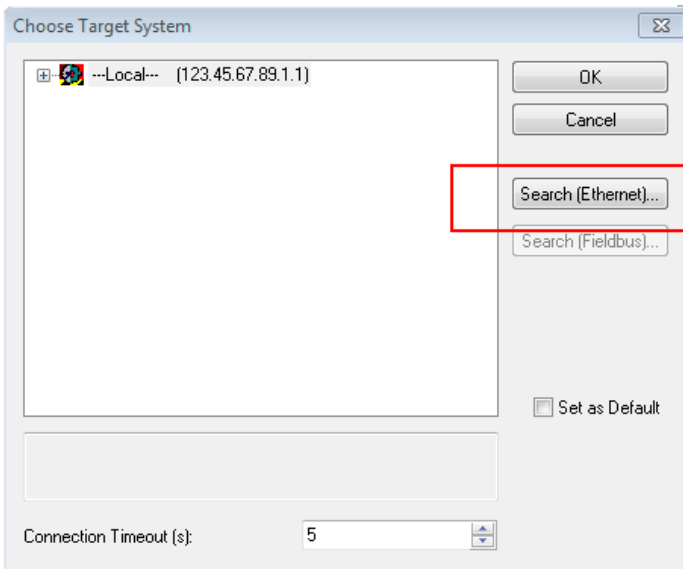


Fig. 51: Selection of the target system

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.

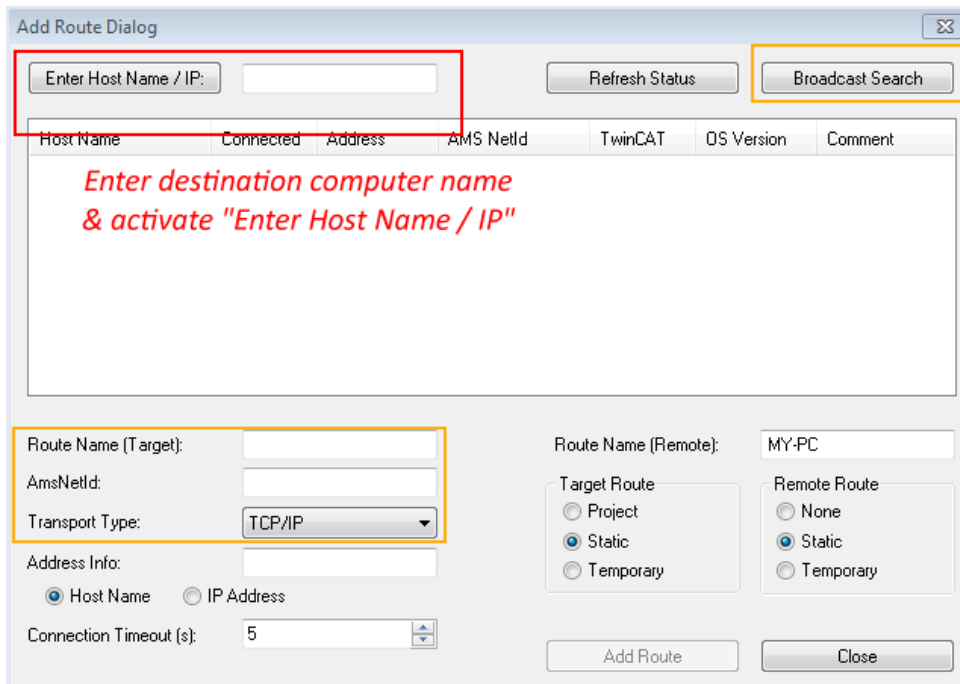
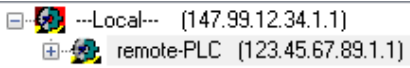


Fig. 52: Specify the PLC for access by the TwinCAT System Manager: selection of the target system



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).

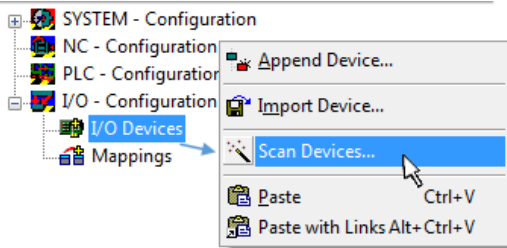


Fig. 53: Select "Scan Devices..."

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

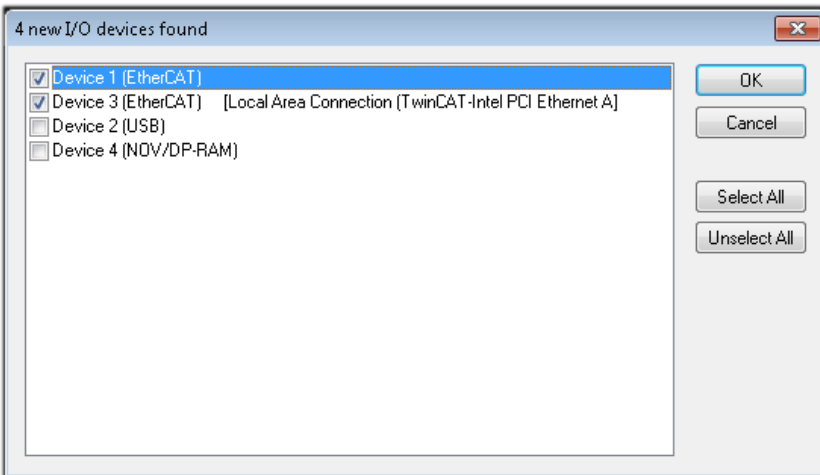


Fig. 54: Automatic detection of I/O devices: selection the devices to be integrated

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

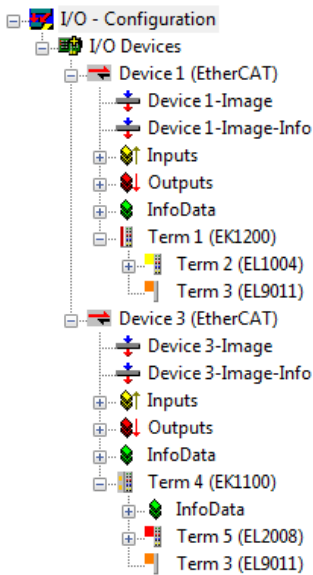


Fig. 55: Mapping of the configuration in the TwinCAT 2 System Manager

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:

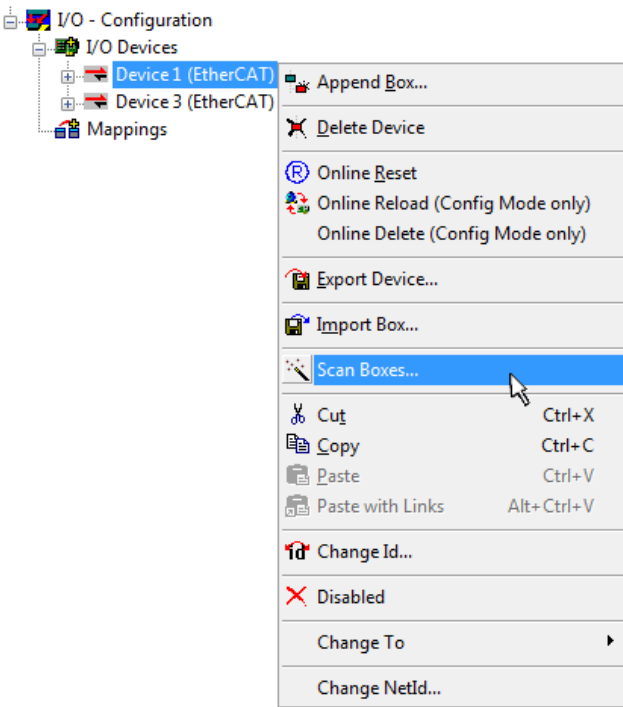


Fig. 56: Reading of individual terminals connected to a device

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:

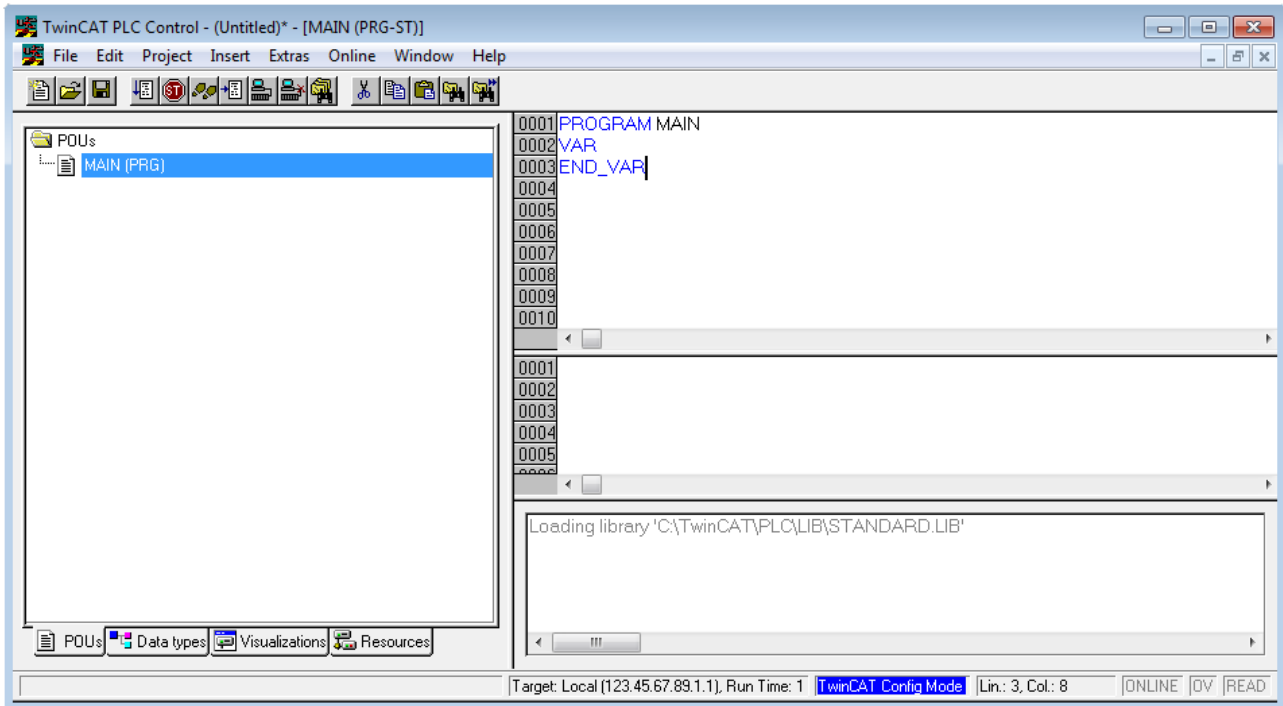


Fig. 57: TwinCAT PLC Control after startup

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

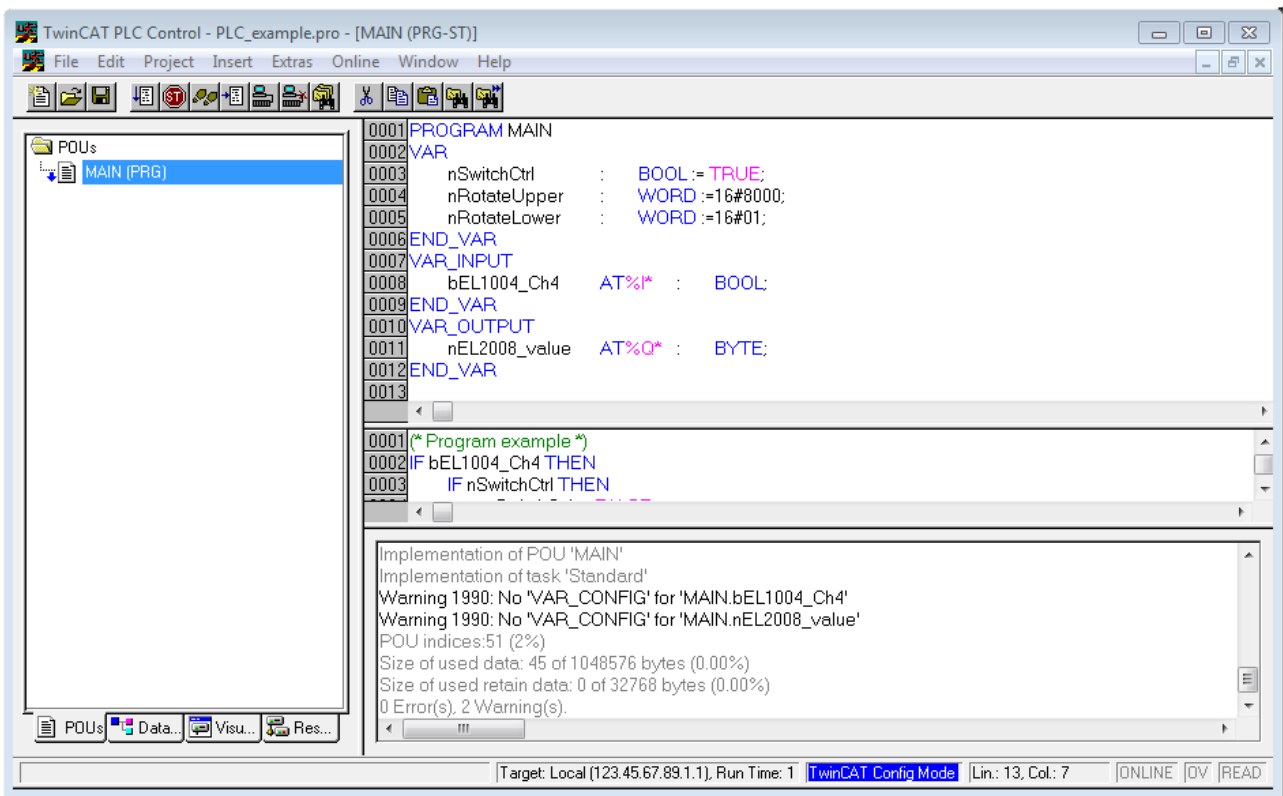


Fig. 58: Sample program with variables after a compile process (without variable integration)

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...":

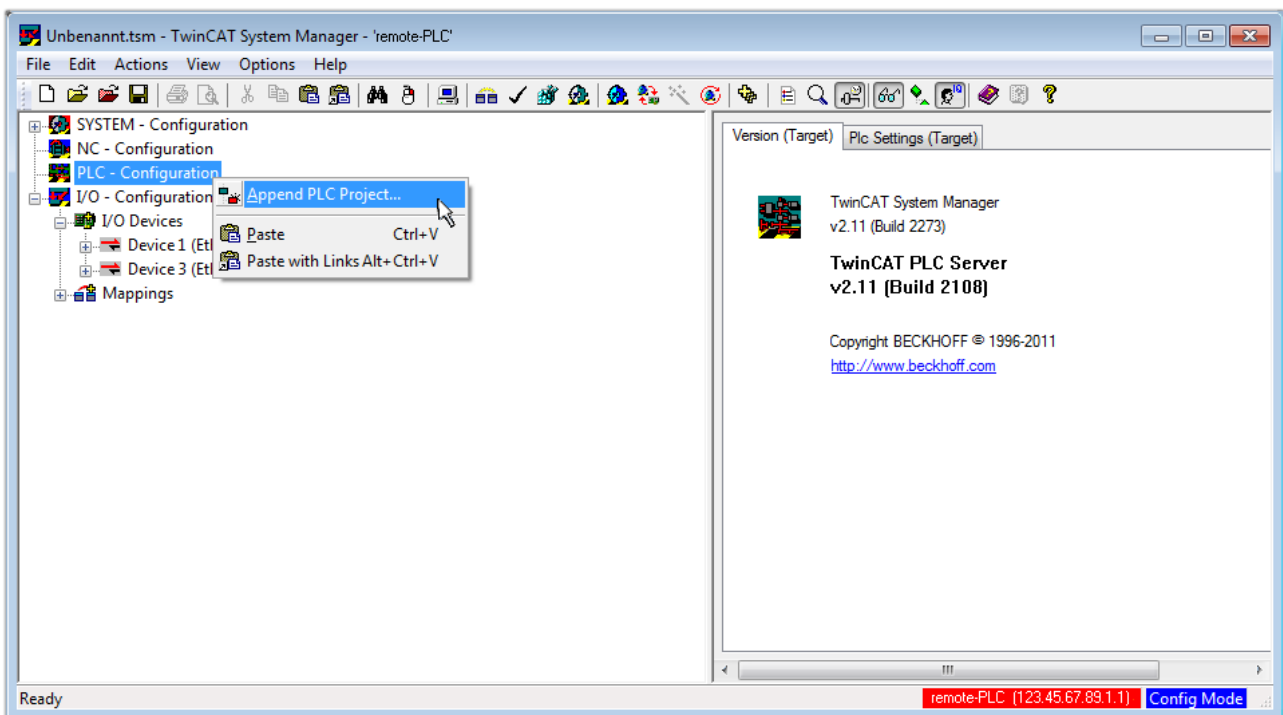


Fig. 59: Appending the TwinCAT PLC Control project

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:

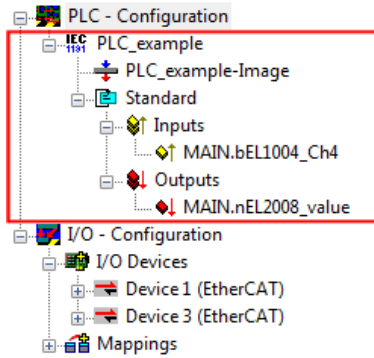


Fig. 60: PLC project integrated in the PLC configuration of the System Manager

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":

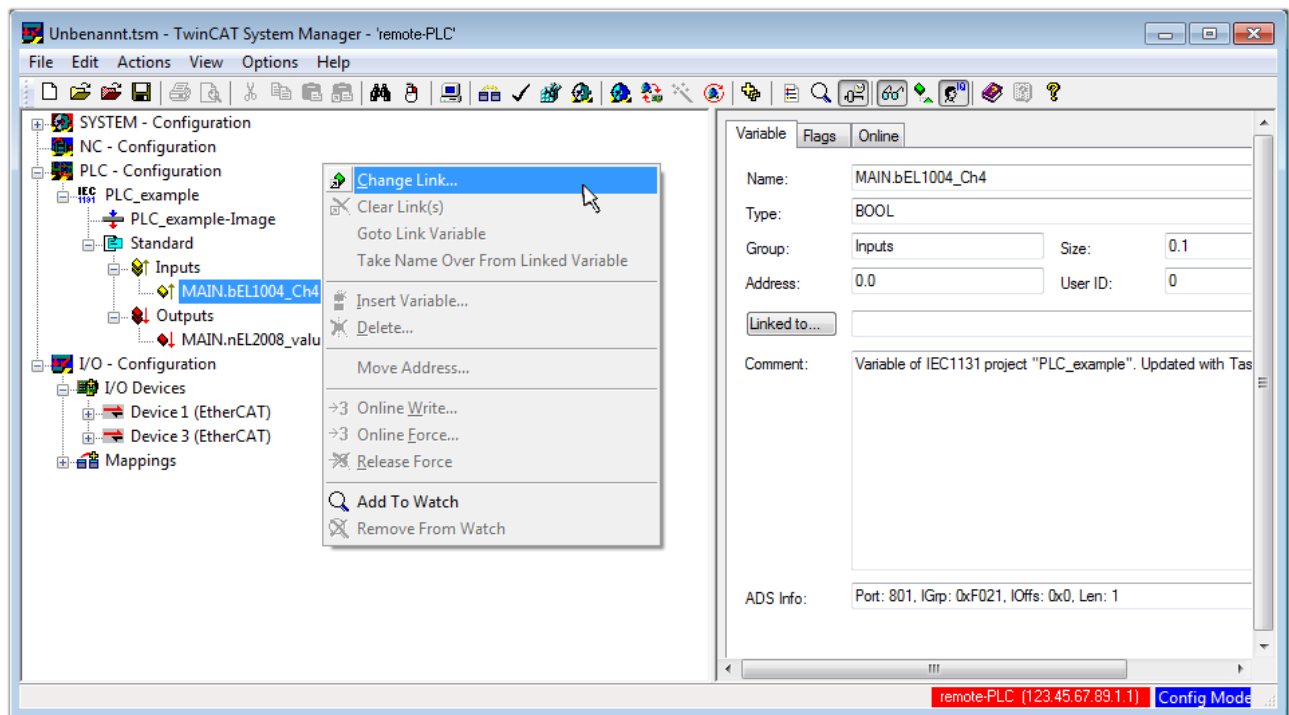


Fig. 61: Creating the links between PLC variables and process objects

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

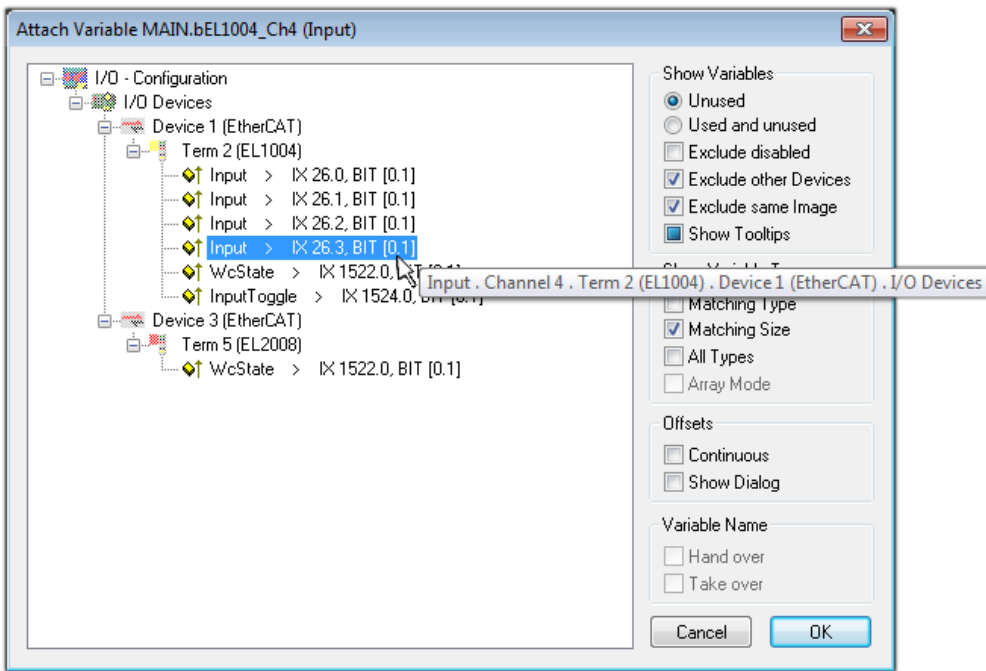


Fig. 62: Selecting PDO of type BOOL

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:

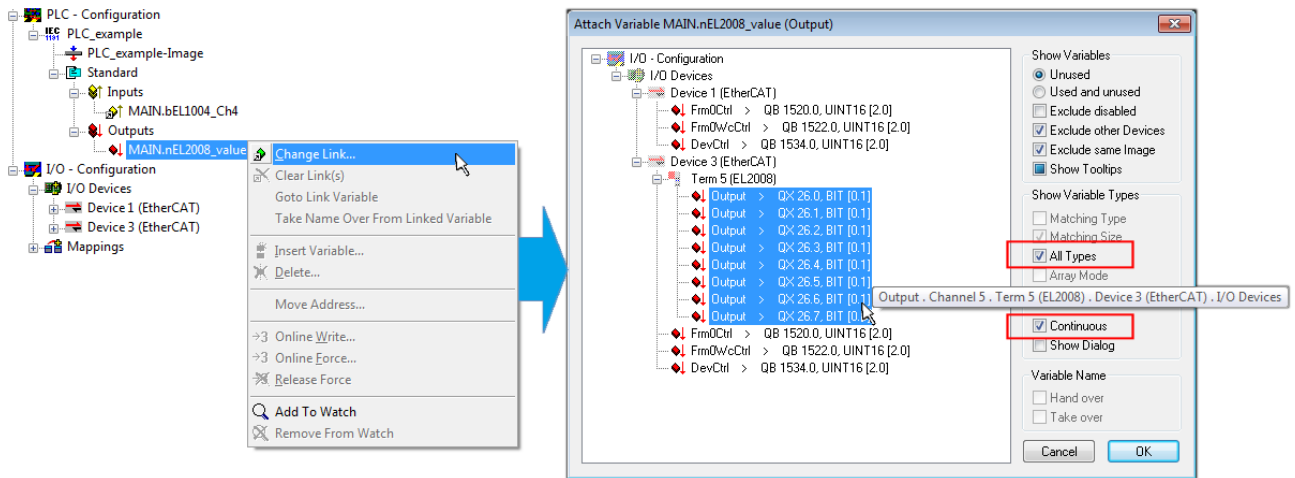



Fig. 63: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

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:

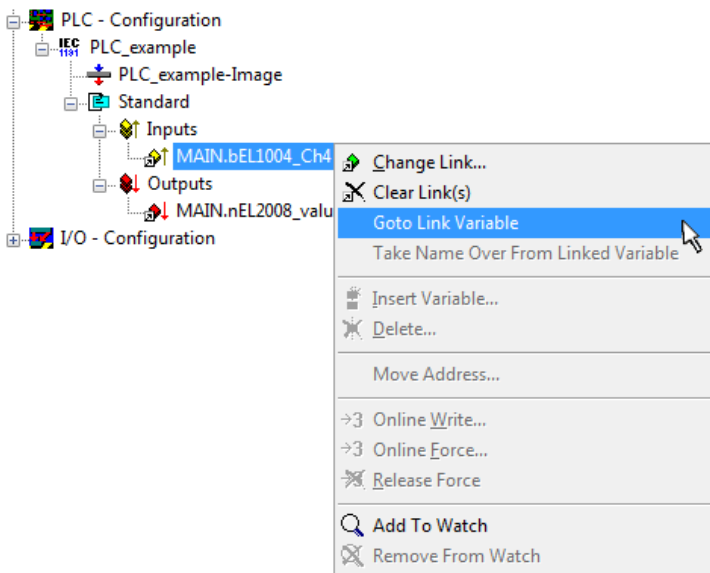

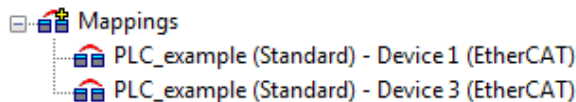


Fig. 64: Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample

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...":

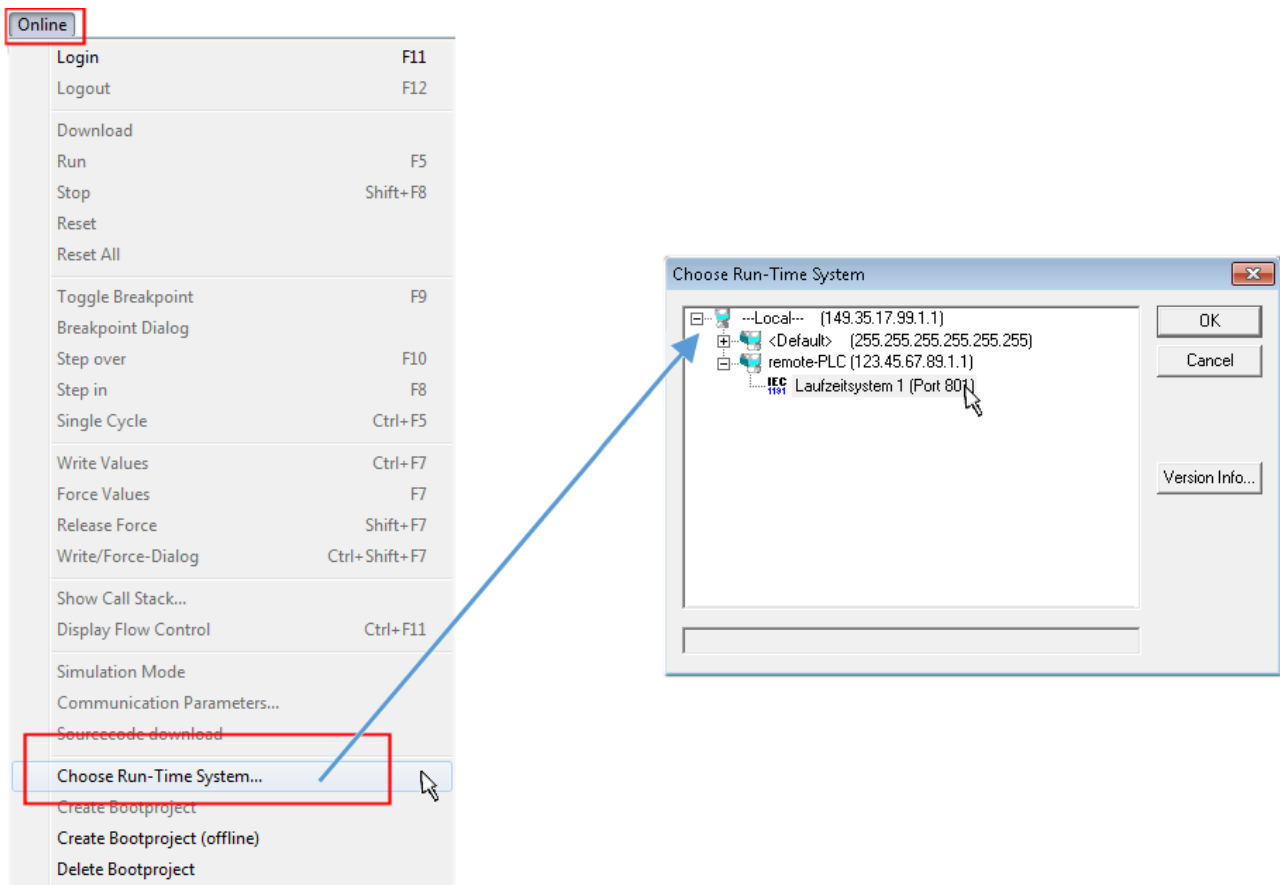



Fig. 65: Choose target system (remote)

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:

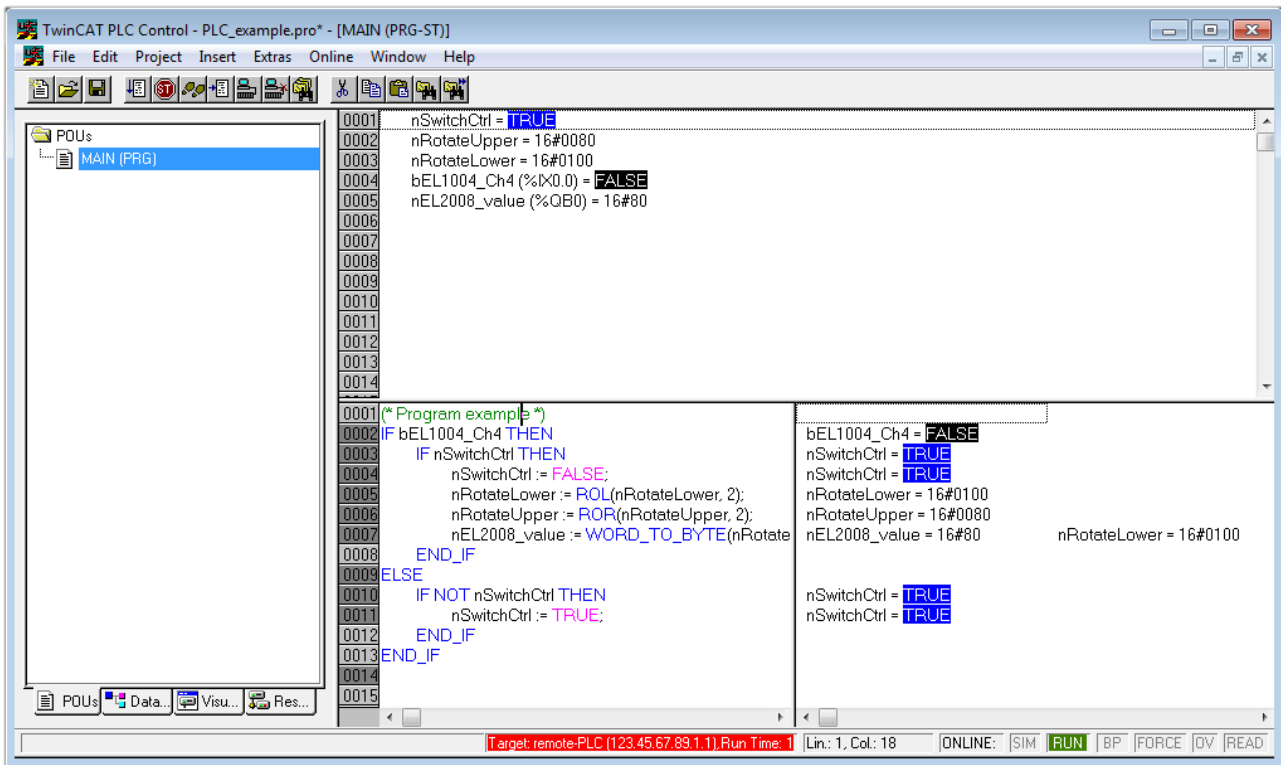


Fig. 66: PLC Control logged in, ready for program startup

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

6.1.2 TwinCAT 3


Startup

TwinCAT 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. 67: 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):

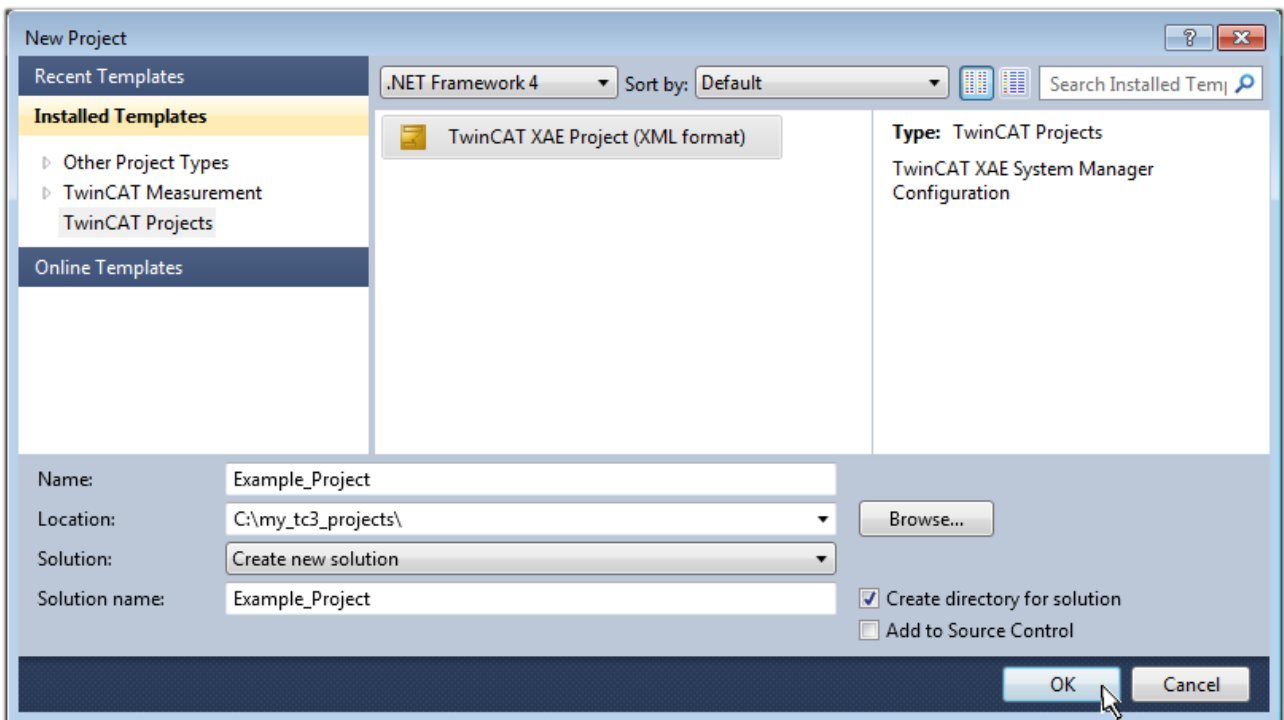


Fig. 68: Create new TwinCAT project

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

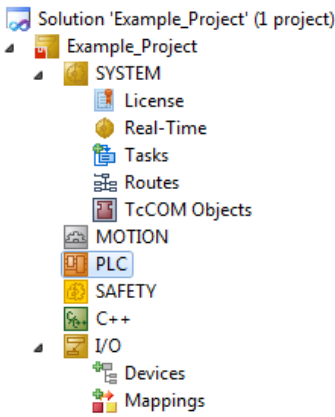
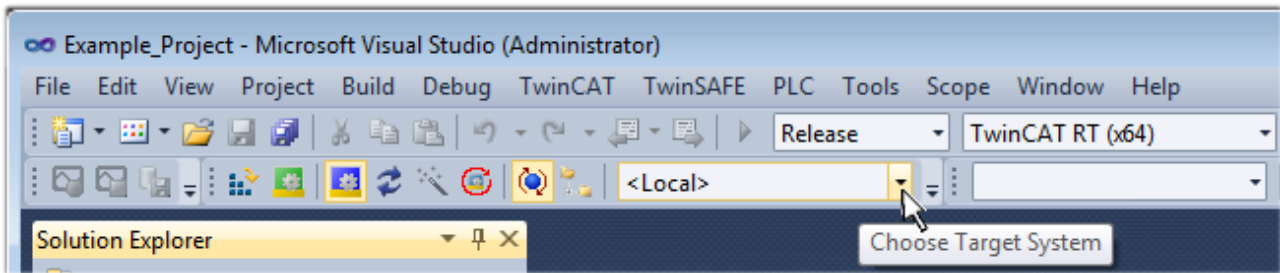


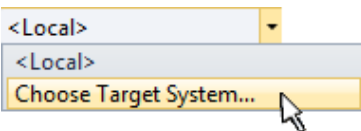
Fig. 69: New TwinCAT3 project in the project folder explorer

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 [▶ 70]".

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:

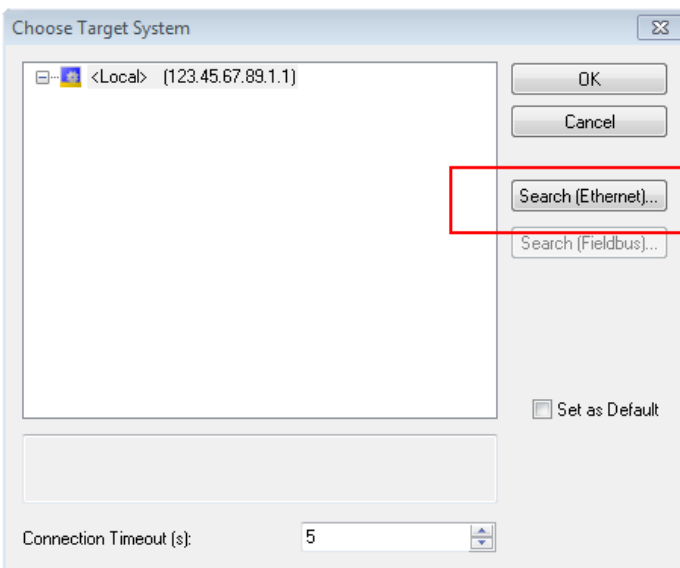


Fig. 70: Selection dialog: Choose the target system

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.

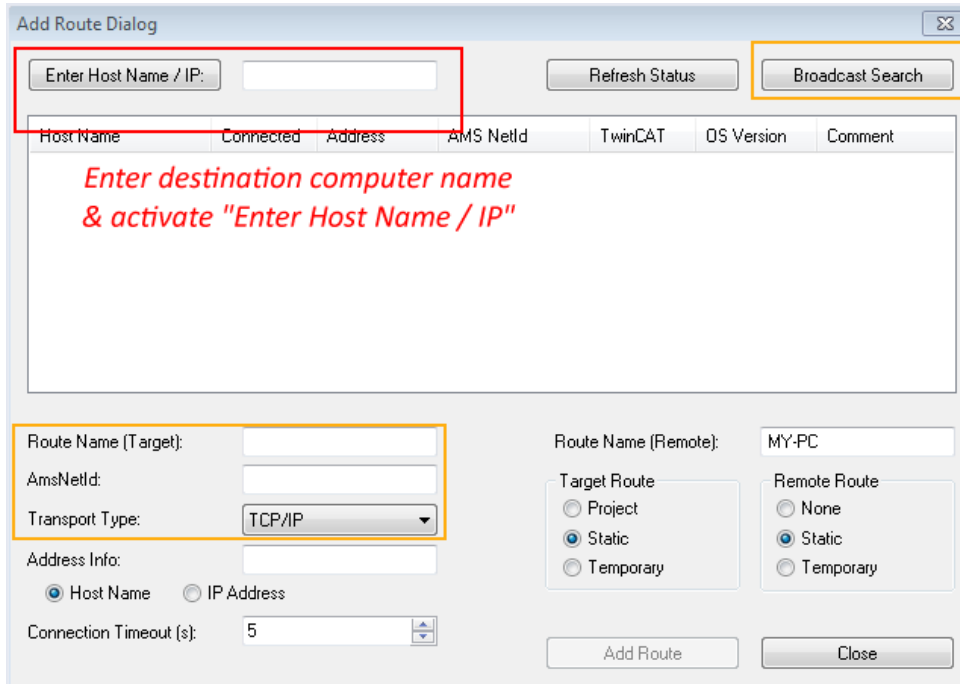
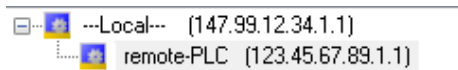


Fig. 71: Specify the PLC for access by the TwinCAT System Manager: selection of the target system


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)".

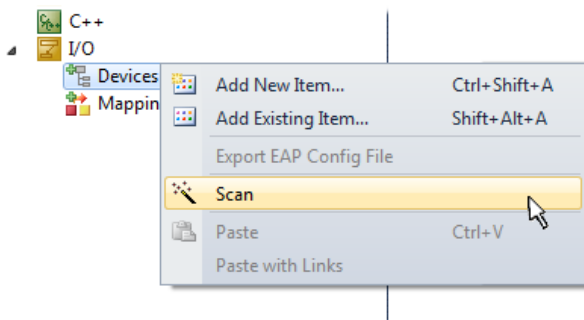


Fig. 72: Select "Scan"

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

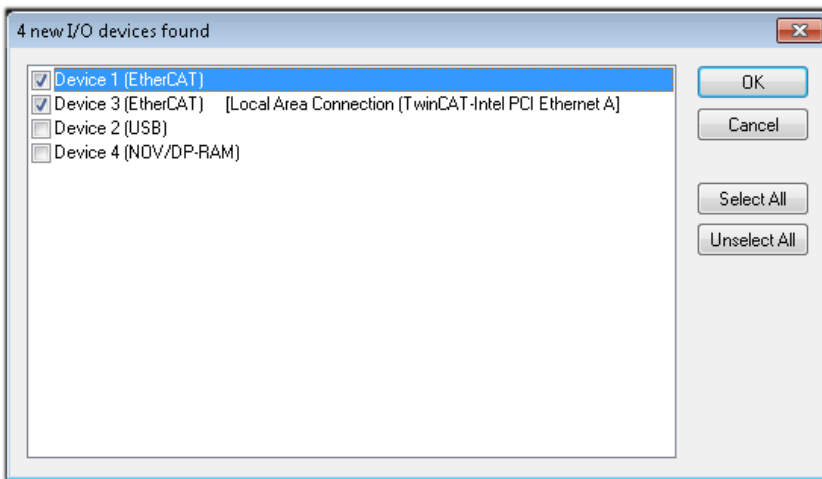


Fig. 73: Automatic detection of I/O devices: selection the devices to be integrated

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 \[▶ 56\]](#) described at the beginning of this section, the result is as follows:

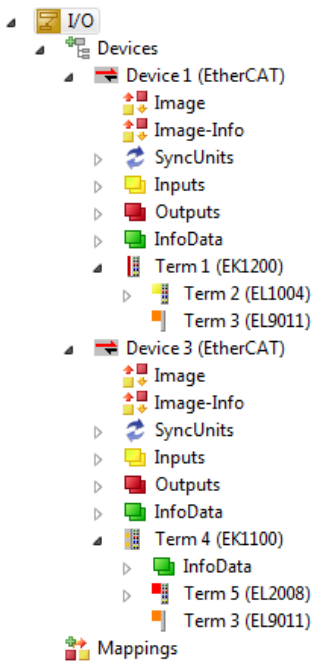


Fig. 74: Mapping of the configuration in VS shell of the TwinCAT3 environment

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:

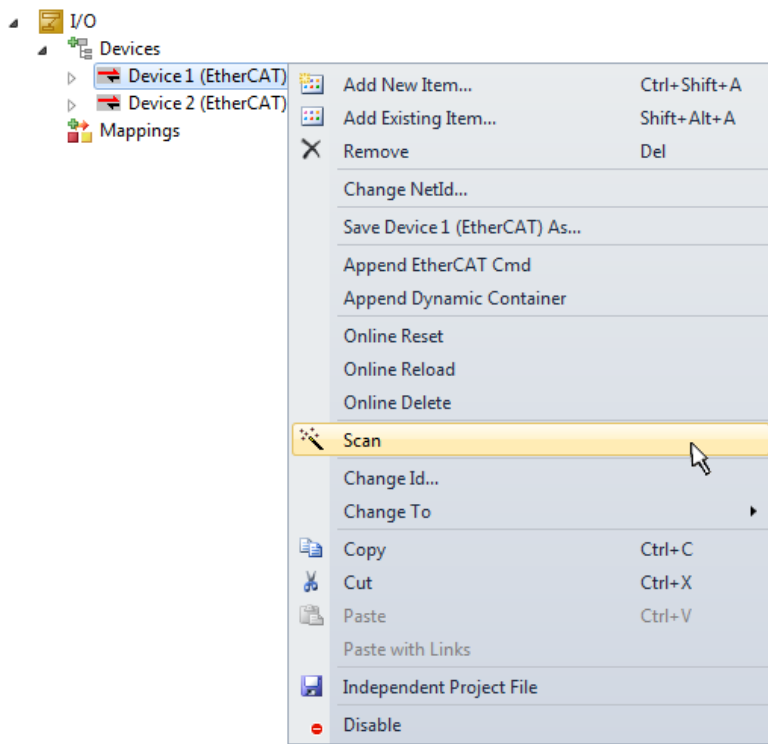


Fig. 75: Reading of individual terminals connected to a device

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....":

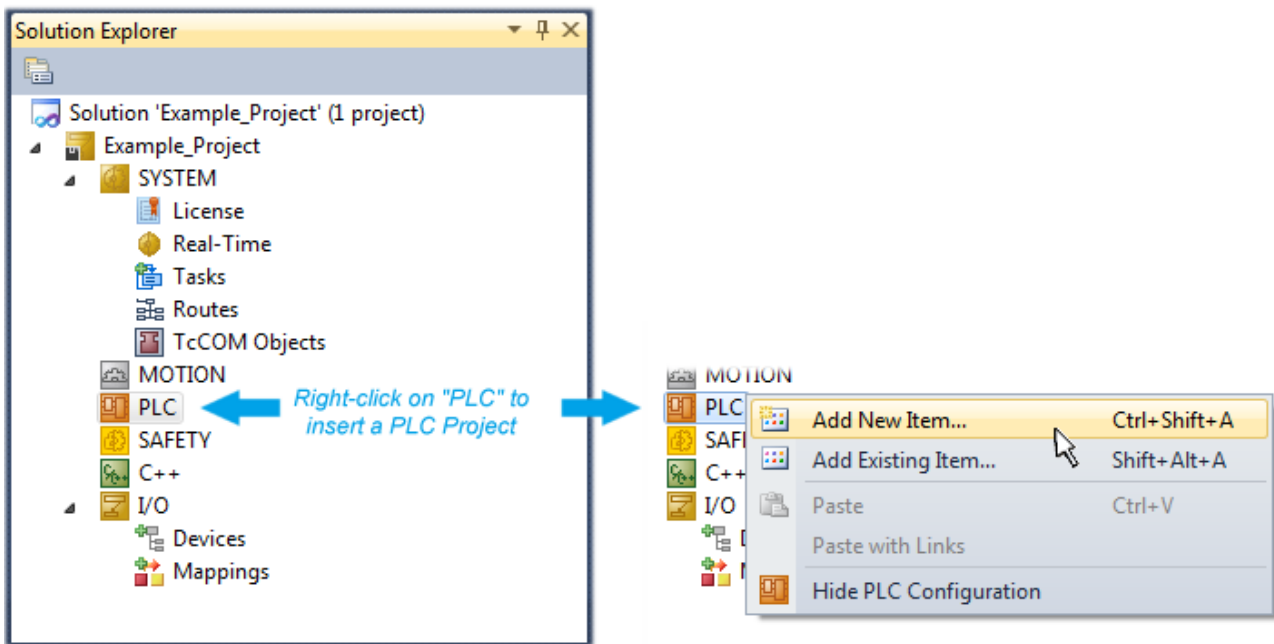


Fig. 76: Adding the programming environment in "PLC"

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

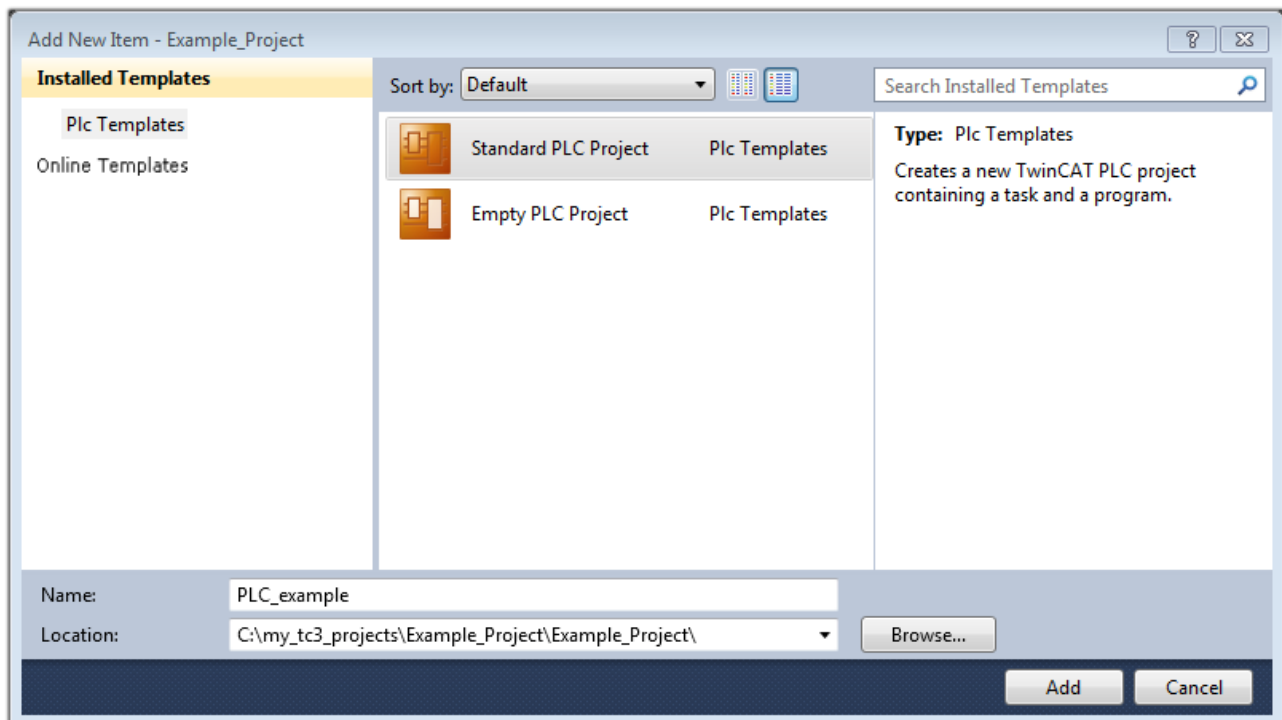


Fig. 77: Specifying the name and directory for the PLC programming environment

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:

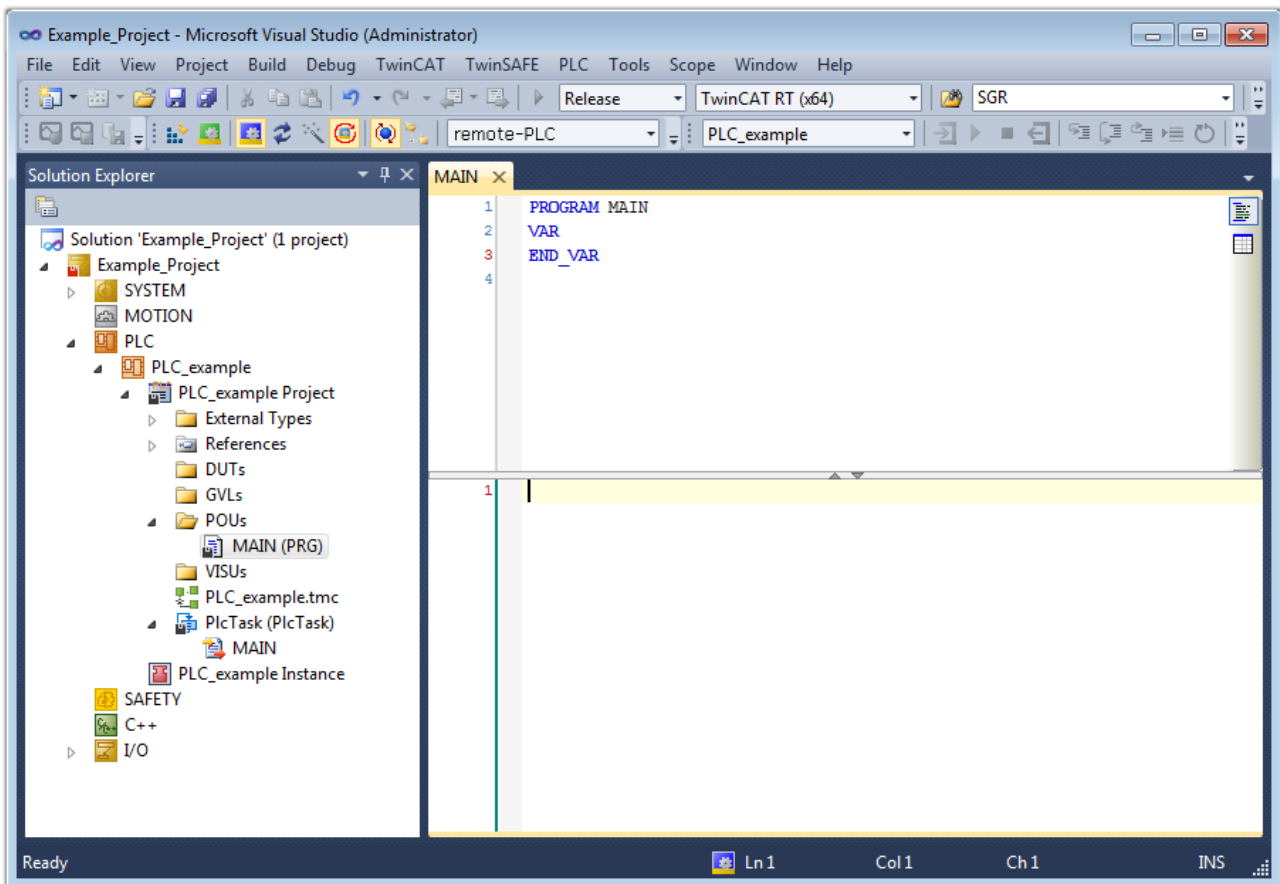


Fig. 78: Initial "Main" program of the standard PLC project

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

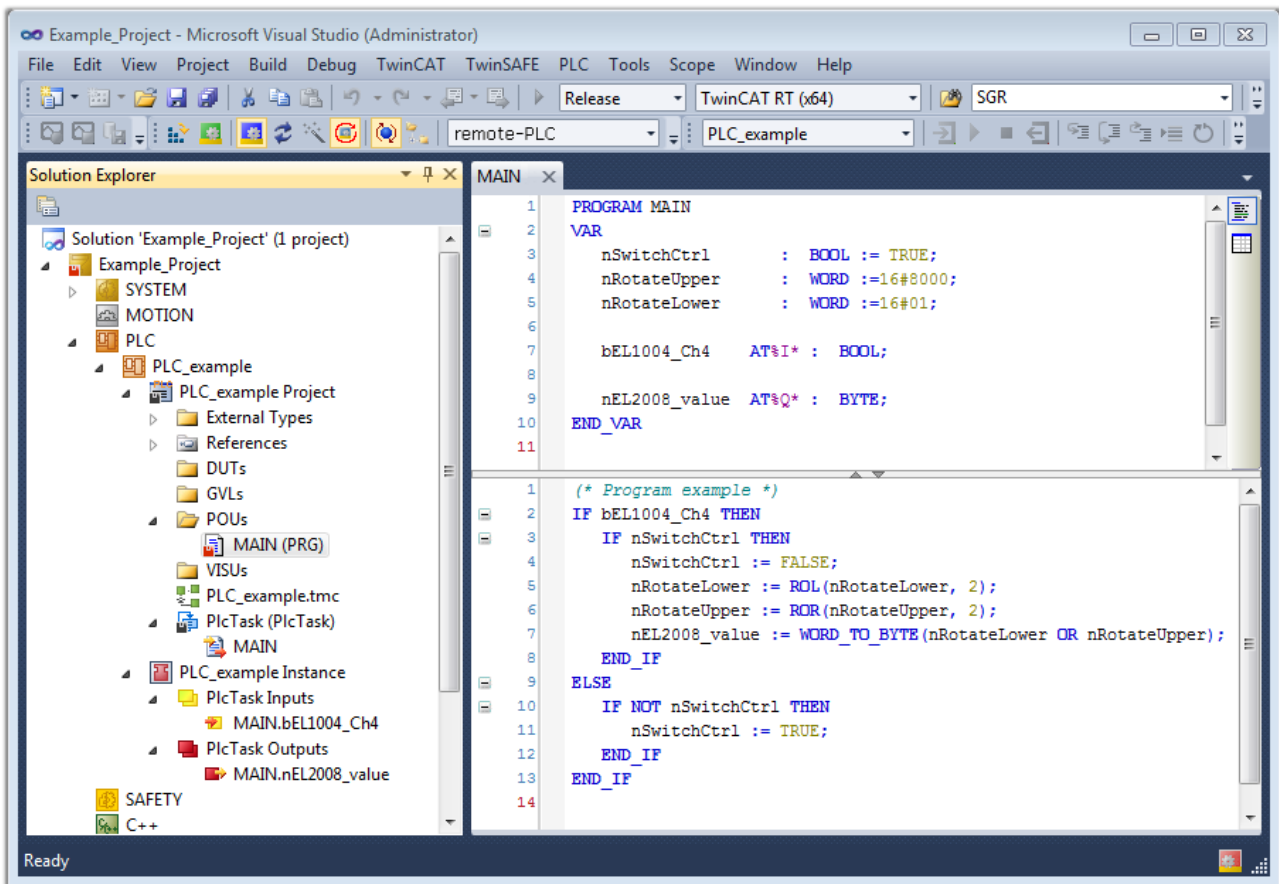


Fig. 79: Sample program with variables after a compile process (without variable integration)

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

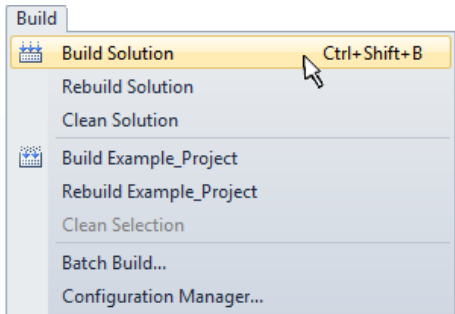
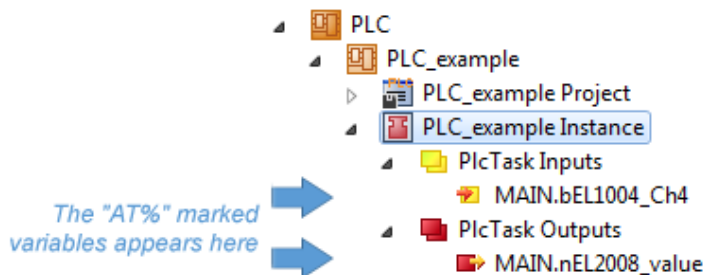


Fig. 80: Start program compilation

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:

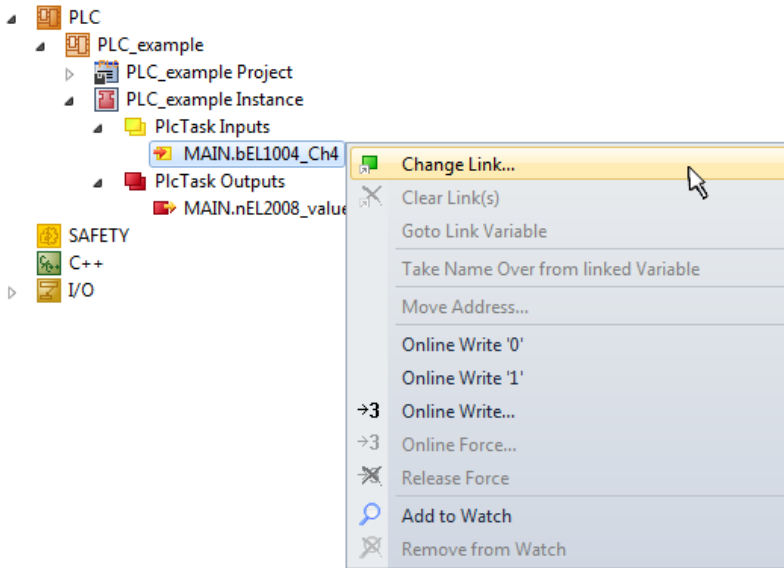


Fig. 81: Creating the links between PLC variables and process objects

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

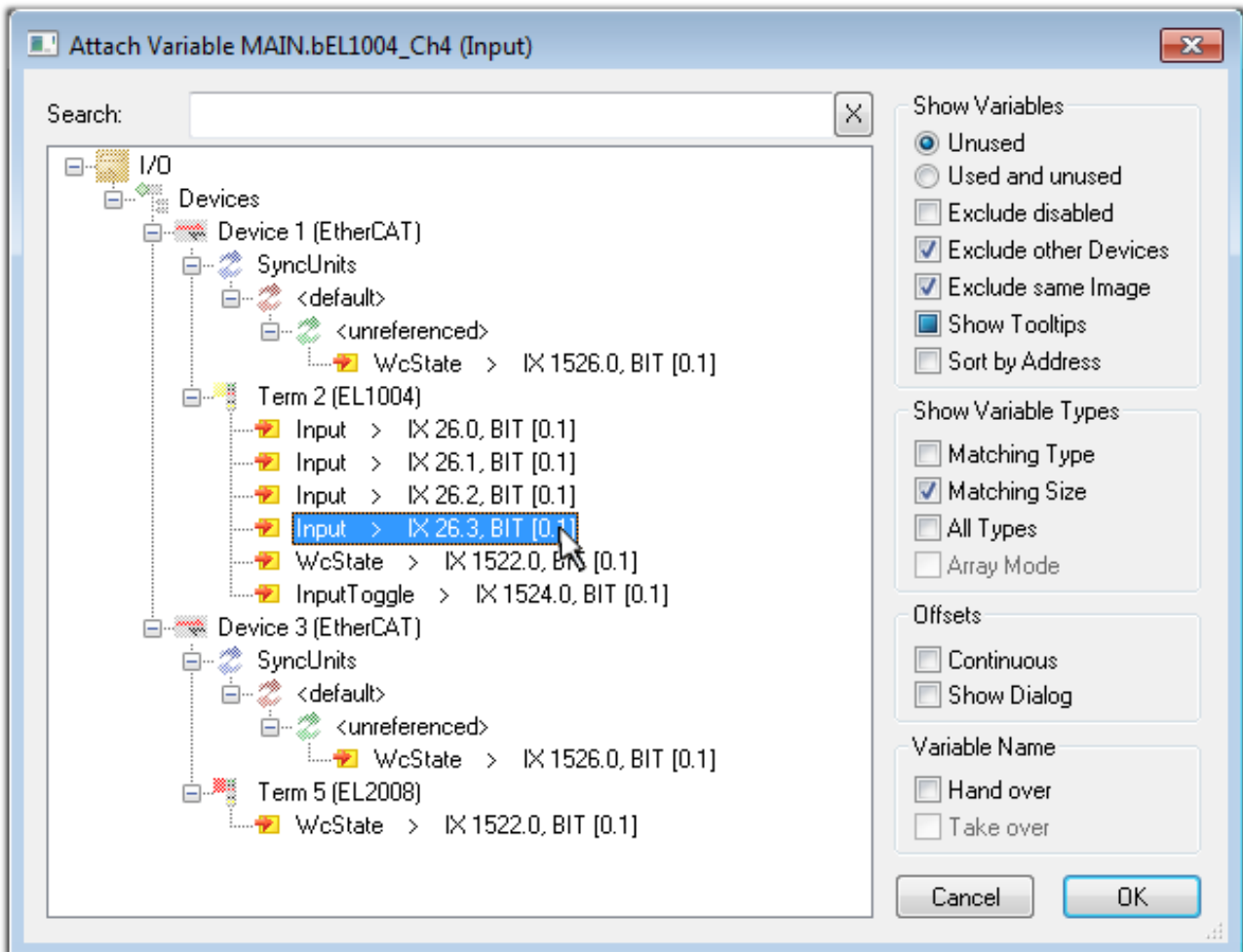


Fig. 82: Selecting PDO of type BOOL

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:

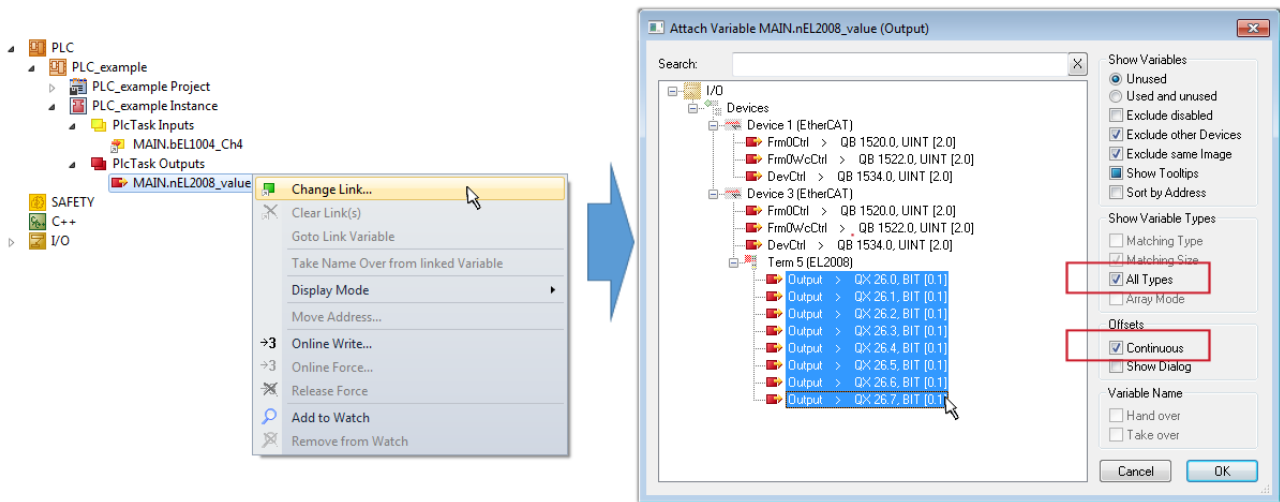



Fig. 83: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

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:

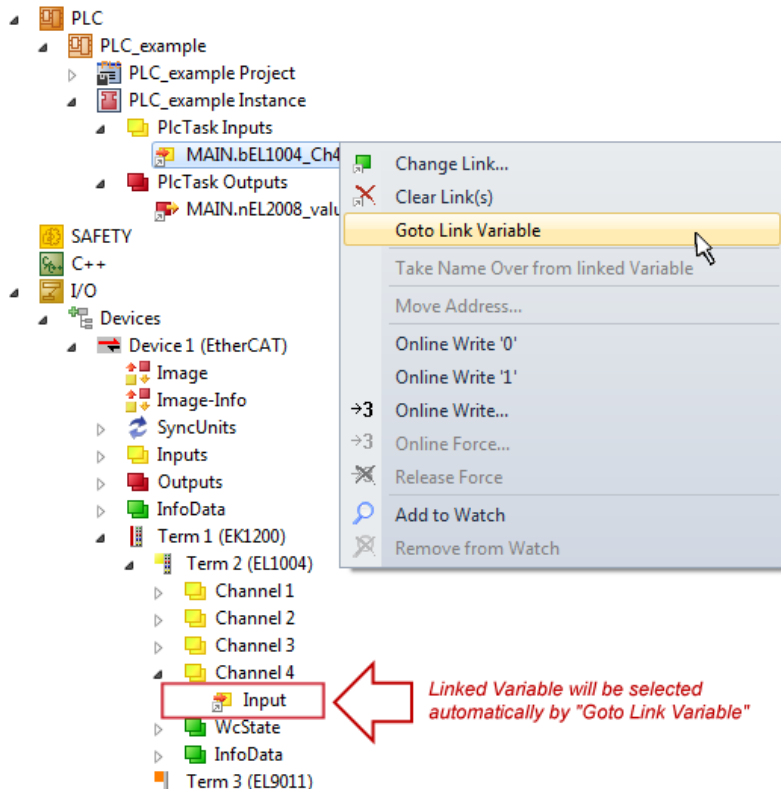


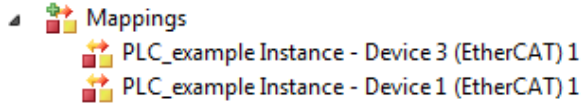
Fig. 84: Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample

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:

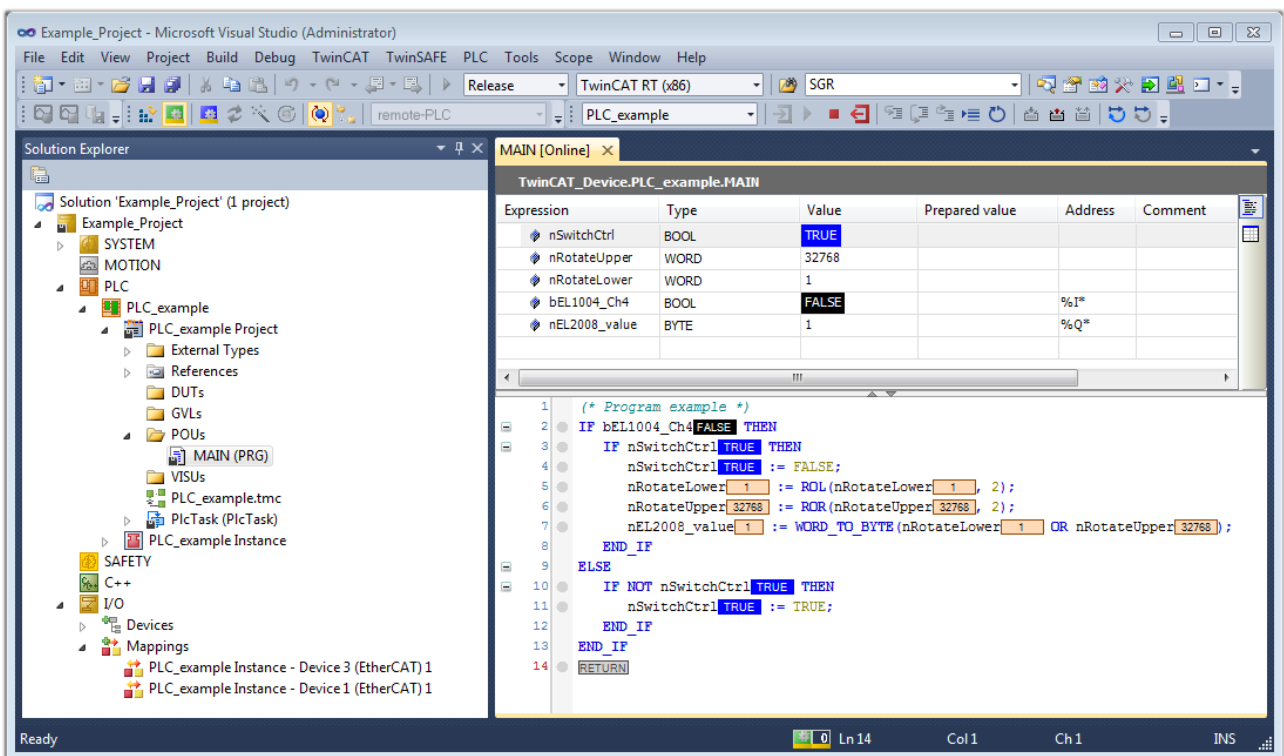


Fig. 85: TwinCAT development environment (VS shell): logged-in, after program startup

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).

6.2 TwinCAT Development Environment

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

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

Details:

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

Additional features:

- **TwinCAT 3 (eXtended Automation):**
 - Visual-Studio®-Integration
 - Choice of the programming language
 - Supports object orientated extension of IEC 61131-3
 - Usage of C/C++ as programming language for real time applications
 - Connection to MATLAB®/Simulink®
 - Open interface for expandability
 - Flexible run-time environment
 - Active support of Multi-Core- 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>.

6.2.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways. 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.

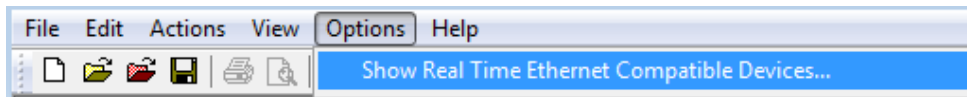


Fig. 86: System Manager "Options" (TwinCAT 2)

This has to be called up by the Menü "TwinCAT" within the TwinCAT 3 environment:

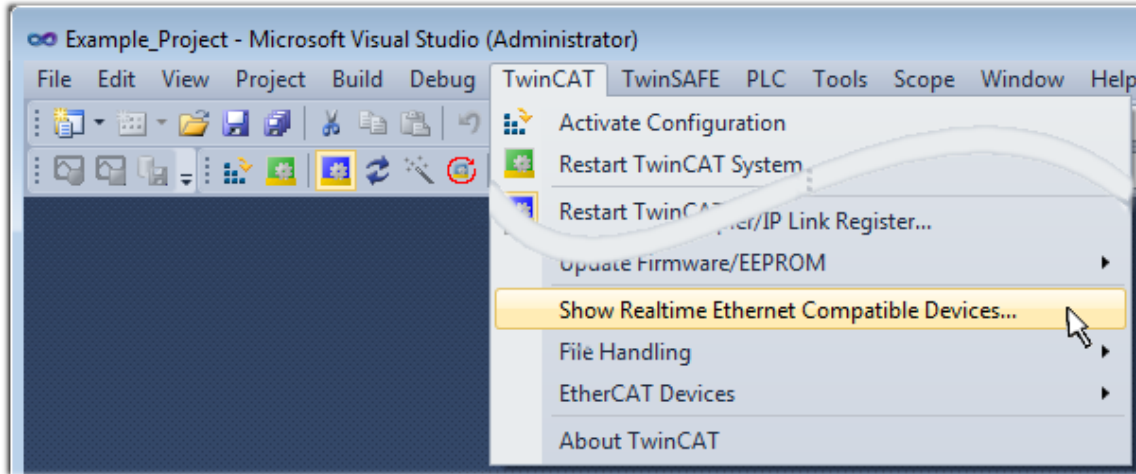


Fig. 87: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

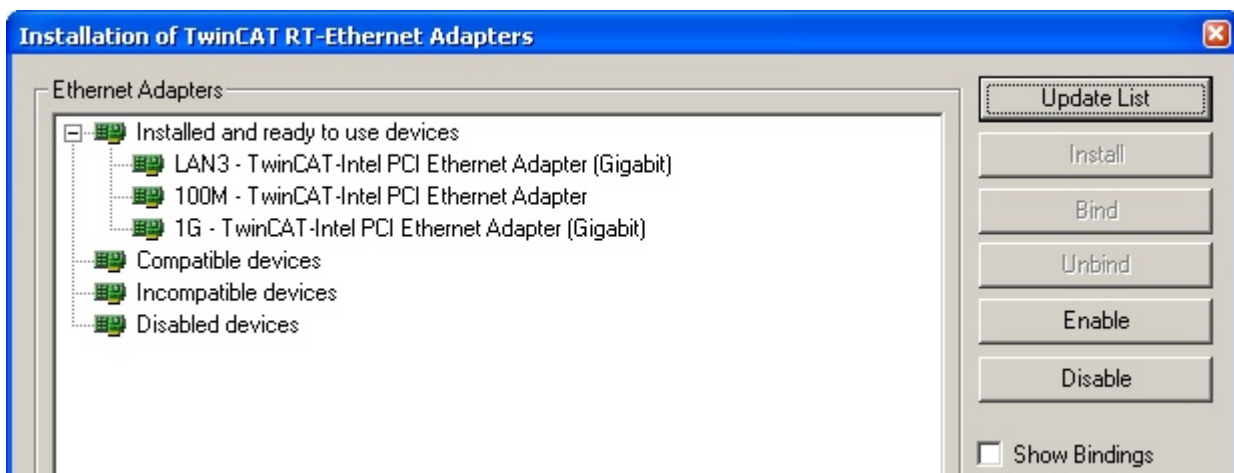


Fig. 88: Overview of network interfaces

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" \[► 90\]](#) in order to view the compatible ethernet ports via its EtherCAT properties (tab „Adapter“, button „Compatible Devices...“):



Fig. 89: 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)

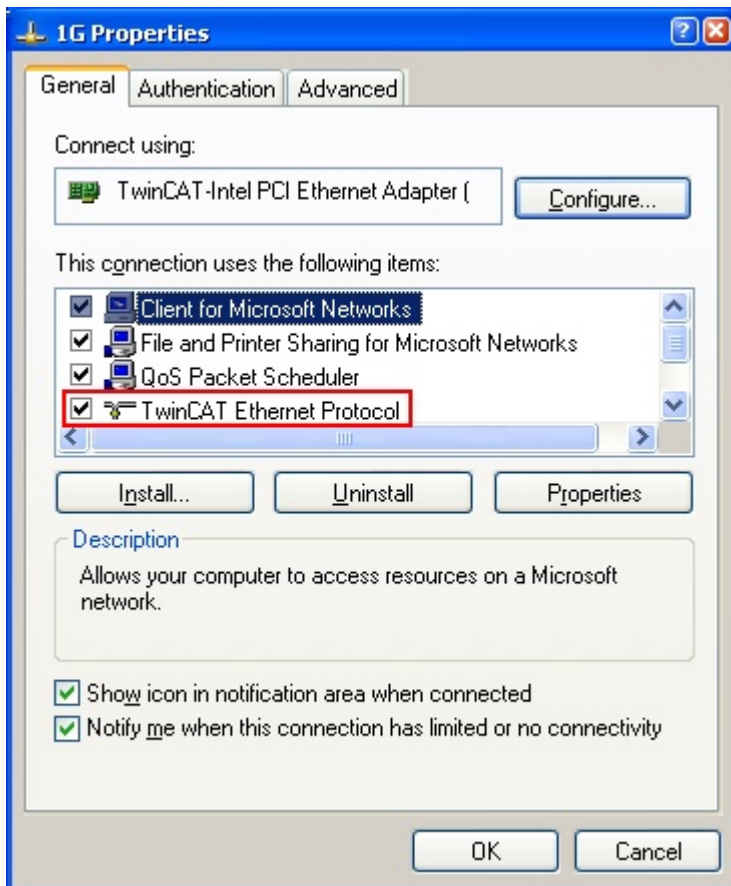


Fig. 90: Windows properties of the network interface

A correct setting of the driver could be:

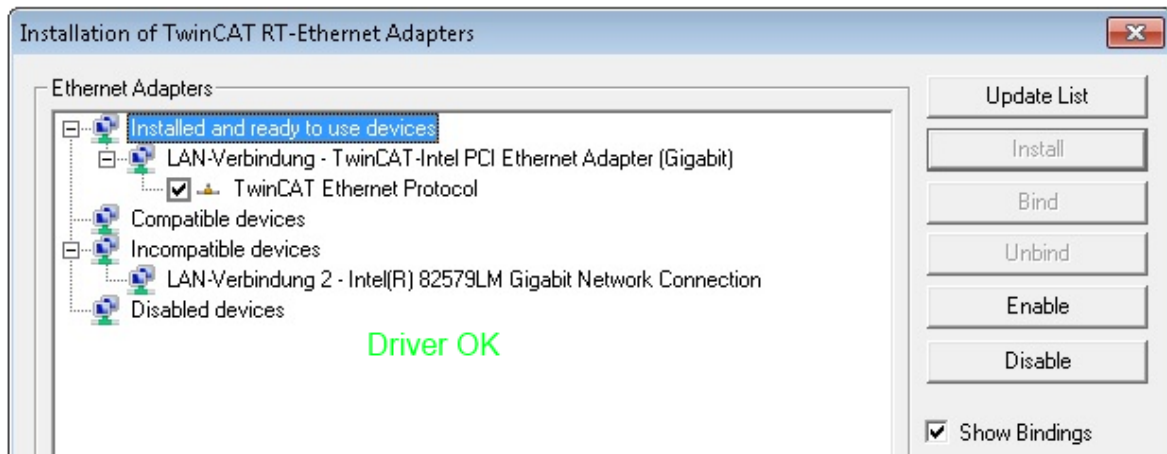


Fig. 91: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

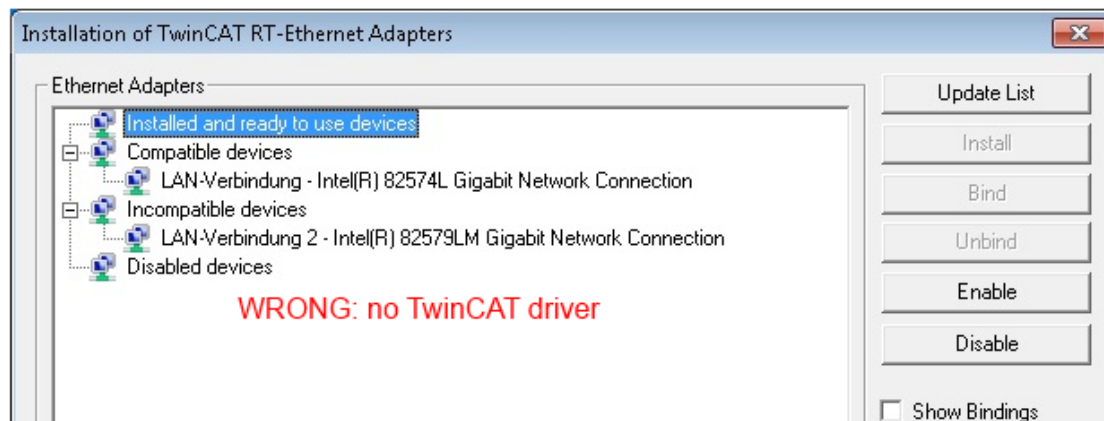
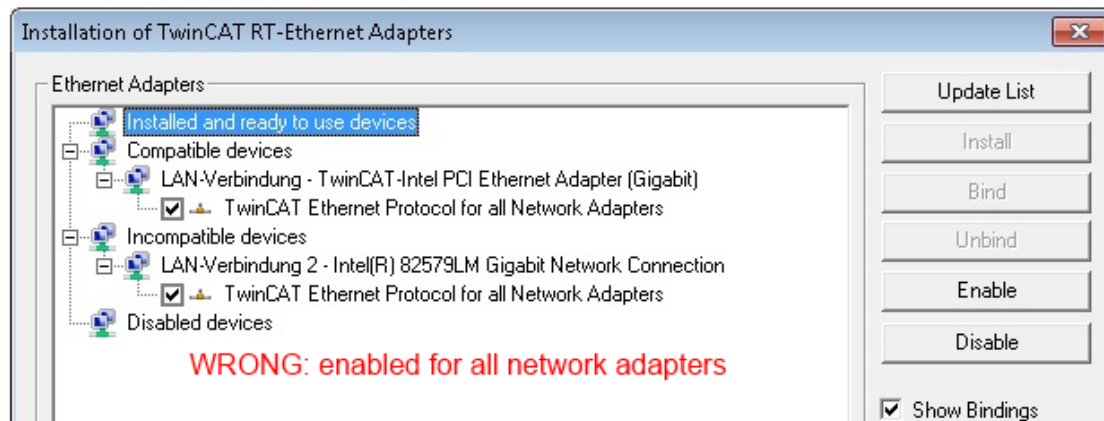
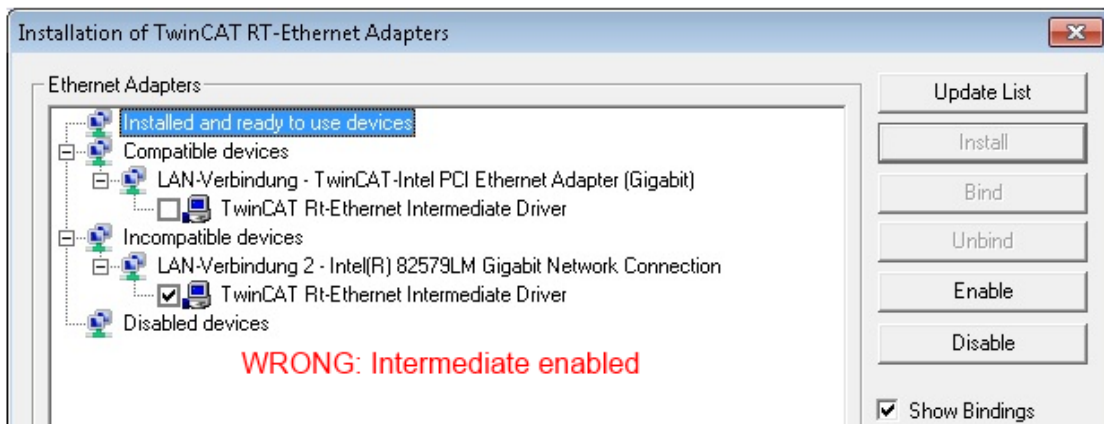
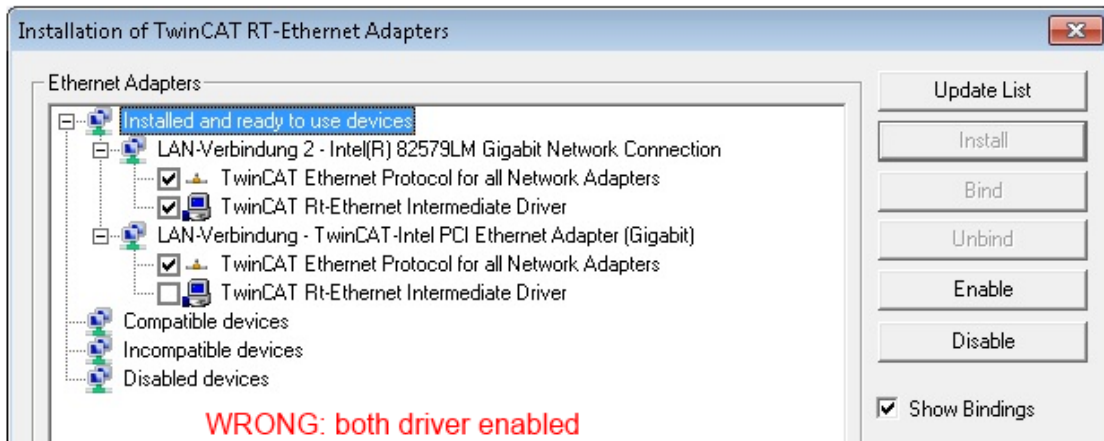


Fig. 92: Incorrect driver settings for the Ethernet port

IP address of the port used

● IP address/DHCP

i 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.

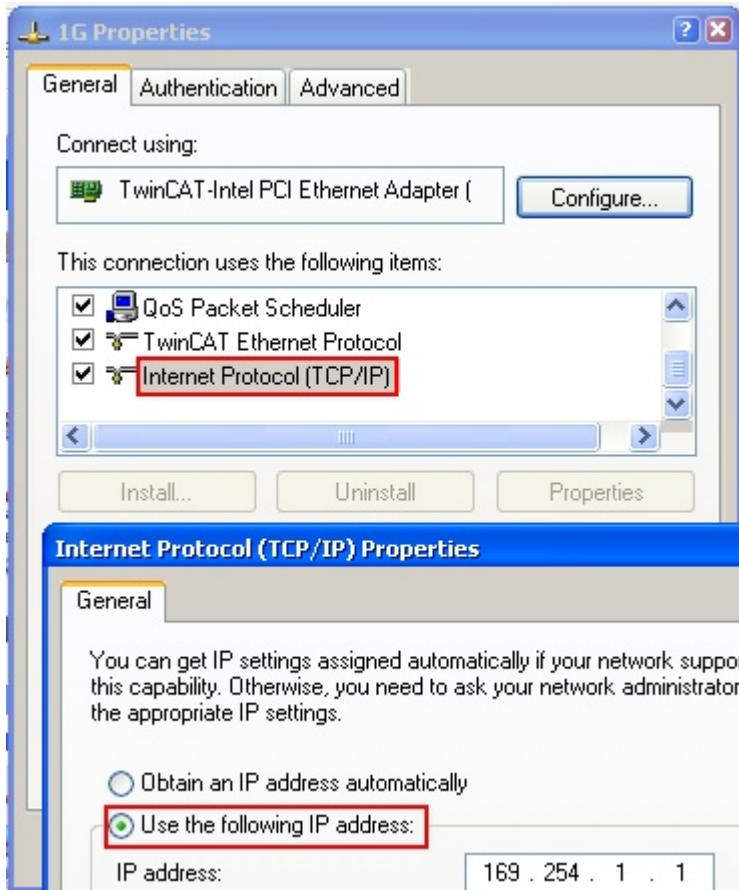


Fig. 93: TCP/IP setting for the Ethernet port

6.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

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

The ESI files for Beckhoff EtherCAT devices are available on the [Beckhoff website](#).

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

Default settings:

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

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

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

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

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

The [TwinCAT ESI Updater \[► 89\]](#) 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”

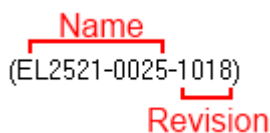


Fig. 94: Identifier structure

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 \[► 9\]](#).

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.

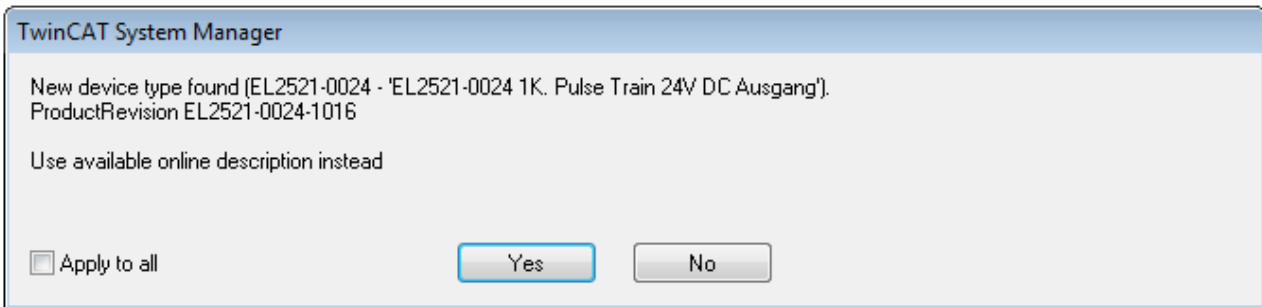


Fig. 95: *OnlineDescription information window (TwinCAT 2)*

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

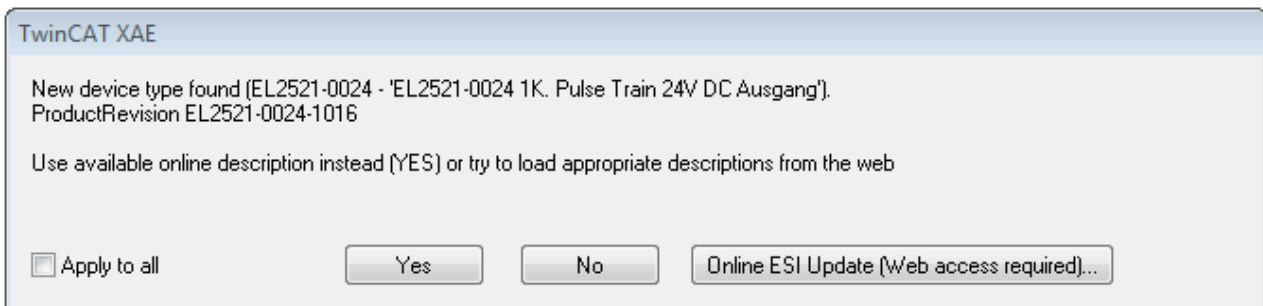


Fig. 96: *Information window OnlineDescription (TwinCAT 3)*

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](#)' [[▶ 90](#)].

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. 97: 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").

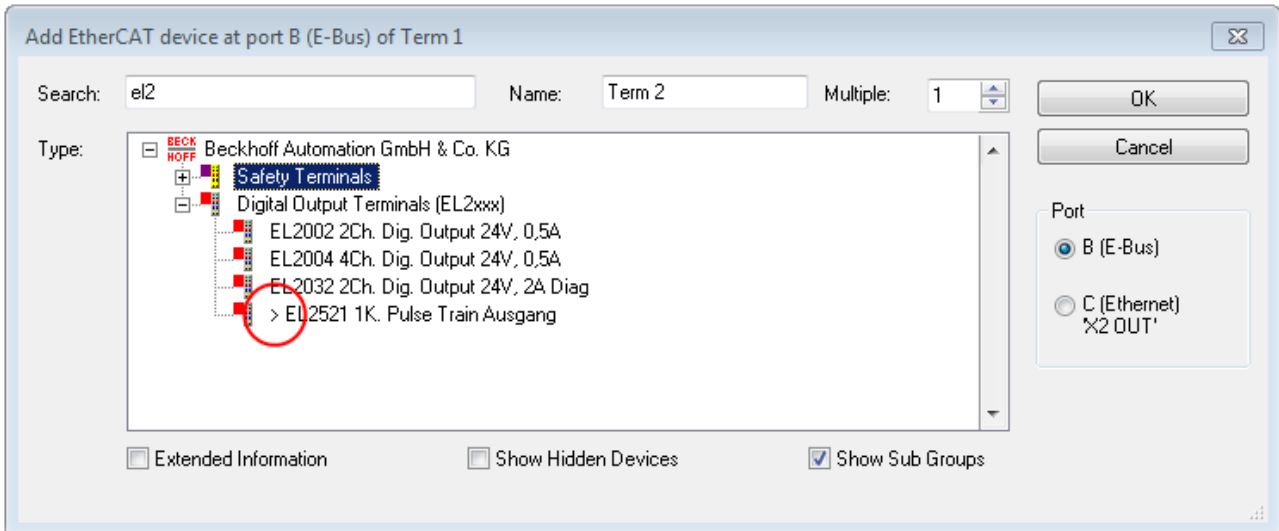


Fig. 98: Indication of an online recorded ESI of EL2521 as an example

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.

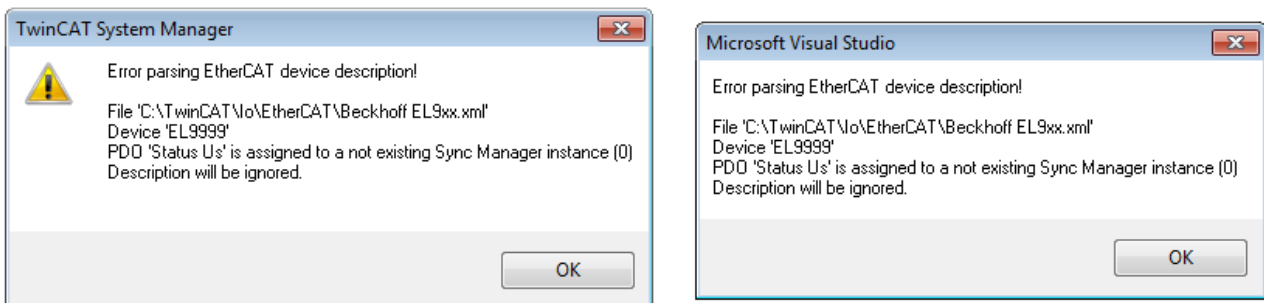


Fig. 99: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

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

6.2.3 TwinCAT ESI Updater

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

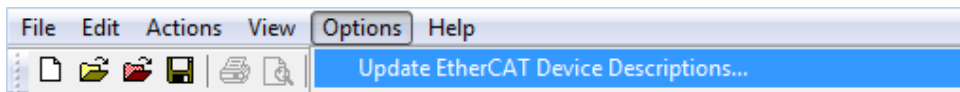


Fig. 100: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

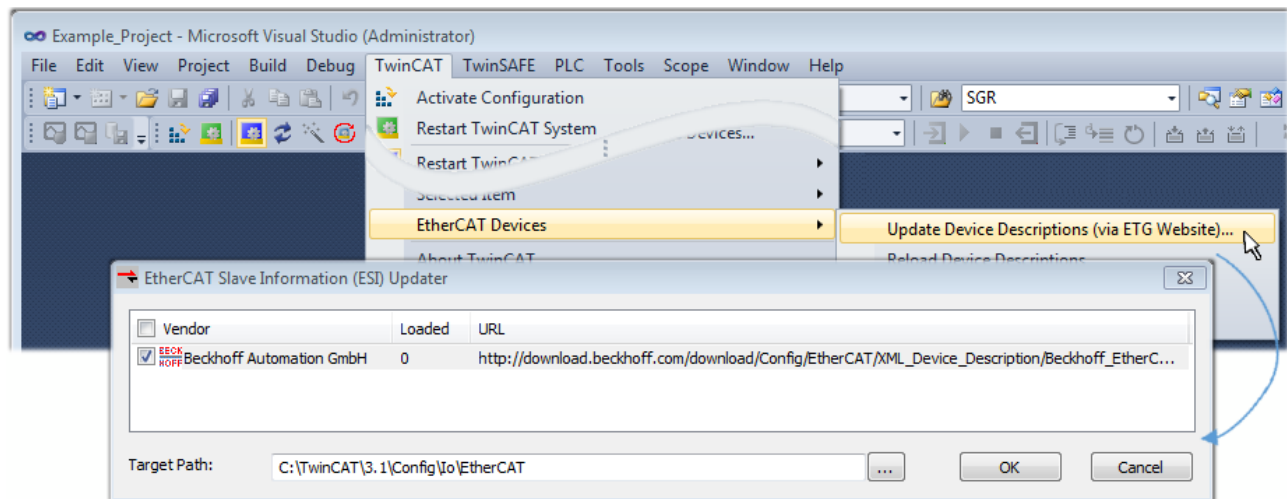


Fig. 101: Using the ESI Updater (TwinCAT 3)

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

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

6.2.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to note "Installation of the latest ESI-XML device description" [▶ 85].

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 [▶ 95] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 96]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 99]

The scan with existing configuration [▶ 100] can also be carried out for comparison.

6.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

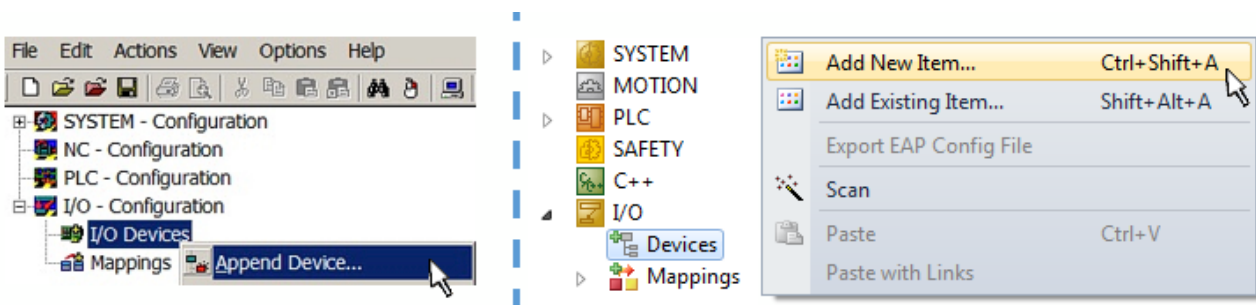


Fig. 102: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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".

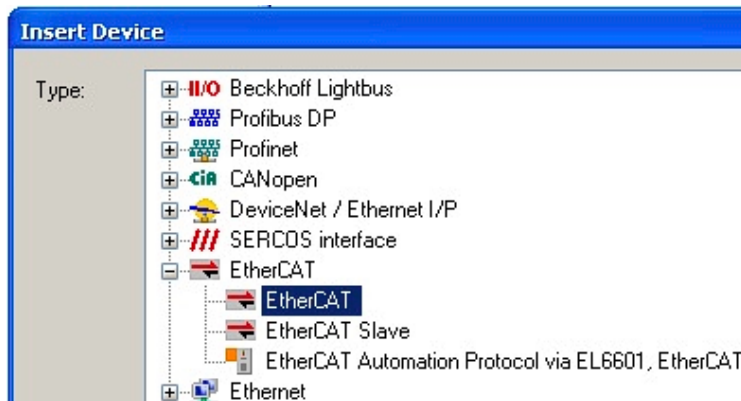


Fig. 103: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

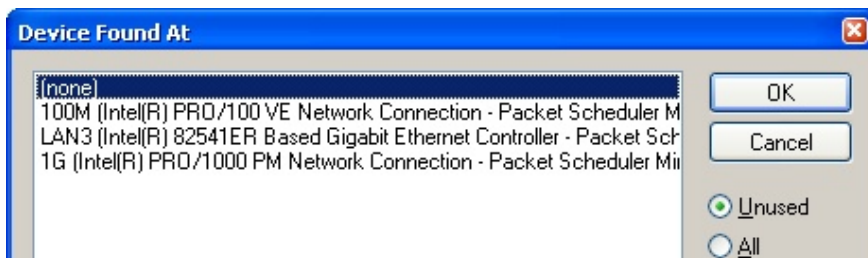


Fig. 104: Selecting the Ethernet port

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)”.

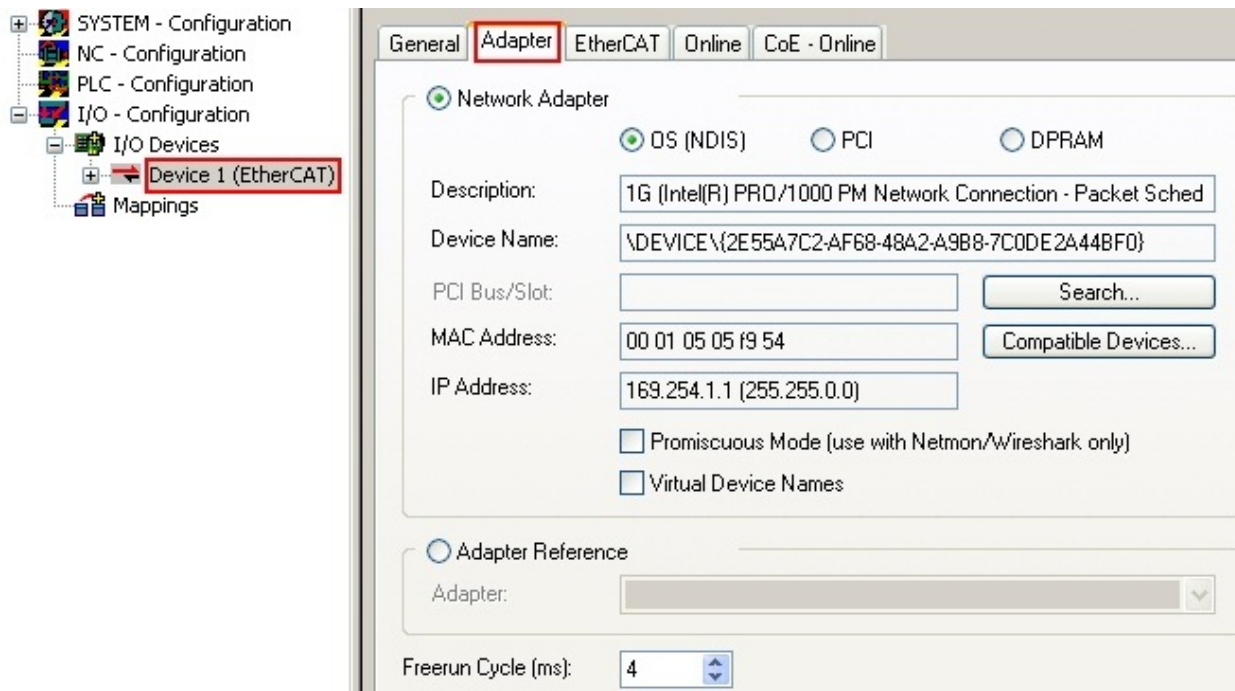
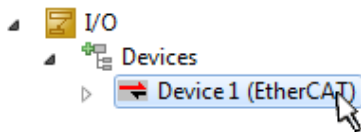


Fig. 105: EtherCAT device properties (TwinCAT 2)

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 \[p. 79\]](#).

Defining EtherCAT slaves

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

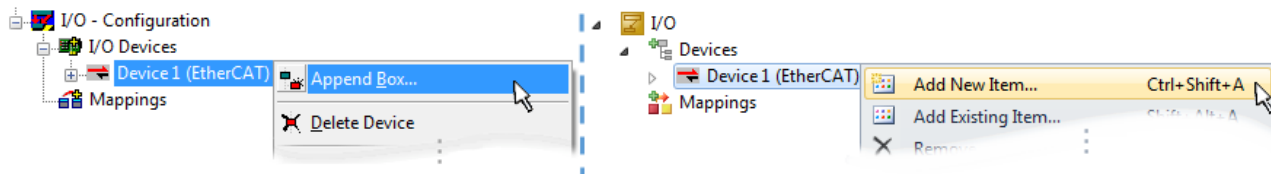


Fig. 106: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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).

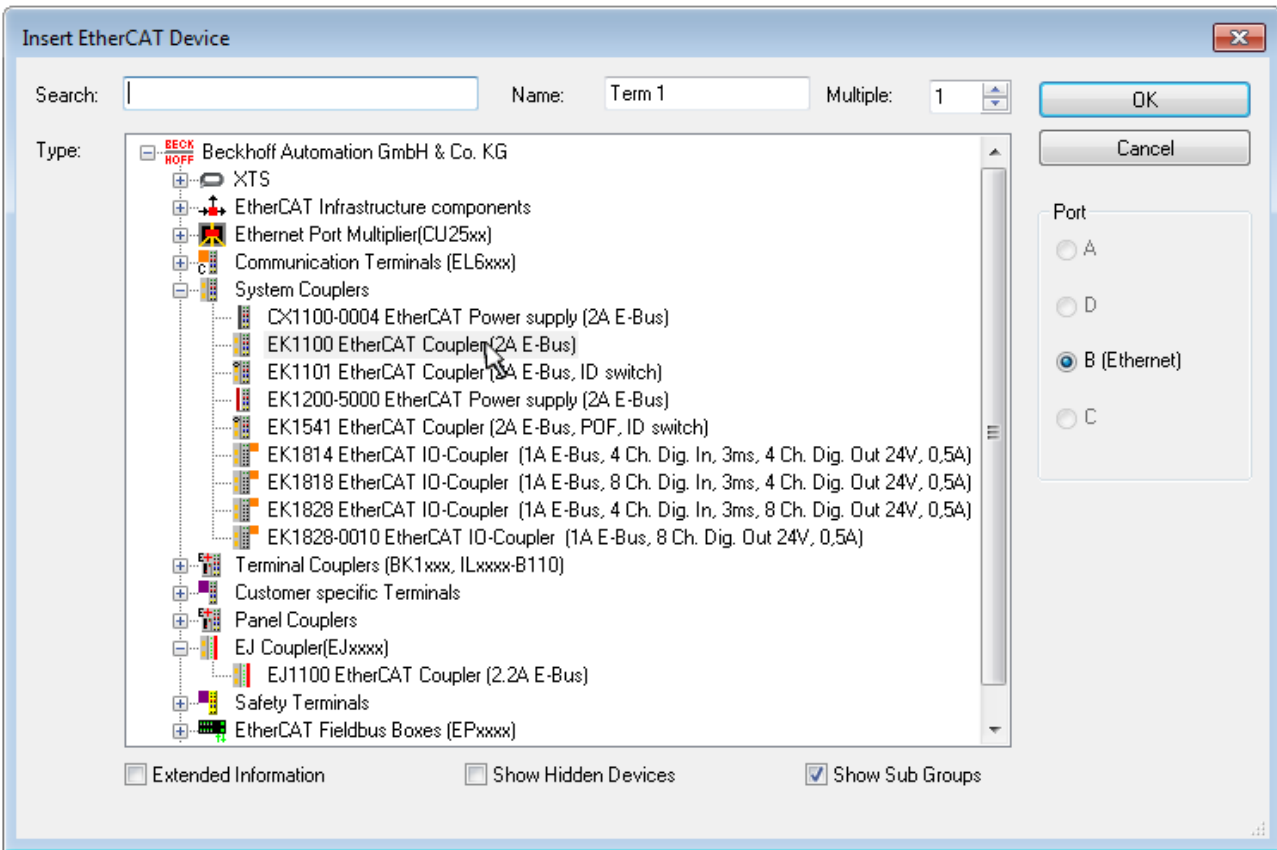


Fig. 107: Selection dialog for new EtherCAT device

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”.

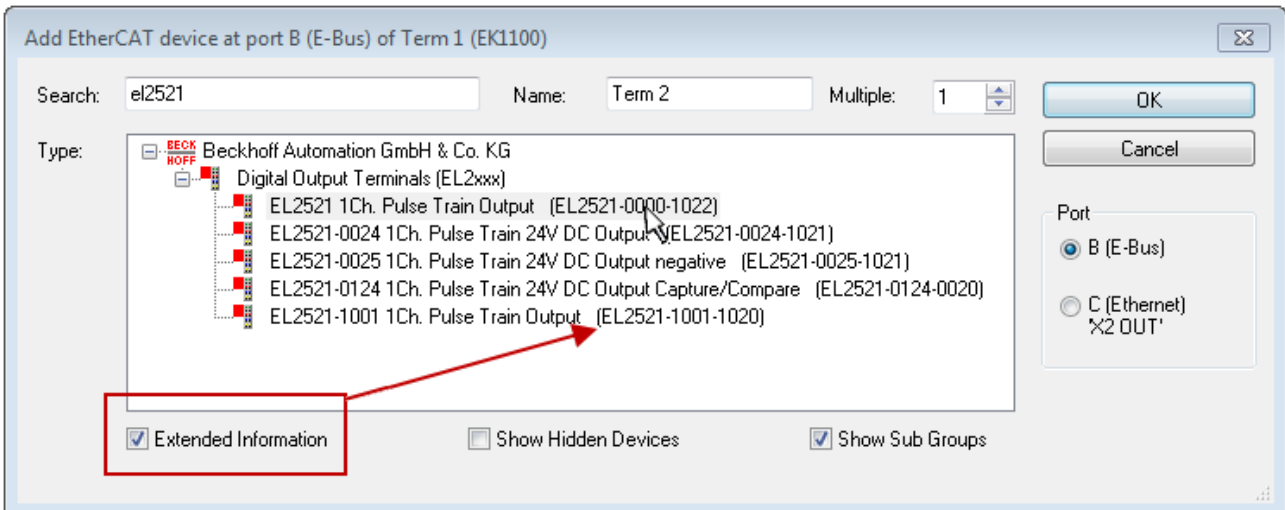


Fig. 108: Display of device revision

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”.

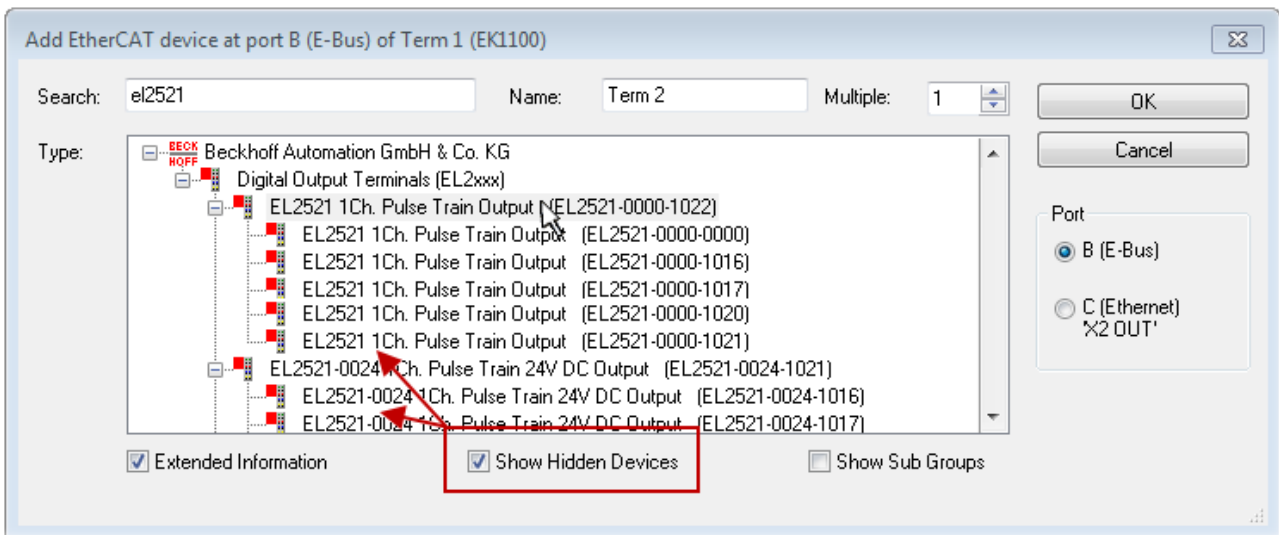


Fig. 109: Display of previous revisions

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.

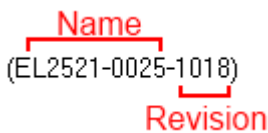


Fig. 110: Name/revision of the terminal

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

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

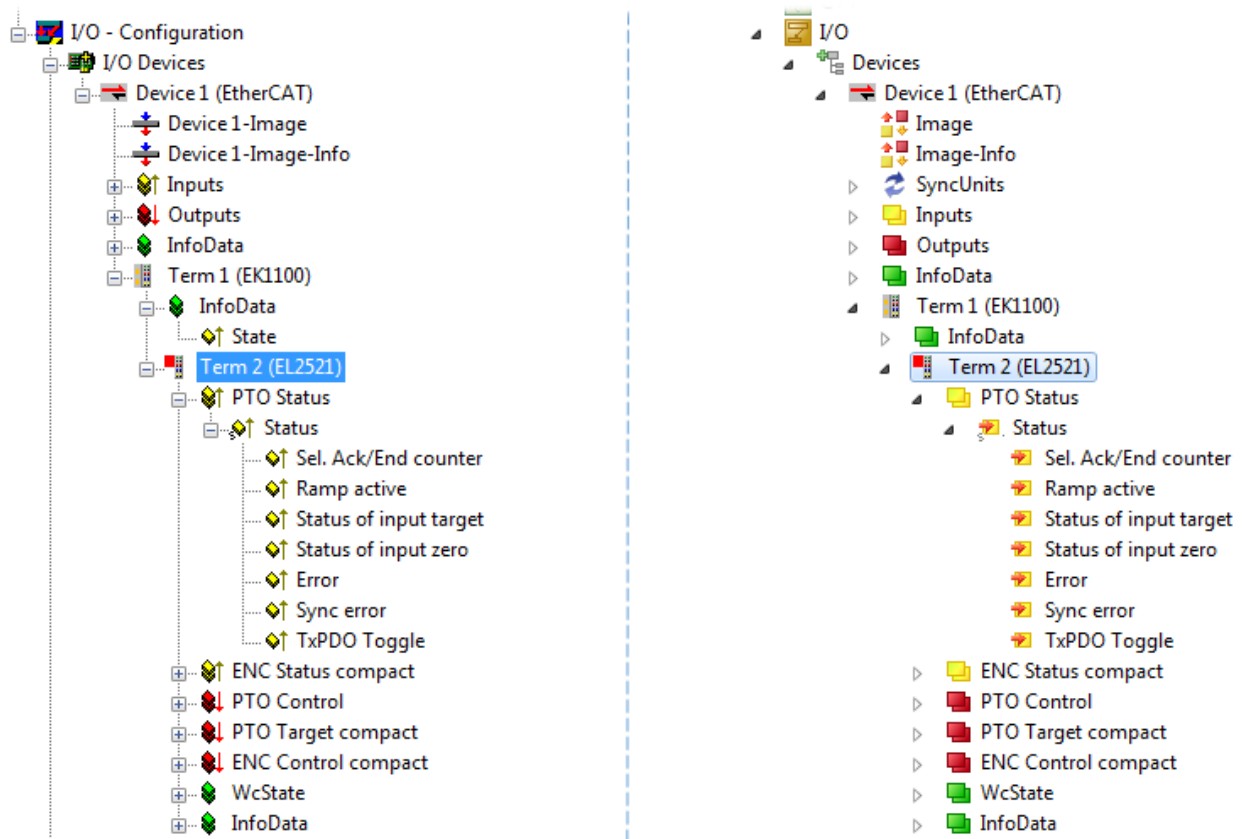




Fig. 111: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



6.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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



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

TwinCAT can be set into this mode:

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

Online scanning in Config mode

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

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

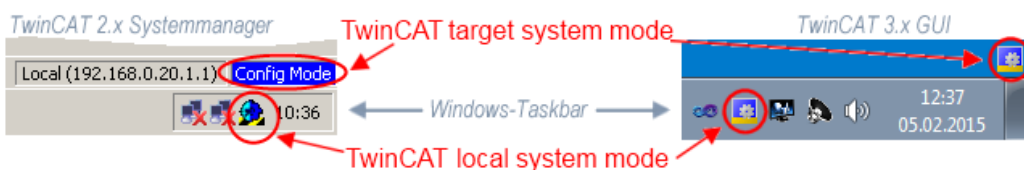


Fig. 112: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

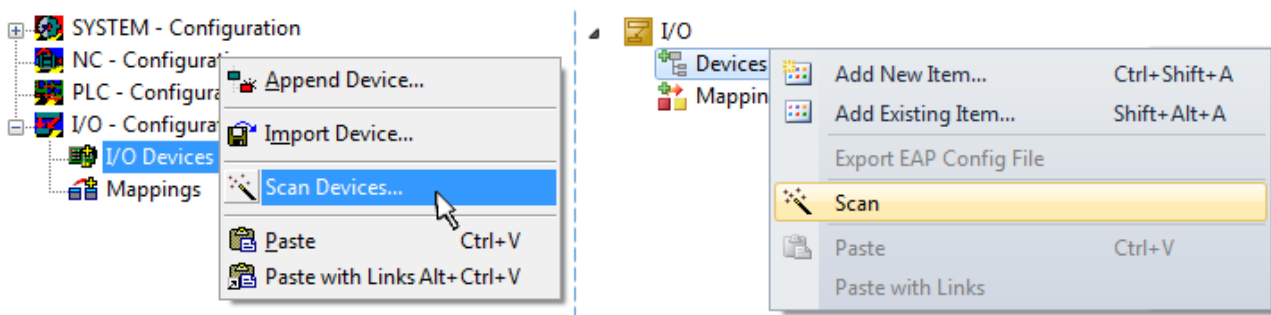


Fig. 113: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

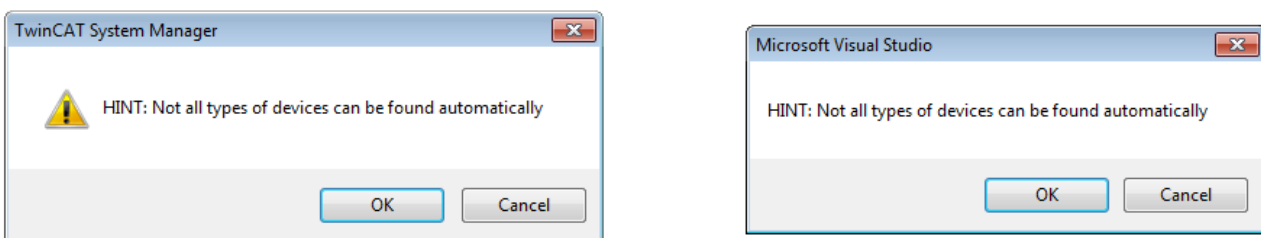


Fig. 114: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

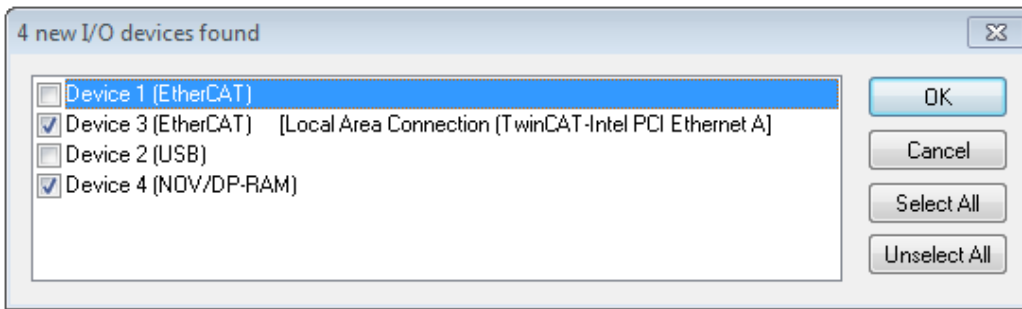


Fig. 115: *Detected Ethernet devices*

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](#) [▶ 79].

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.

Name
(EL2521-0025-1018)
Revision

Fig. 116: *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](#) [▶ 100] 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:

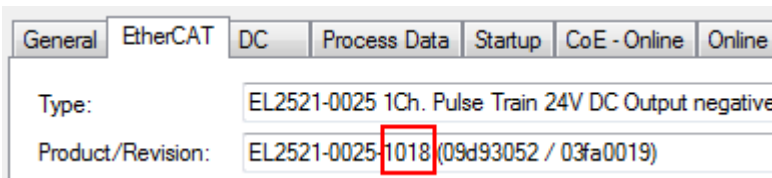


Fig. 117: Installing EtherCAT terminal with revision -1018

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 [▶ 100] 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:

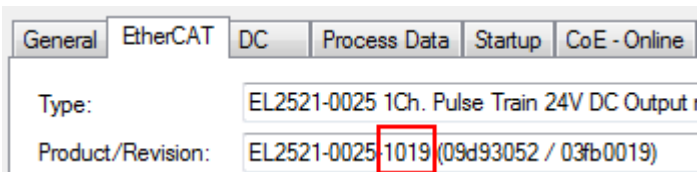


Fig. 118: Detection of EtherCAT terminal with revision -1019

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

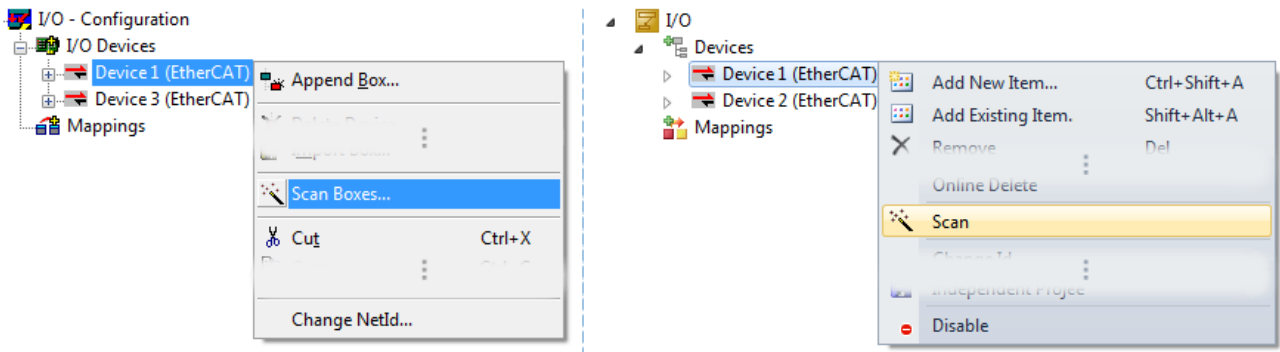


Fig. 120: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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. 121: Scan progress exemplary by TwinCAT 2

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



Fig. 122: 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. 123: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 124: 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”.

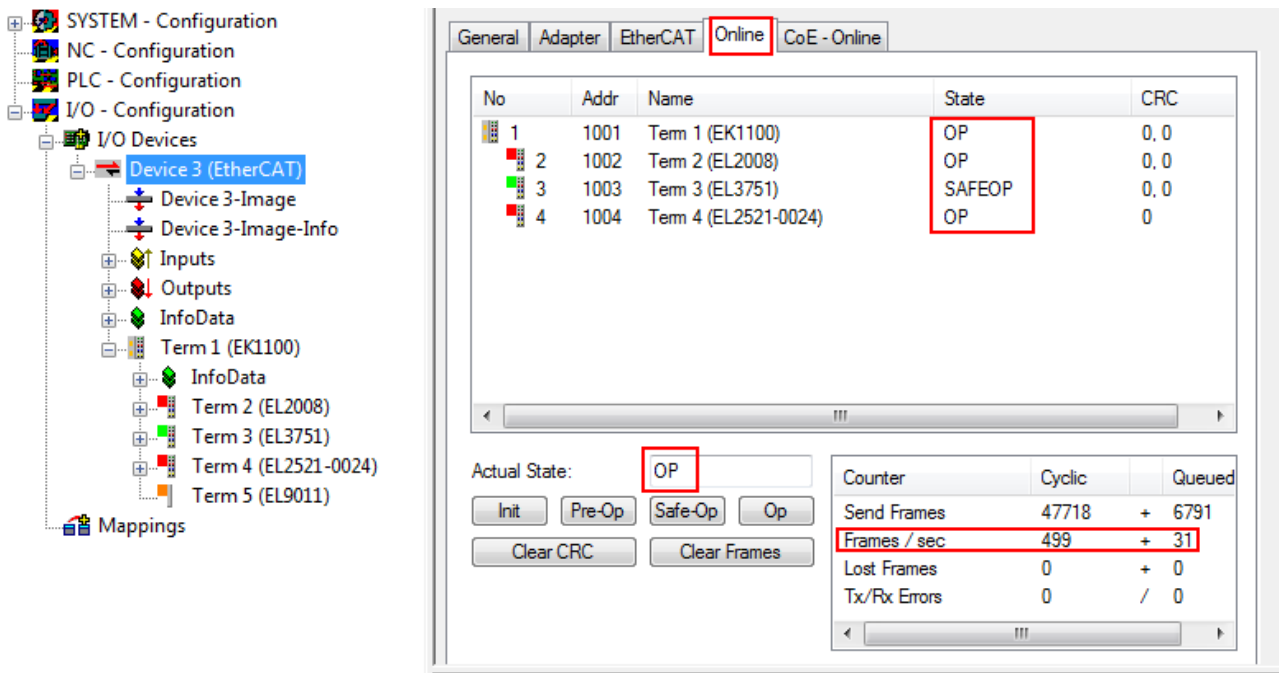


Fig. 125: Online display example

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 \[► 90\]](#).

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.

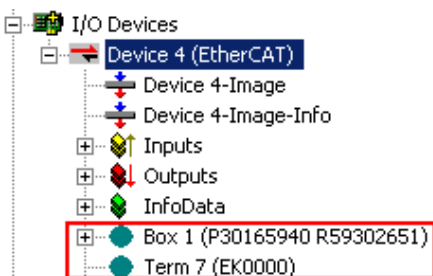


Fig. 126: Faulty identification

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. 127: 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.

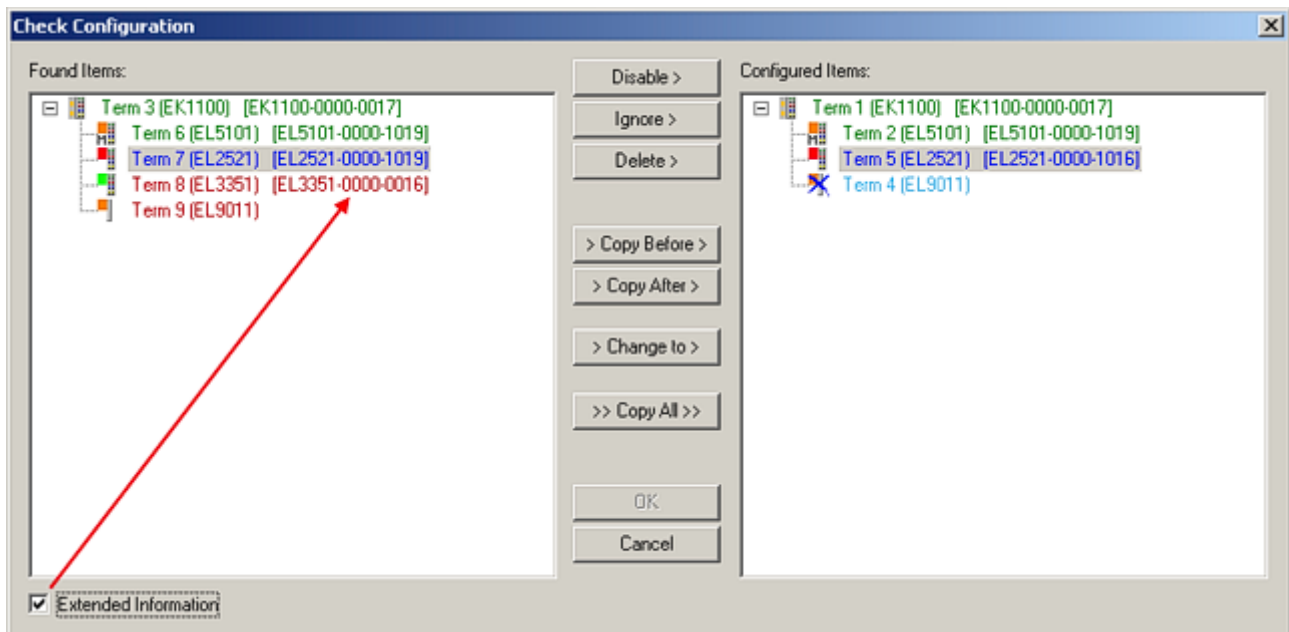


Fig. 128: Correction dialog

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. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

● Device selection based on revision, compatibility

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 >= 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.

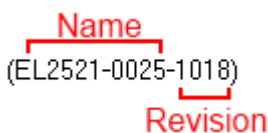


Fig. 129: Name/revision of the terminal

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, ...

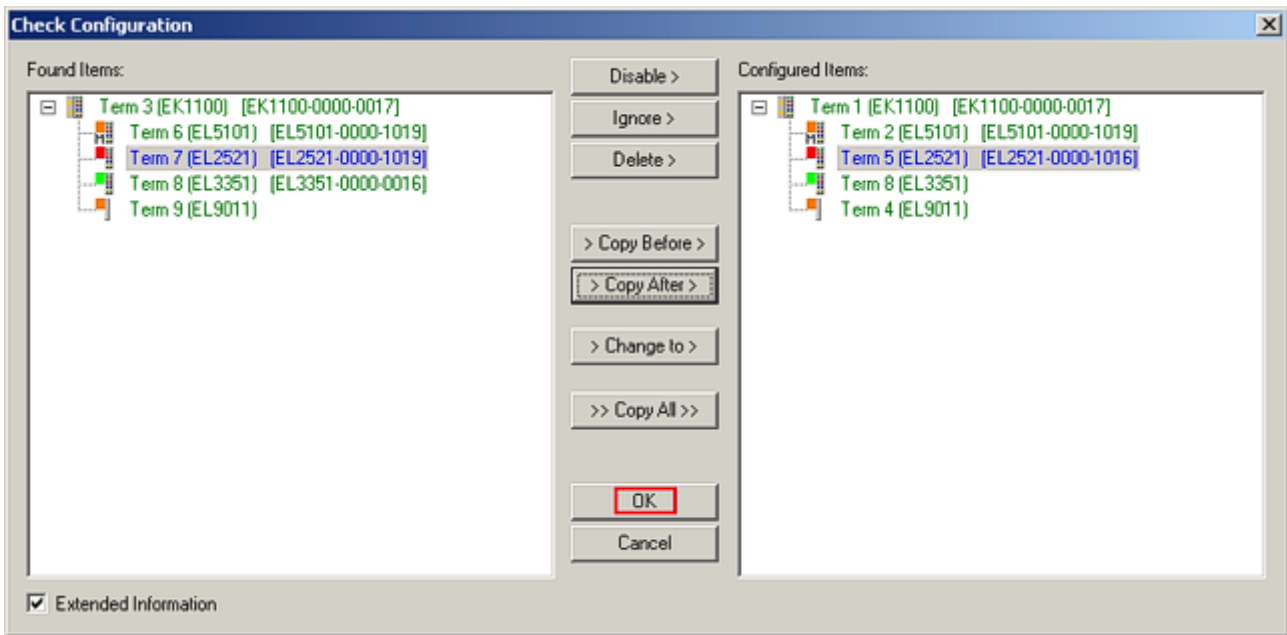


Fig. 130: Correction dialog with modifications

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.

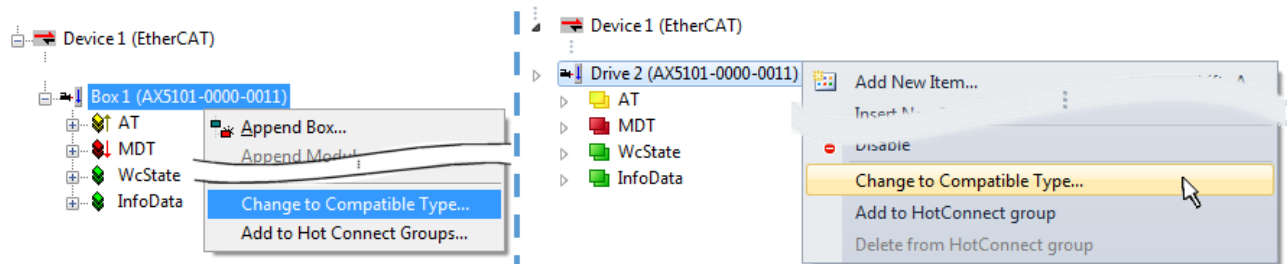


Fig. 131: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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*

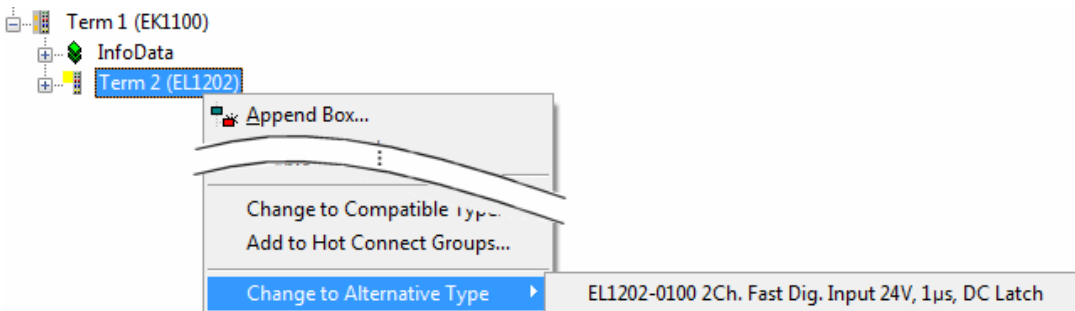


Fig. 132: TwinCAT 2 Dialog Change to Alternative Type

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

6.2.7 EtherCAT subscriber configuration

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

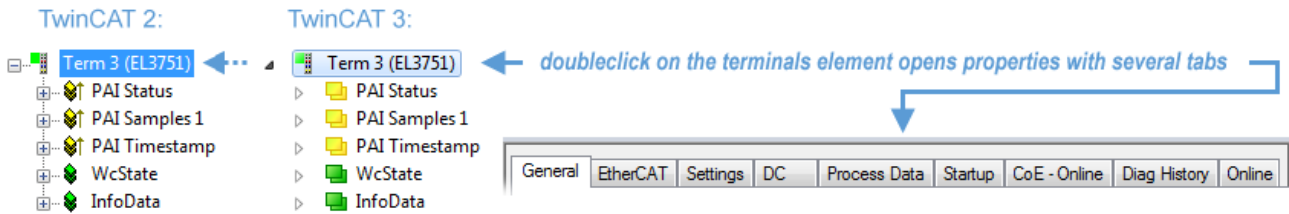


Fig. 133: Branch element as terminal EL3751

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

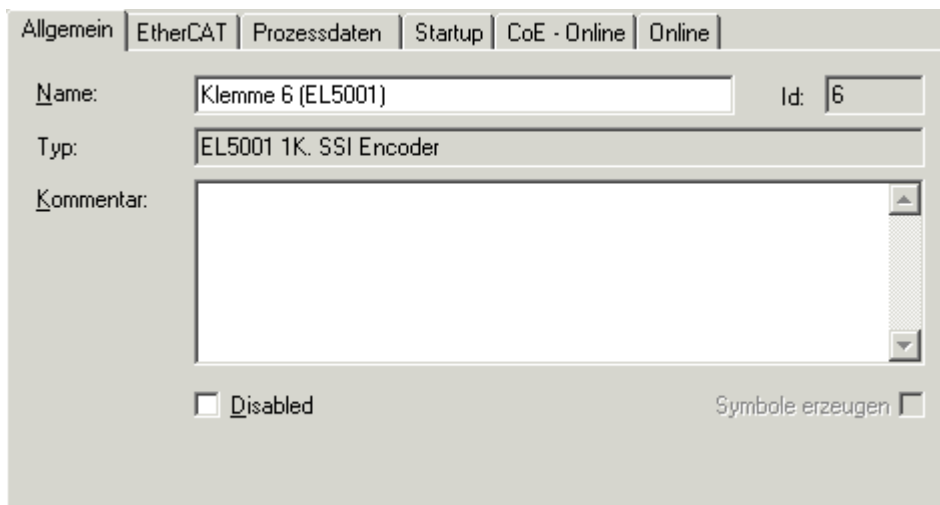


Fig. 134: "General" tab

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. 135: „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 ($FFFF_{\text{hex}}$, $FFFE_{\text{hex}}$ etc.).

EtherCAT Addr.

Fixed address of an EtherCAT slave. This address is allocated by the EtherCAT master during the start-up phase. Tick the control box to the left of the input field in order to modify the default value.

Previous Port

Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this device is to be connected can be selected.

Advanced Settings

This button opens the dialogs for advanced settings.

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

“Process Data” tab

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

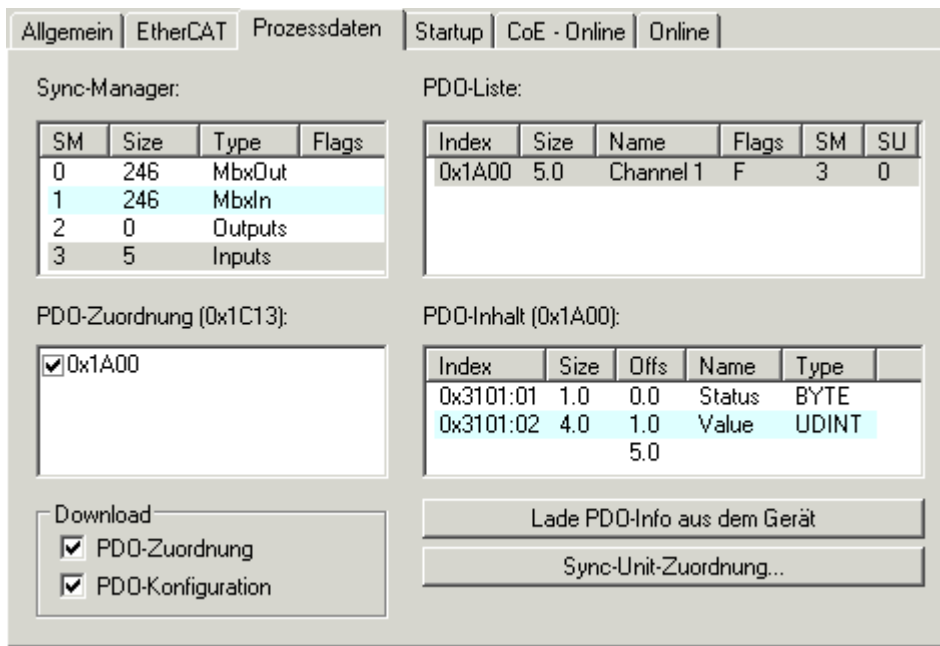


Fig. 136: "Process Data" tab

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").

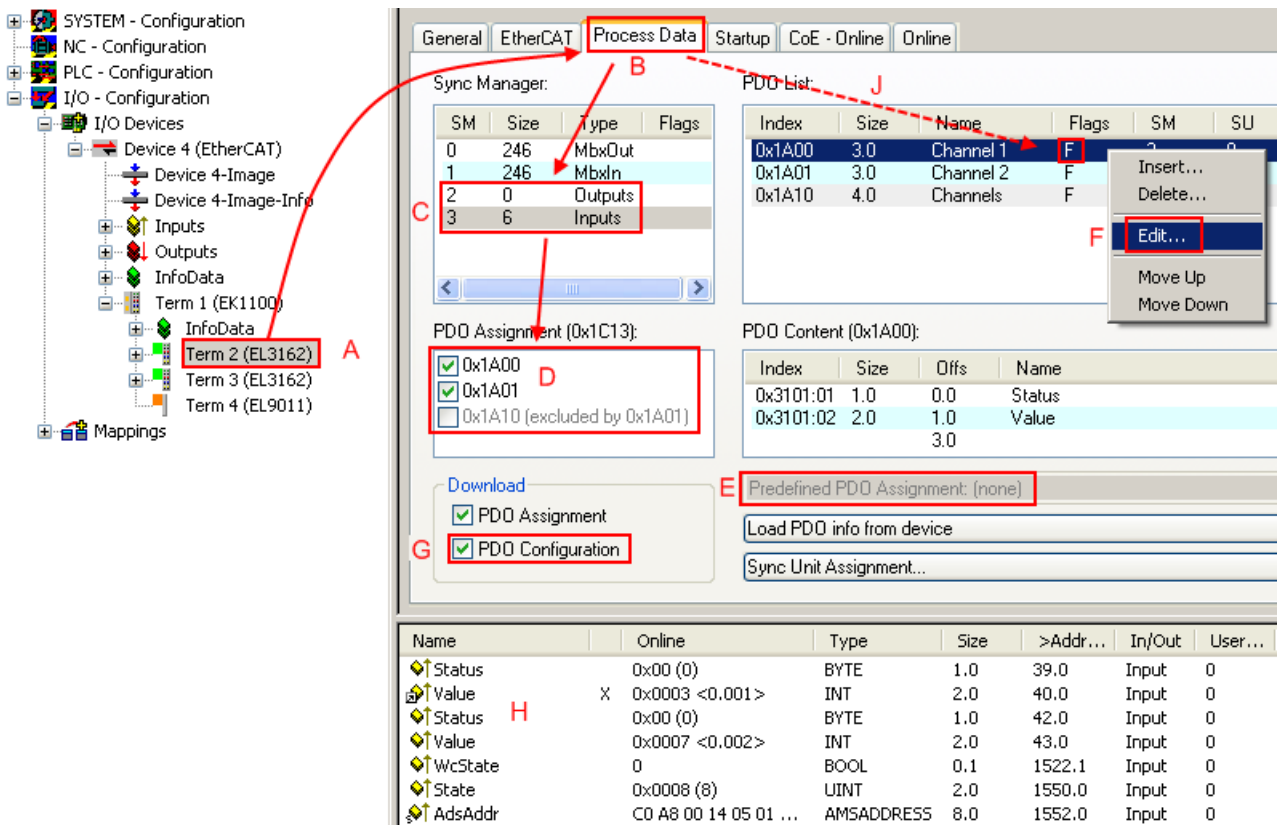


Fig. 137: Configuring the process data

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 [► 111] 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.

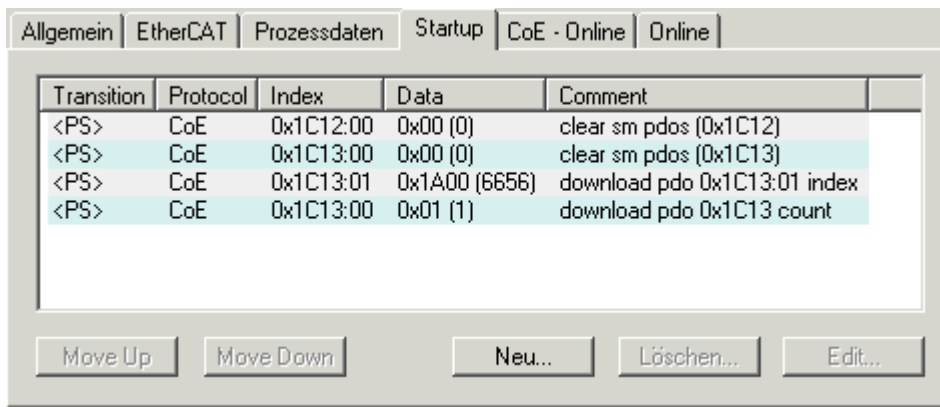


Fig. 138: „Startup“ tab

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.

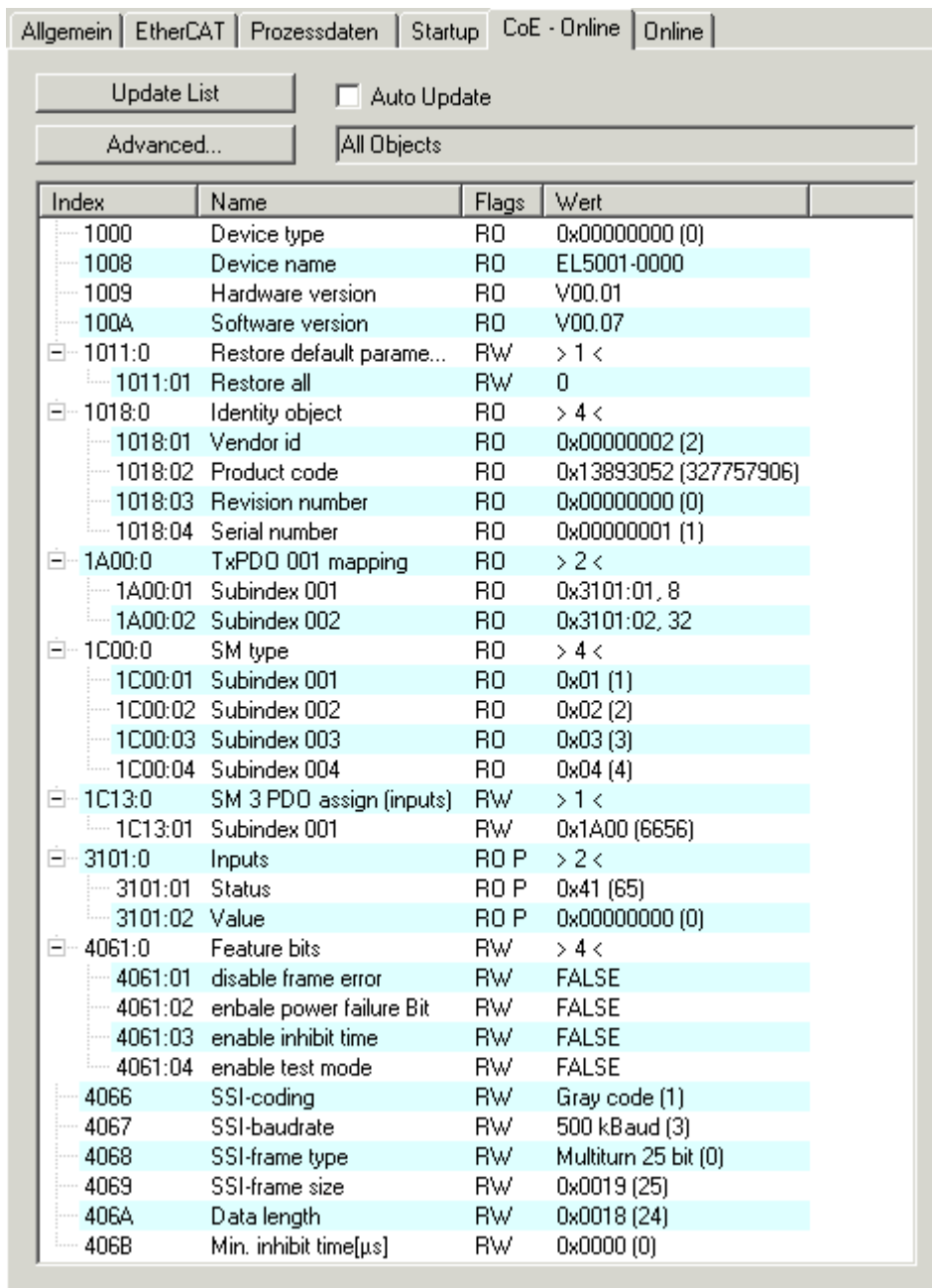


Fig. 139: "CoE – Online" tab

Object list display

Column	Description						
Index	Index and sub-index of the object						
Name	Name of the object						
Flags	<table border="1"> <tr> <td>RW</td> <td>The object can be read, and data can be written to the object (read/write)</td> </tr> <tr> <td>RO</td> <td>The object can be read, but no data can be written to the object (read only)</td> </tr> <tr> <td>P</td> <td>An additional P identifies the object as a process data object.</td> </tr> </table>	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.
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.

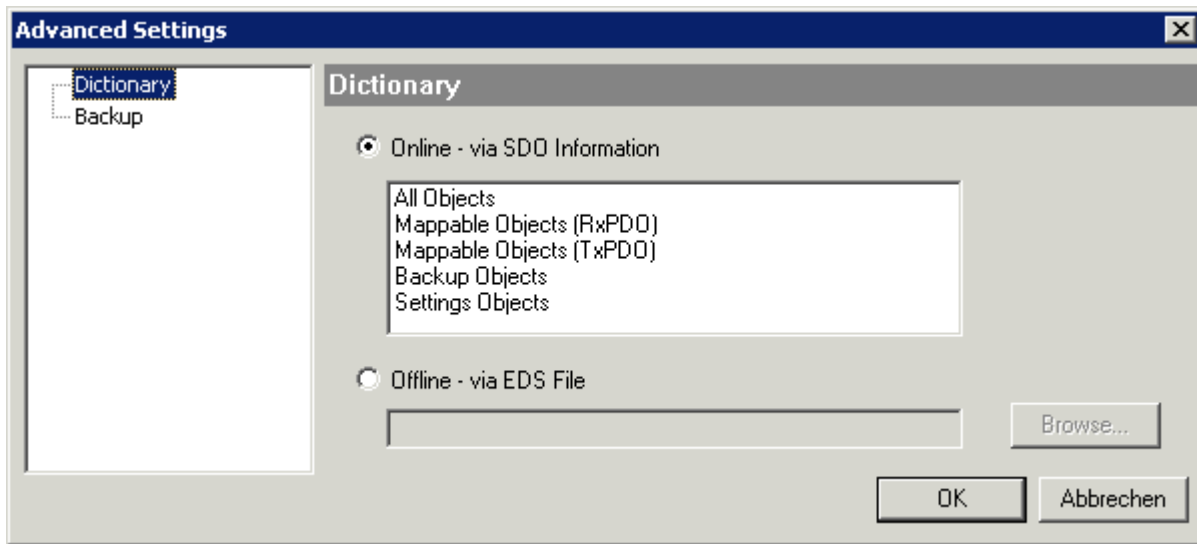


Fig. 140: Dialog “Advanced settings”

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

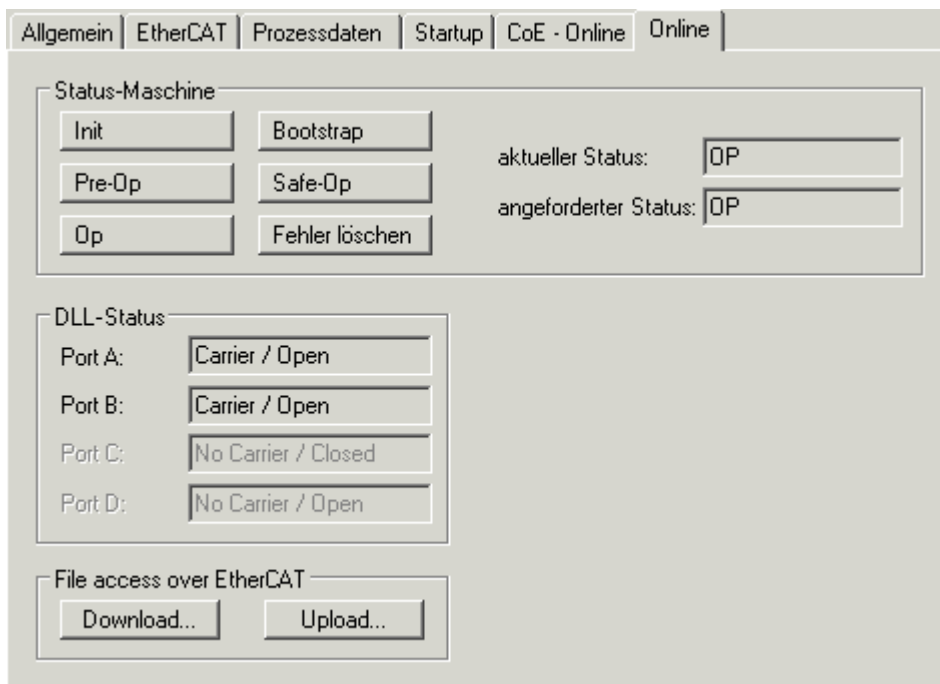


Fig. 141: „Online“ tab

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)

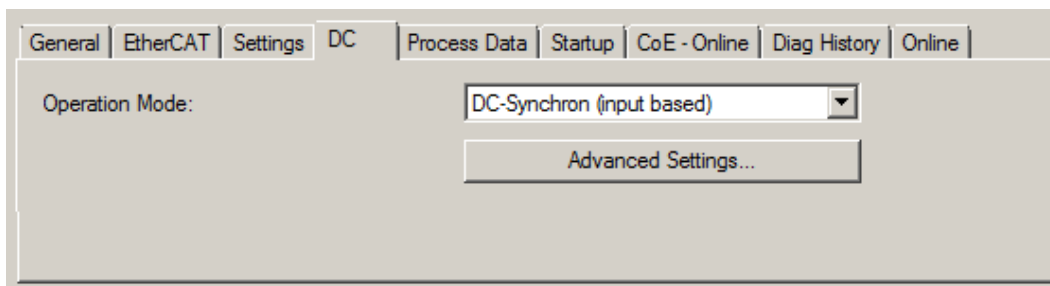


Fig. 142: "DC" tab (Distributed Clocks)

- 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

6.2.7.1 Detailed description of Process Data tab

Sync Manager

Lists the configuration of the Sync Manager (SM).

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

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

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

PDO Assignment



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

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

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

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 \[▶ 109\]](#)),
 - 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 [▶ 106] tab.

PDO Configuration

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

6.3 General Notes - EtherCAT Slave Application

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

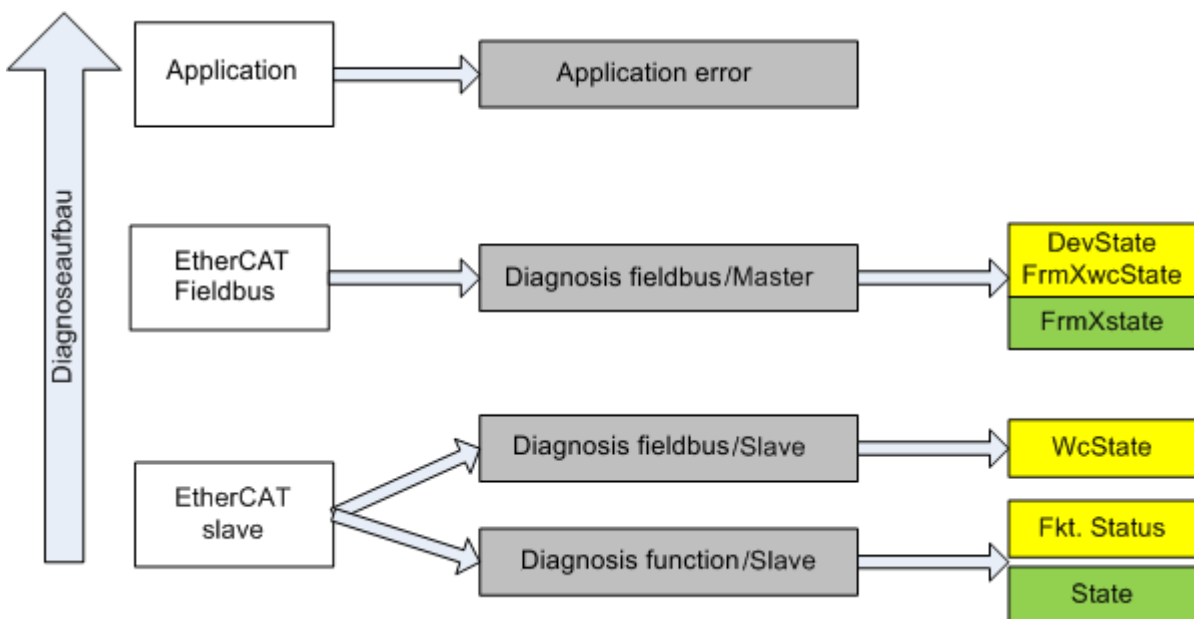


Fig. 143: Selection of the diagnostic information of an EtherCAT Slave

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.

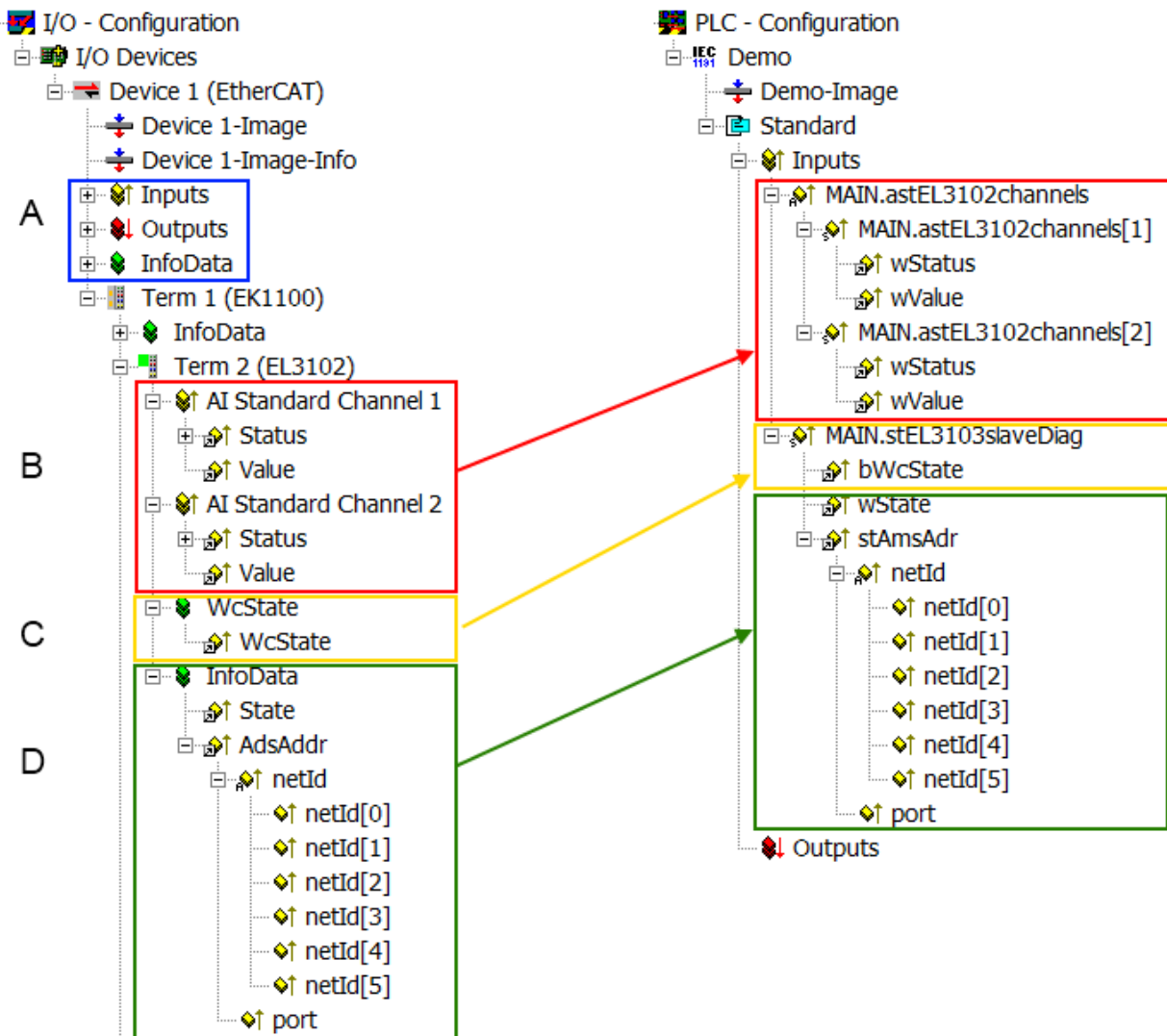


Fig. 144: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

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

NOTE

Diagnostic information

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

CoE Parameter Directory

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

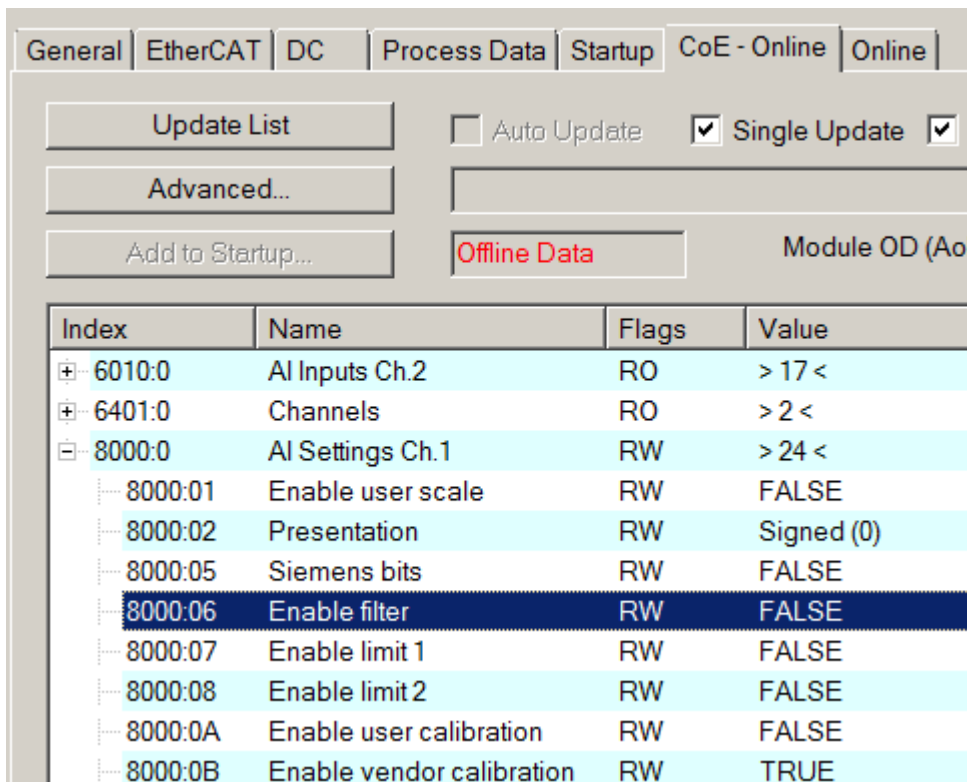


Fig. 145: EL3102, CoE directory

● 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.

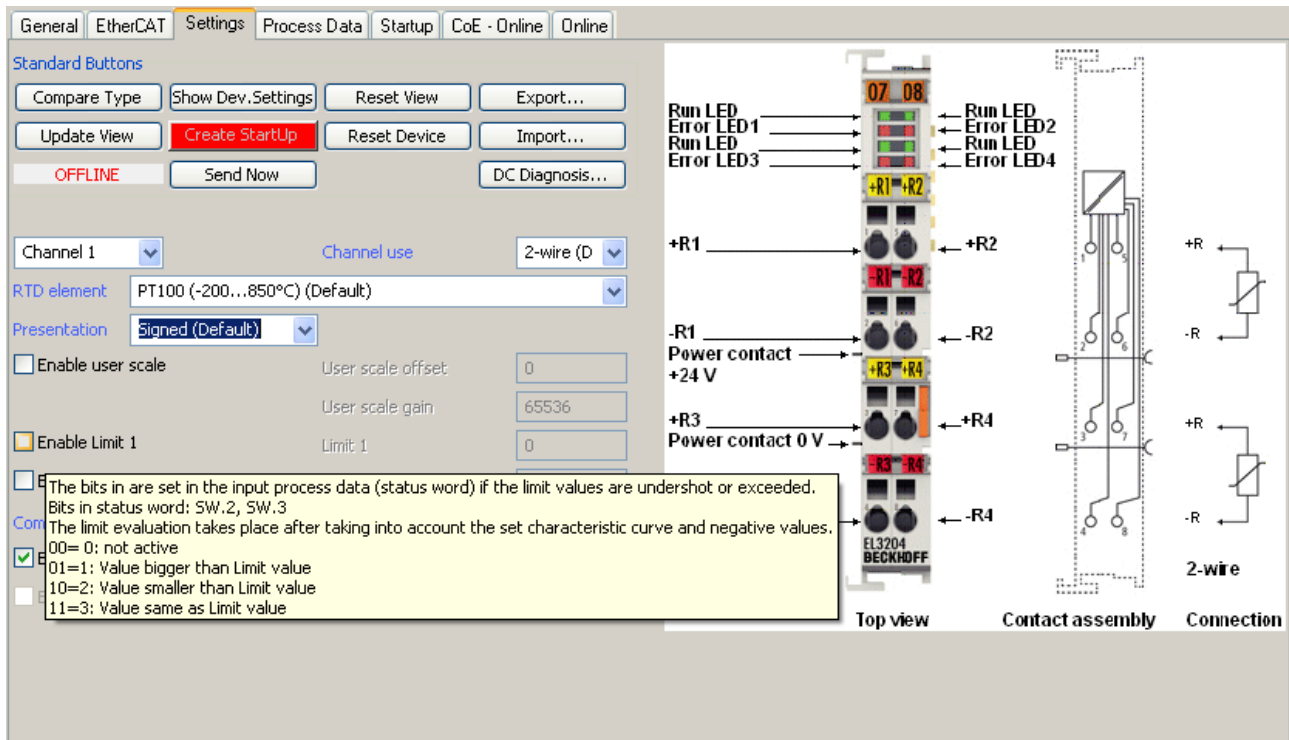


Fig. 146: Example of commissioning aid for a EL3204

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 \[► 30\]](#)" 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.

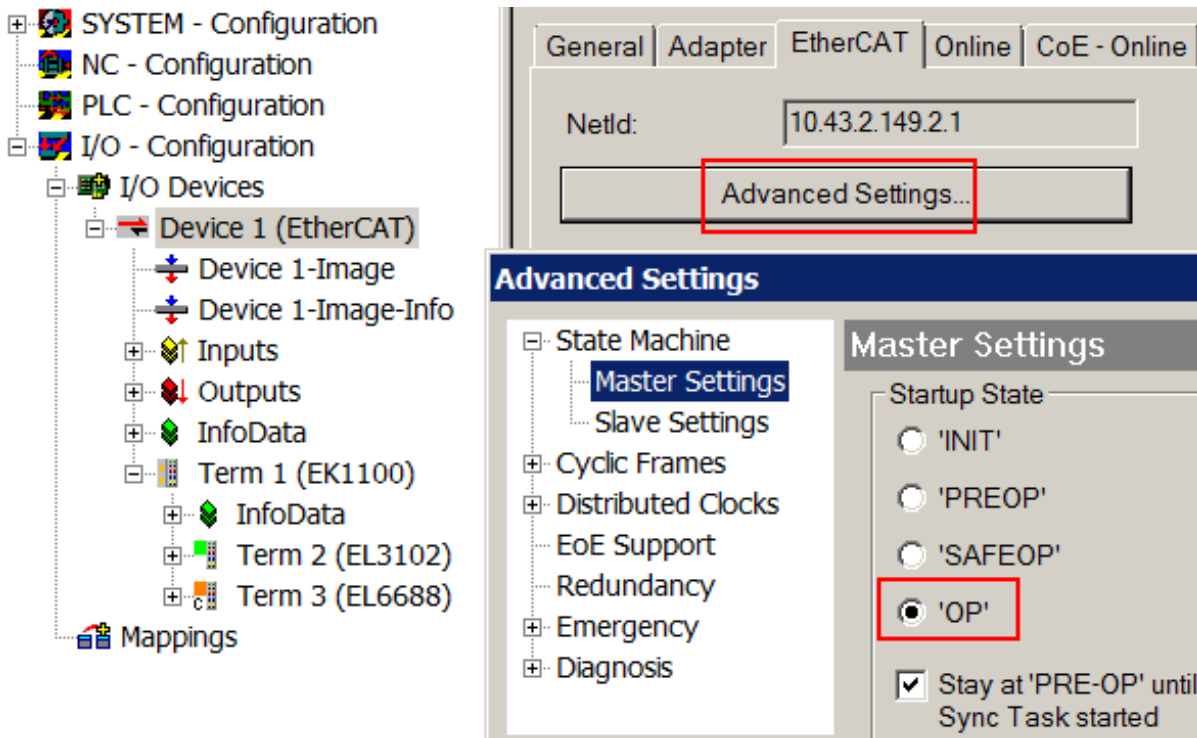


Fig. 147: Default behaviour of the System Manager

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

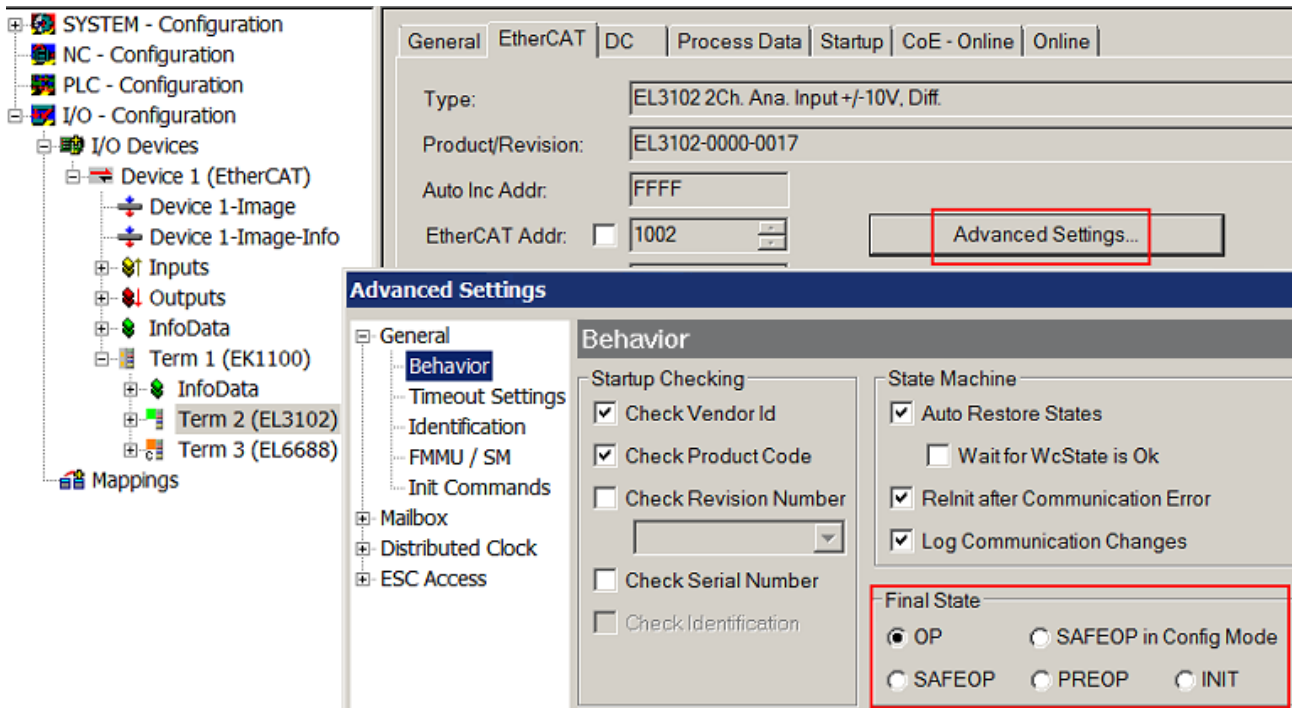


Fig. 148: Default target state in the Slave

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.

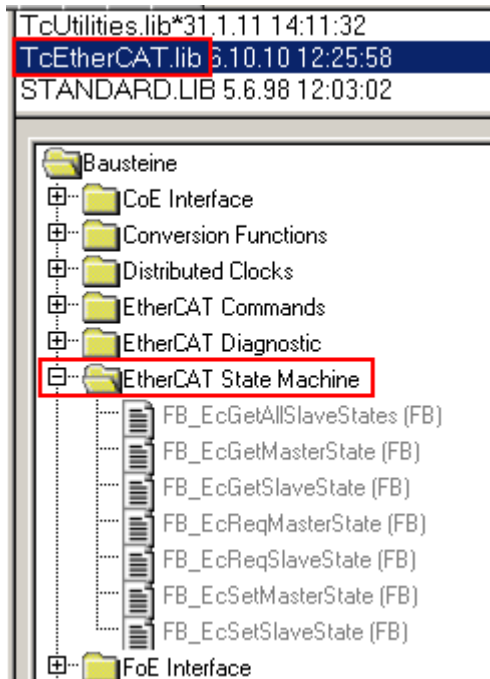


Fig. 149: PLC function blocks

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. 150: 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:

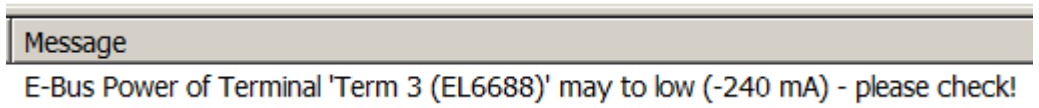


Fig. 151: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

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

6.4 Oversampling terminals and TwinCAT Scope

Generally input data of a terminal 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 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.

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 read of the used ADC (could be a deltaSigma ADC also) digital sample data n-times more than the PLC/ bus cycle time is set and transfers every sample to the control – bundled as an oversampling PDO package.

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

EL3102

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

EL3702

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

Fig. 152: Oversampling PDO of the EL37xx series and in the comparison with EL31xx

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

6.4.1 TwinCAT 3 procedure

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

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

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 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“ have 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).

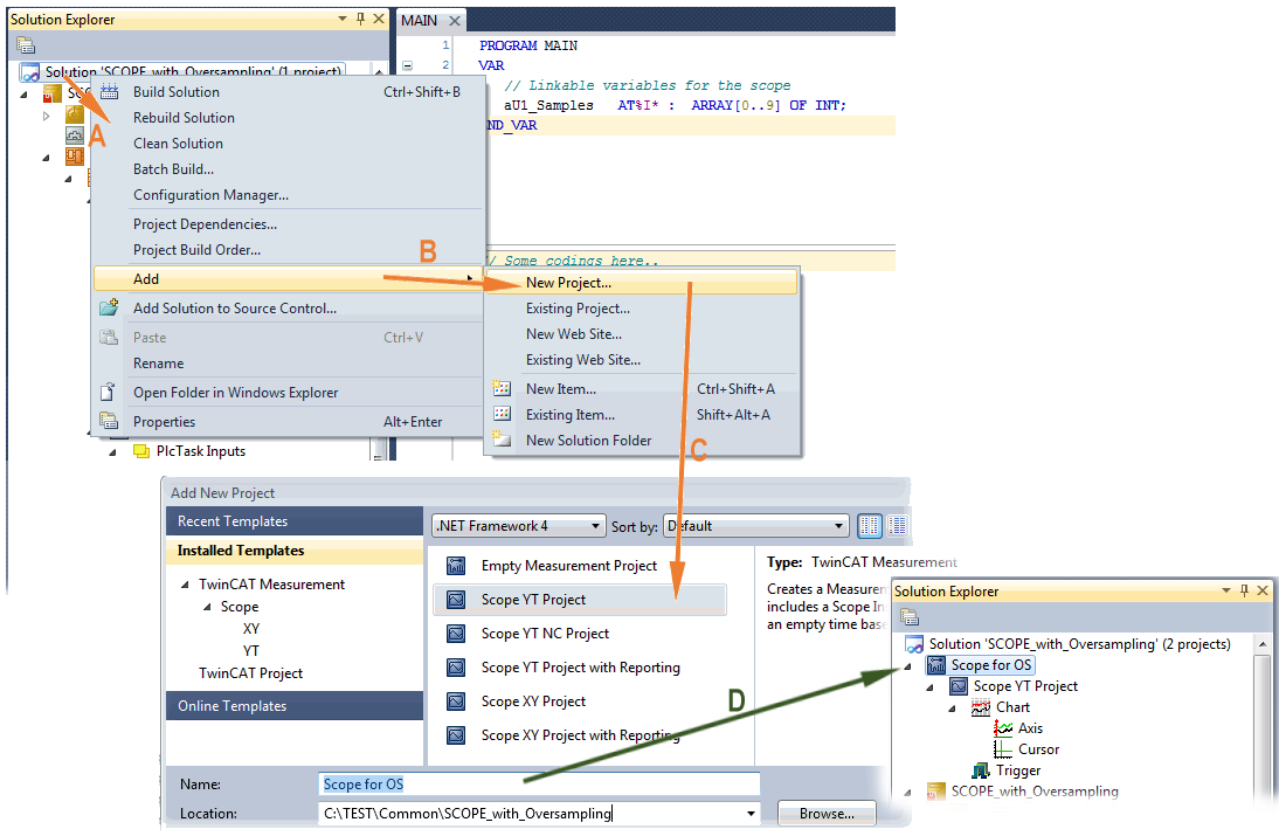


Fig. 153: Adding a Scope project into an already existing project

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 later. Notice that at least the number of elements have 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:

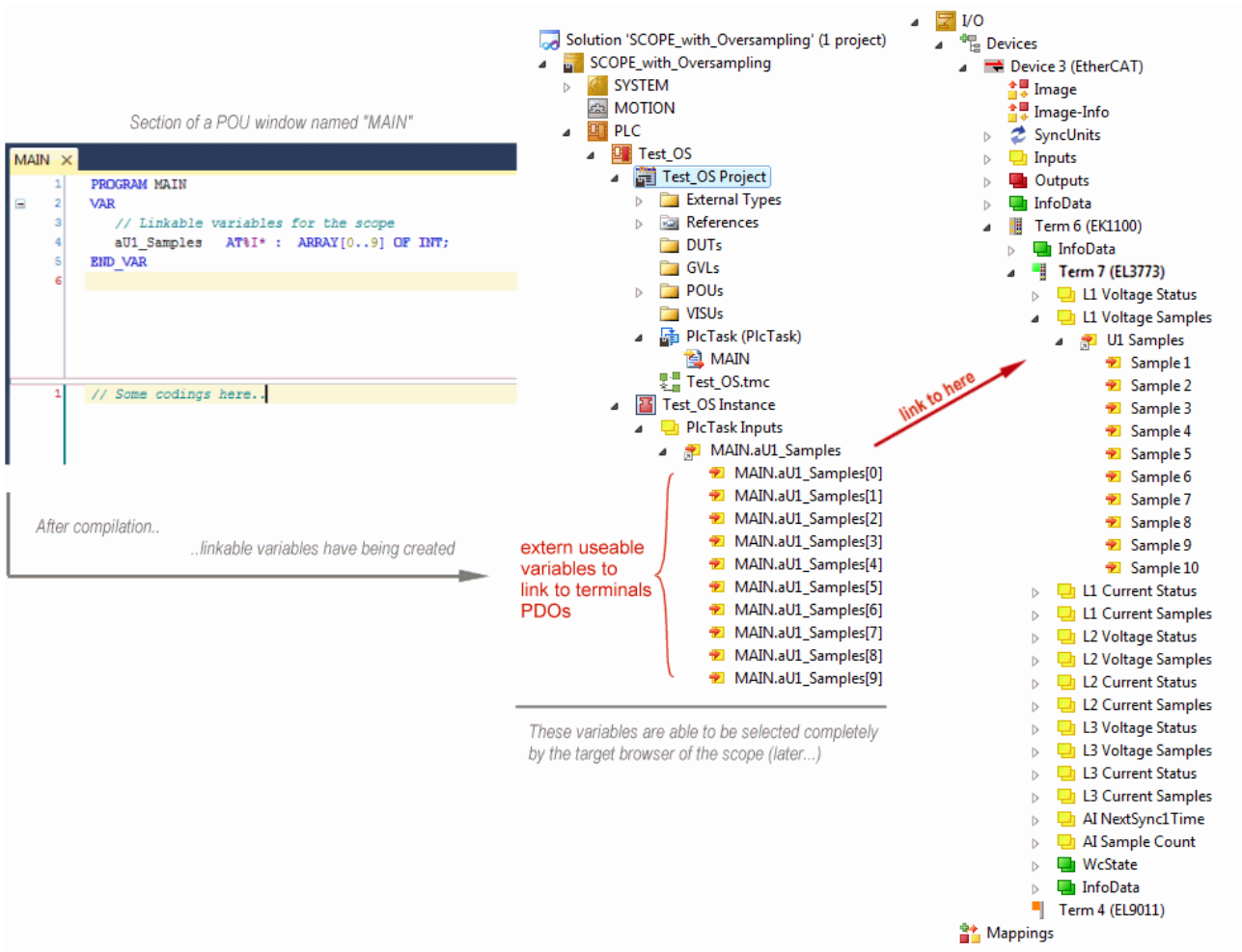


Fig. 154: Representation of a created PLC array variable („aU1_Samples“) to link with oversampling PDOs of EL3773

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...”.

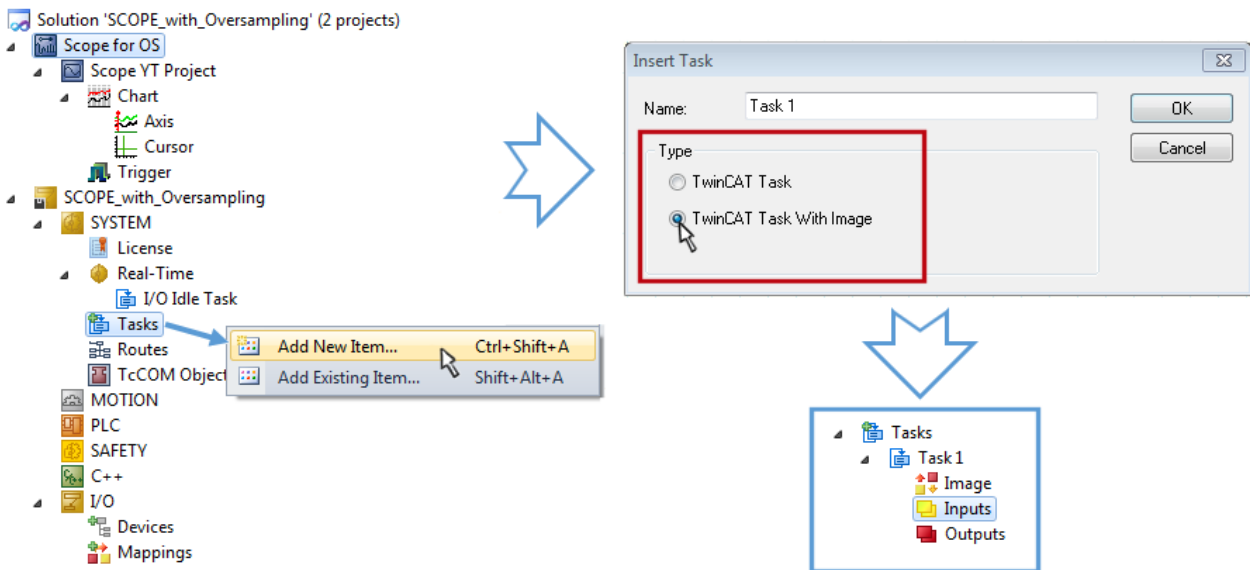


Fig. 155: Insertion of a free task

The Task have 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 have 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:

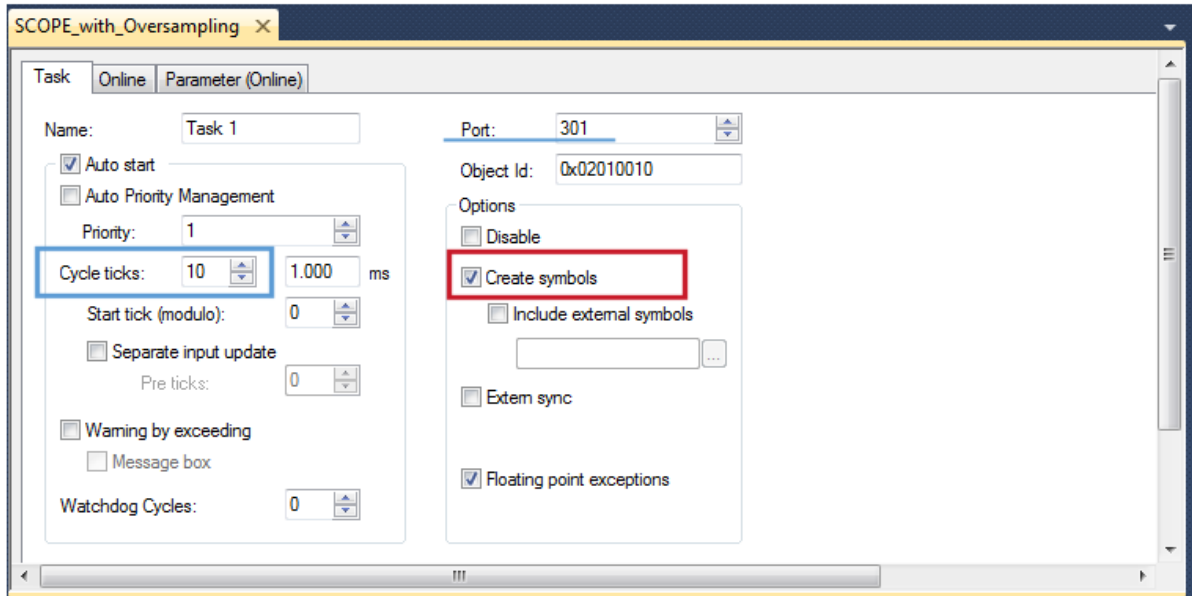


Fig. 156: Task property "Create symbols" must be activated

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

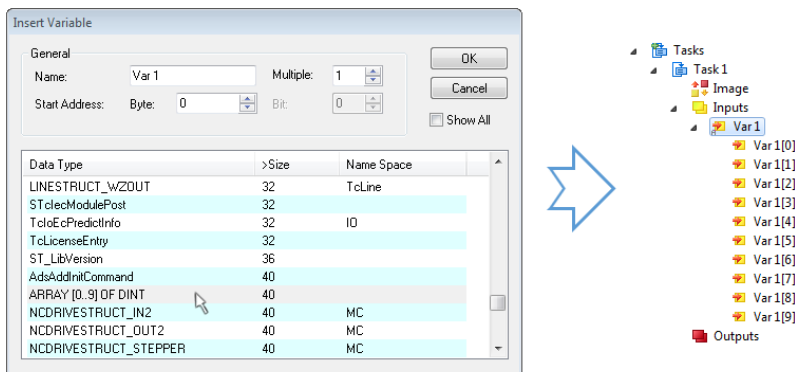


Fig. 157: Insertion of variable "Var 1" fitting to the oversampling (-factor)

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:

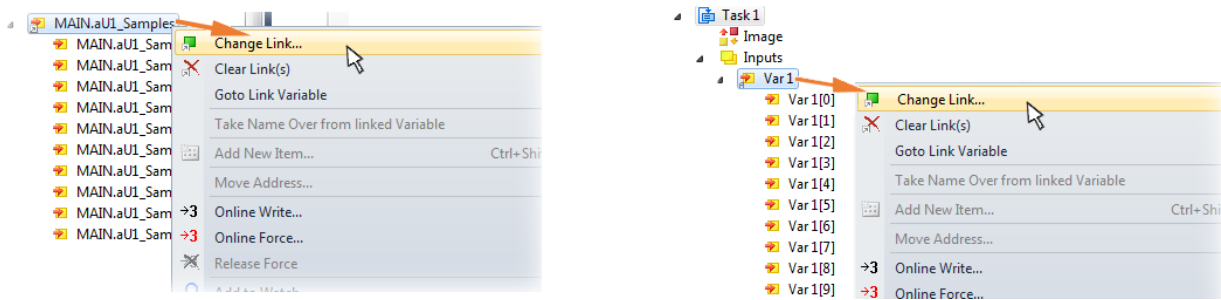


Fig. 158: Set up the link of the PLC array variable (left: for the last preceding paragraph Step 2a, right: for the last preceding paragraph Step 2b)

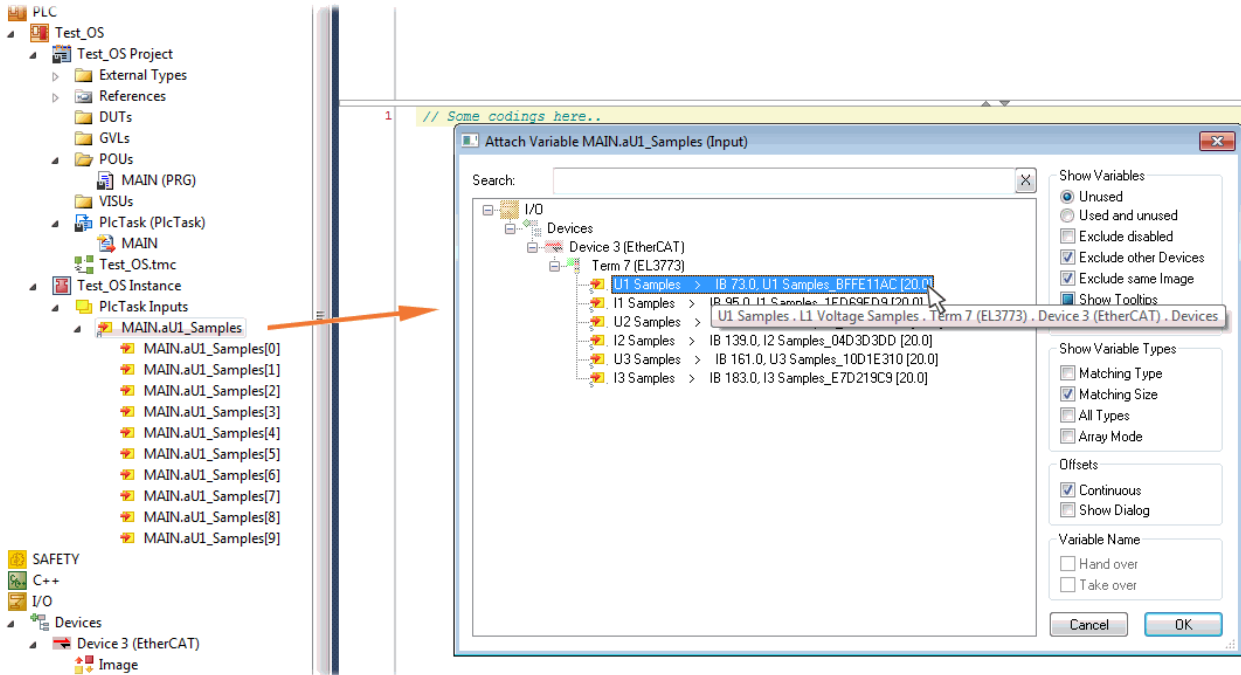


Fig. 159: Select the EL3773 PDO "L1 Voltage Samples" to create a link to the PLC array variable „aUI_Samples“

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):

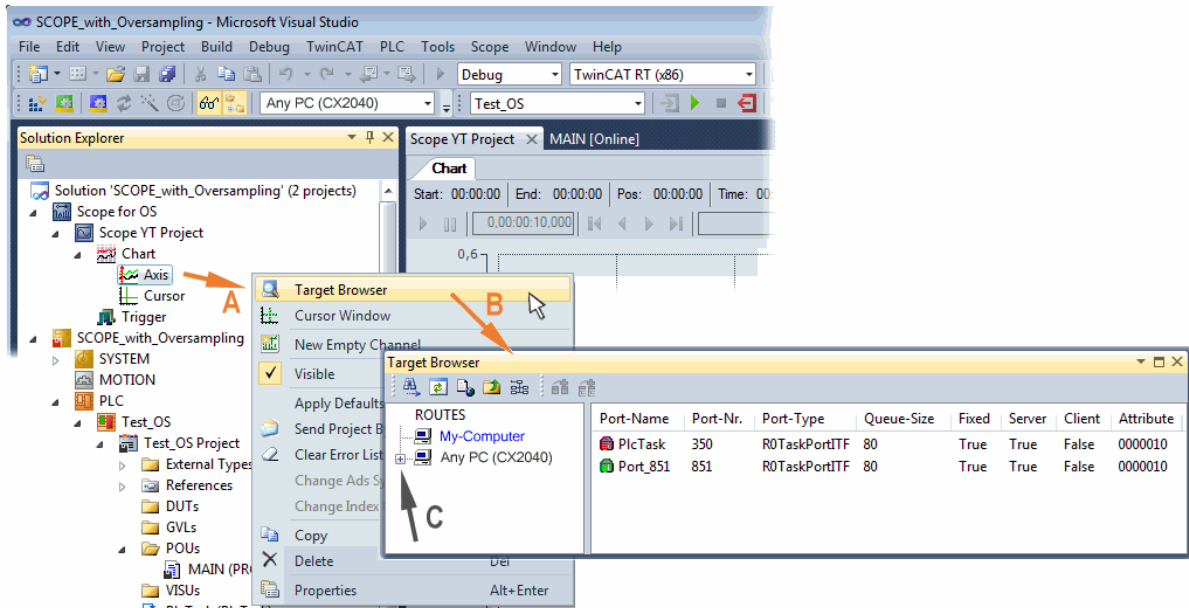


Fig. 160: Selection of the oversampling variable with the target browser

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.

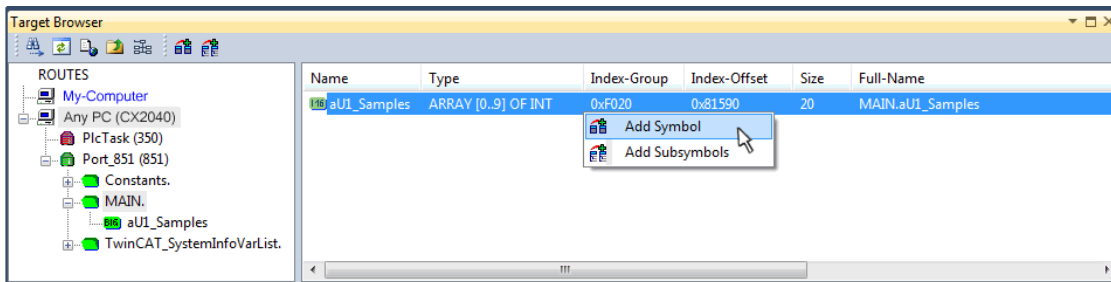


Fig. 161: Appending the variable "aUI_Samples" below "axis" within the scope project of the solution explorer

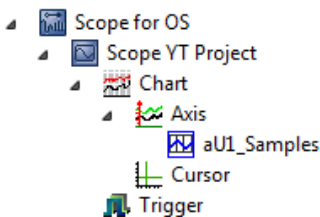
i Variable don't appears into the target browser



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 have 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:

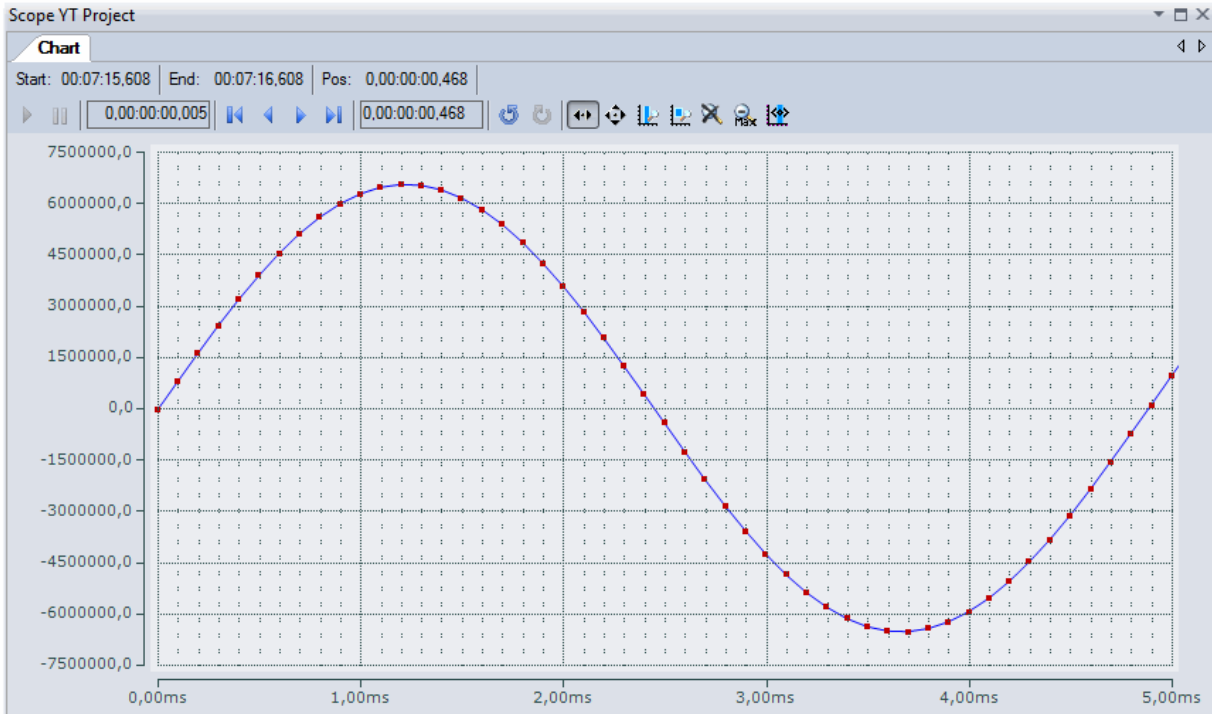




Fig. 162: Example of recording a sine signal with 10 x oversampling at 1 ms measurement cycle time

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 \[▶ 121\]](#) above) the oversampling PDO of the respective oversampling able terminal can be made visible for the ScopeView by activation of the ADS server.

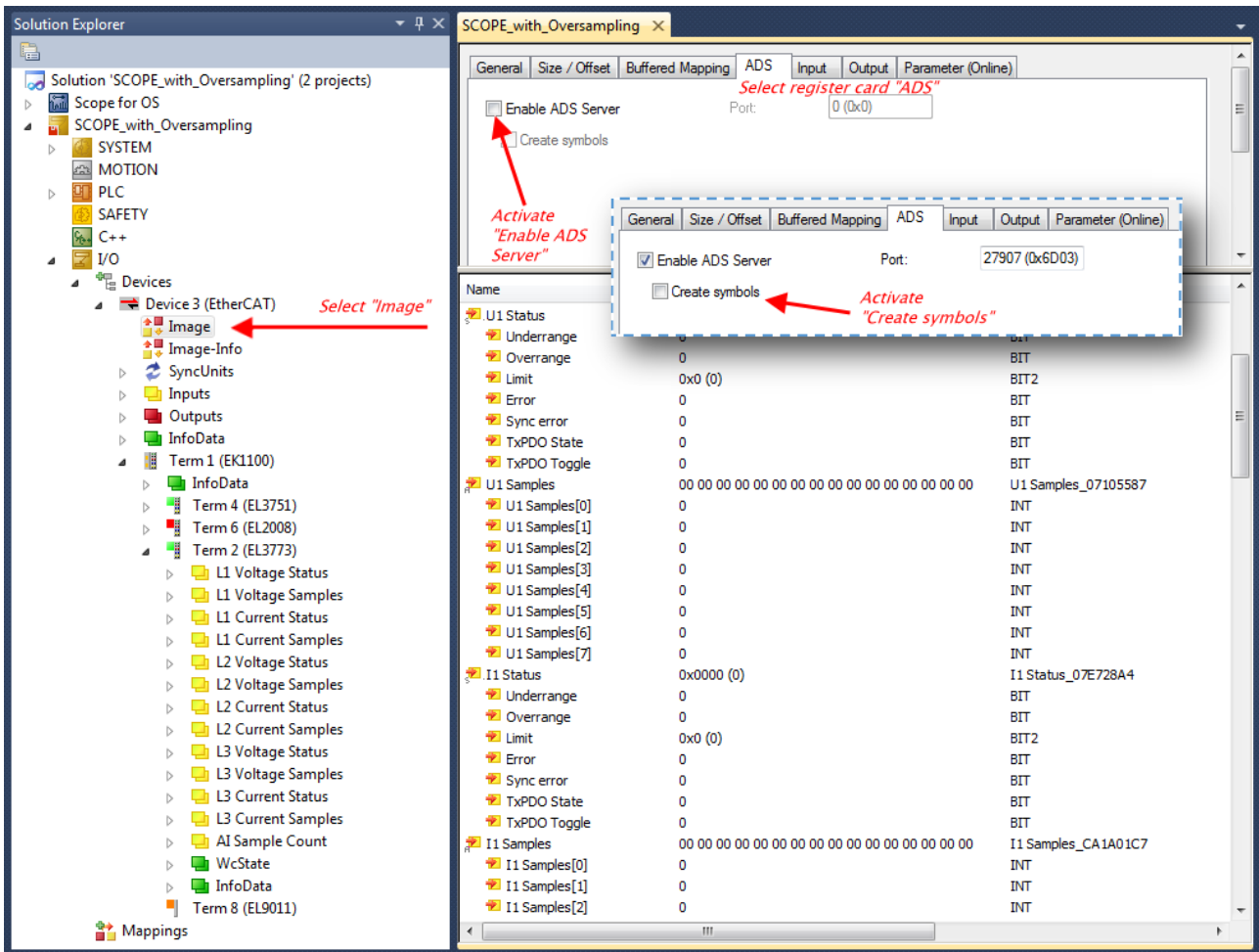


Fig. 163: Activation of the ADS server of the EtherCAT device (TwinCAT 3)

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:

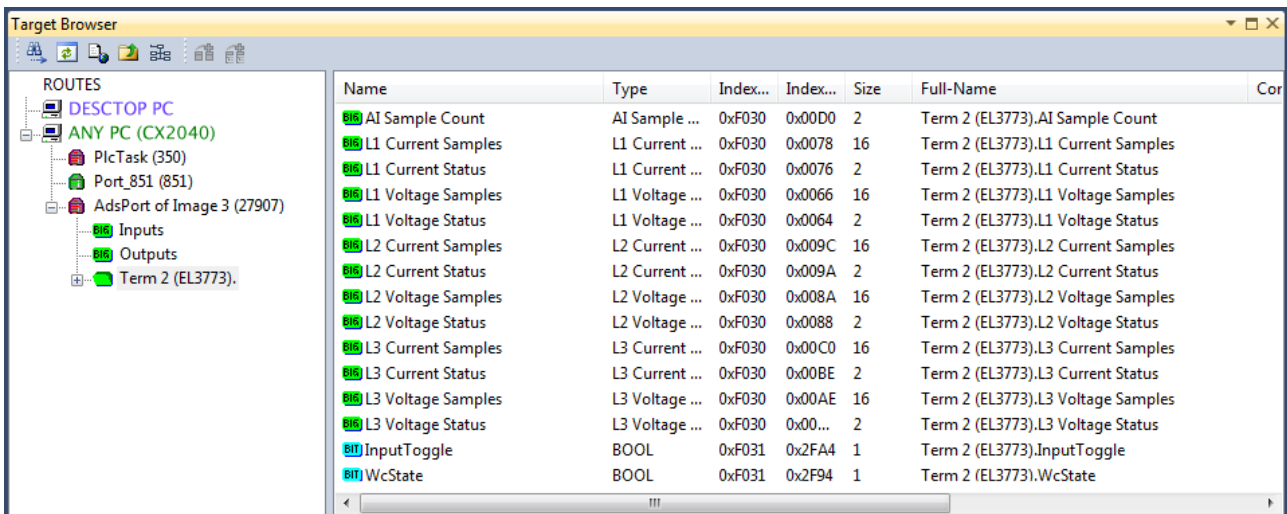
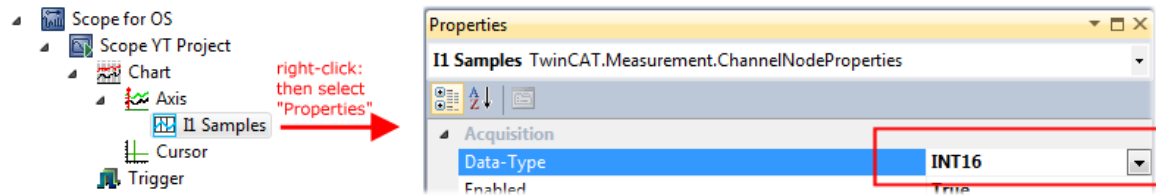


Fig. 164: Direct access to PDOs of the terminal by ScopeView

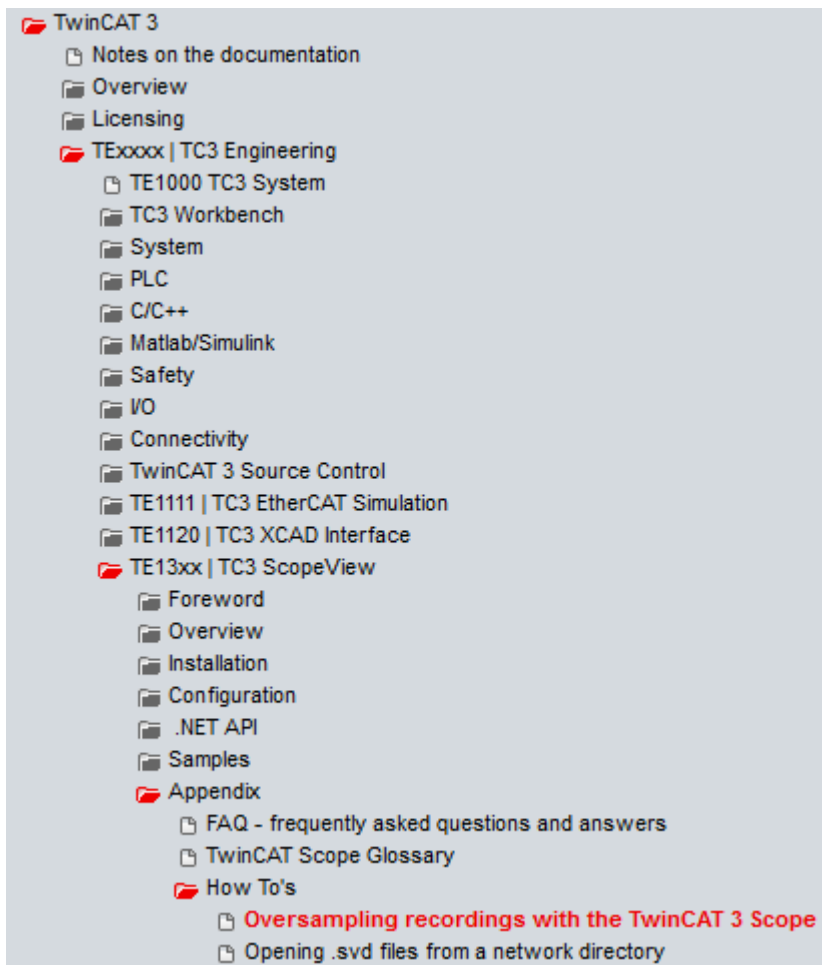
i Data type not valid

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:



6.4.2 TwinCAT 2 procedure

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

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](#) | [130](#)

- or directly in the System Manager if only one free task is present, see [step 1b \[p. 130\]](#)

The same settings are to be made in the Scope2 for both cases, see [step 2 \[p. 132\]](#)

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. 165: 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.

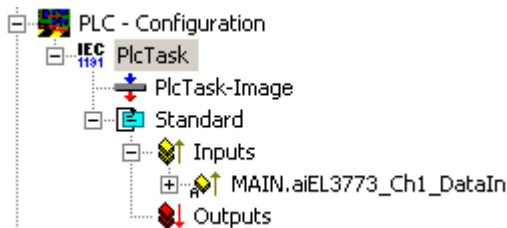


Fig. 166: PLC in the System Manager

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.

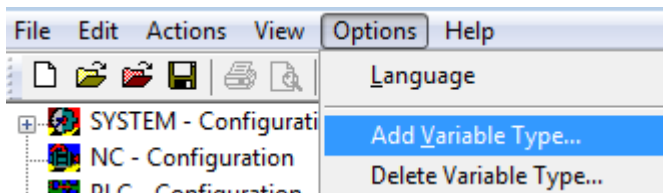


Fig. 167: Add Variable Type

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.

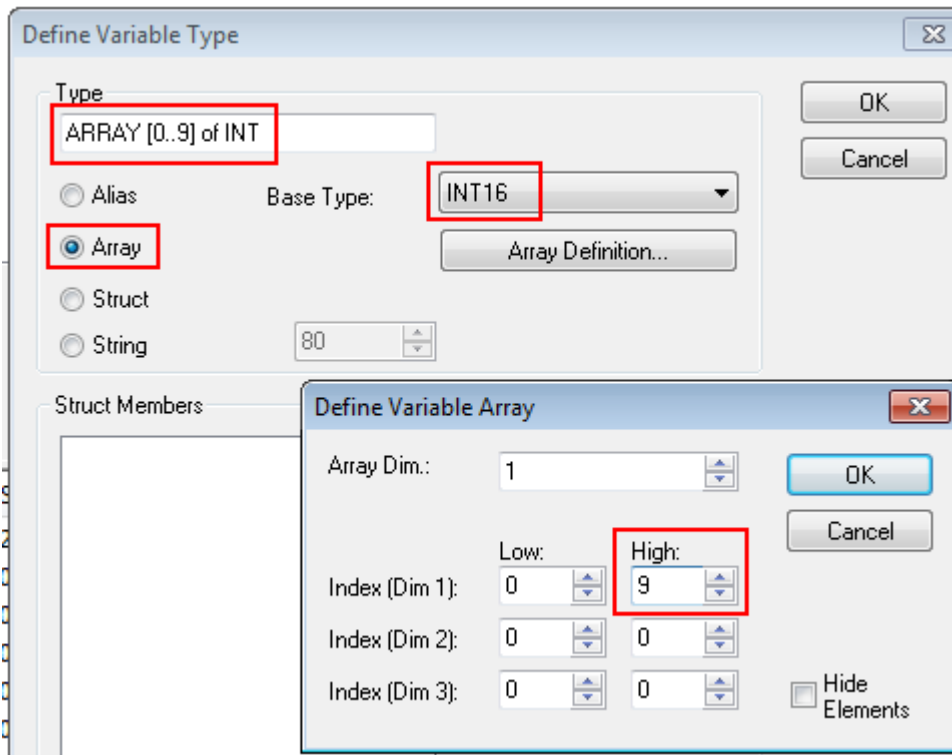


Fig. 168: Definition of the variable type

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.

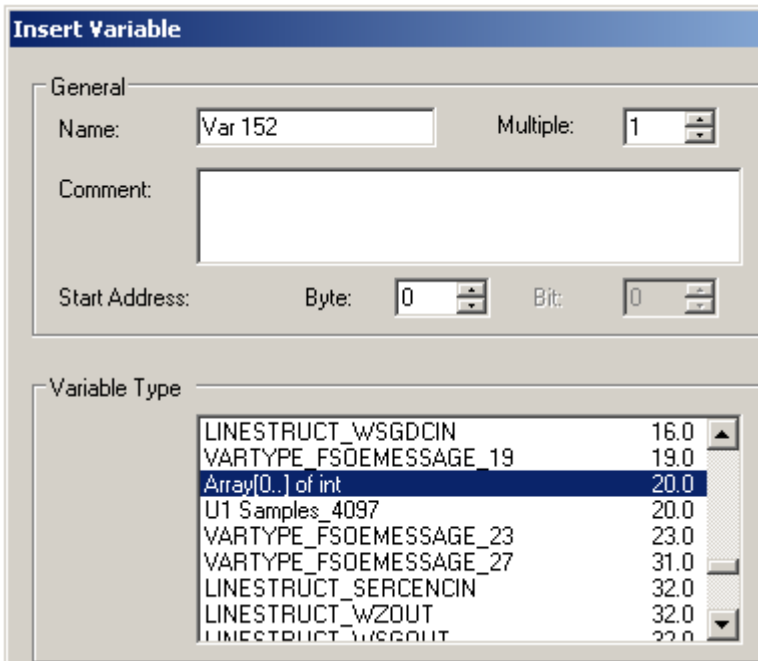


Fig. 169: Overview of declared types

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

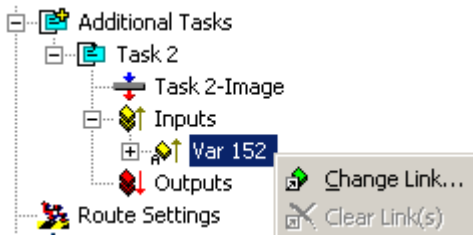


Fig. 170: Linking

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

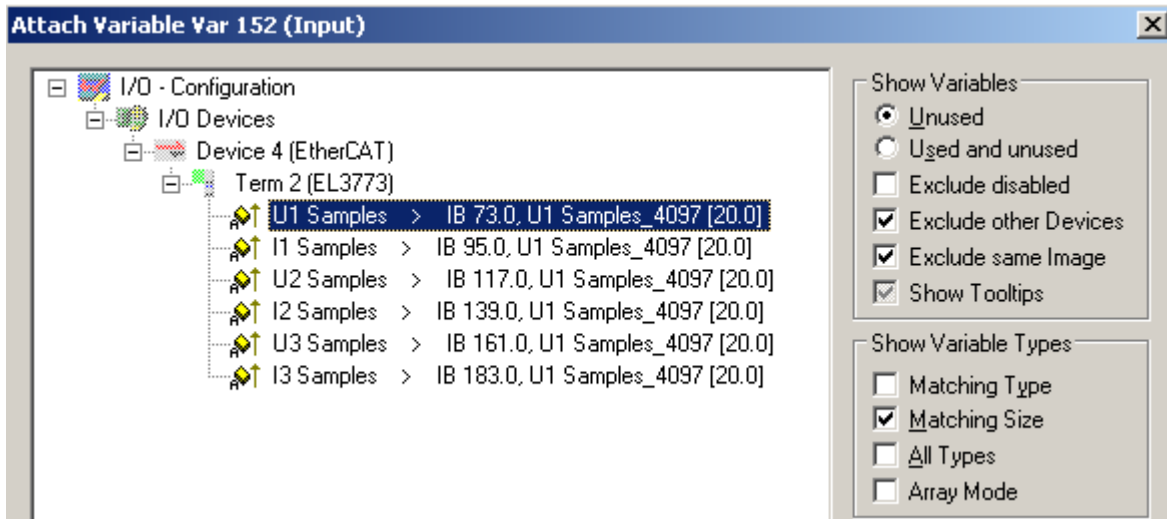


Fig. 171: Array variables of an oversampling terminal

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.

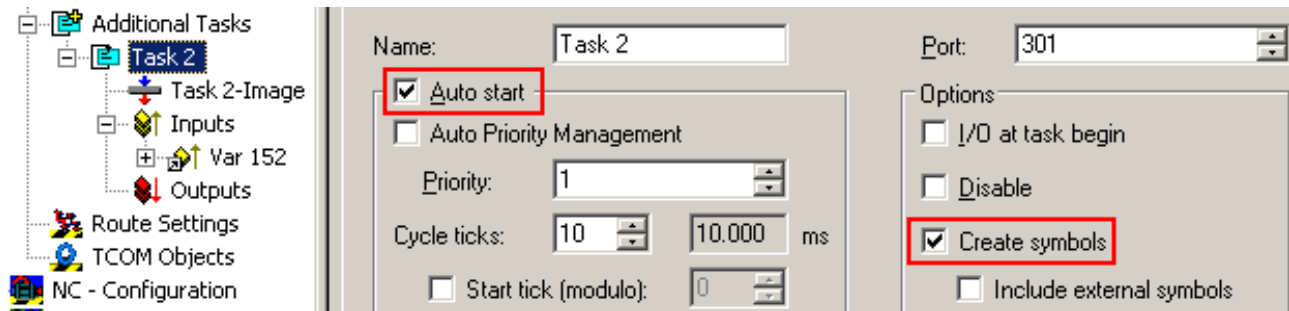


Fig. 172: Settings in the additional task

Step 2: Configuration in the Scope2

So that the linking works, an array variable with the channel data of the respective terminal 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 \[p. 130\]](#).

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

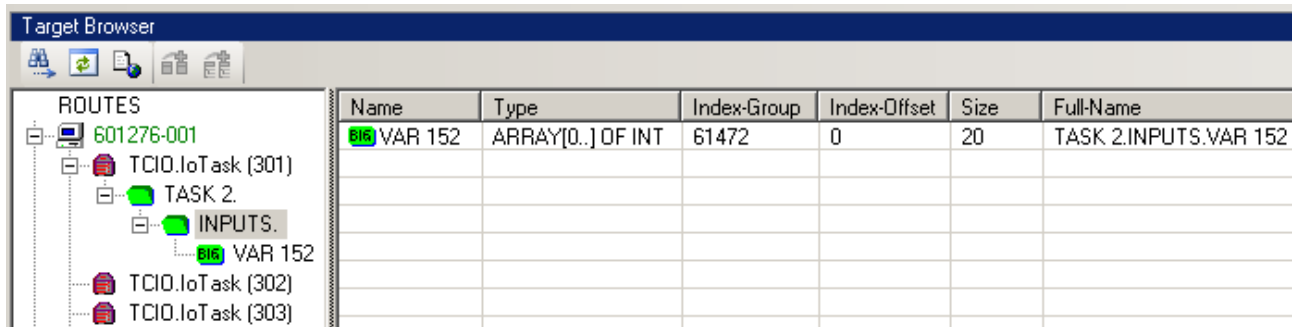


Fig. 173: Variable browser up to the array VAR152

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

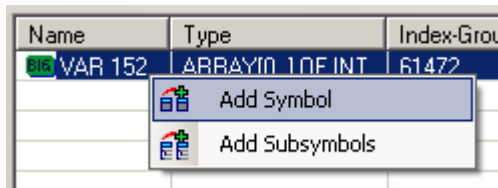


Fig. 174: AddSymbol on the array

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.

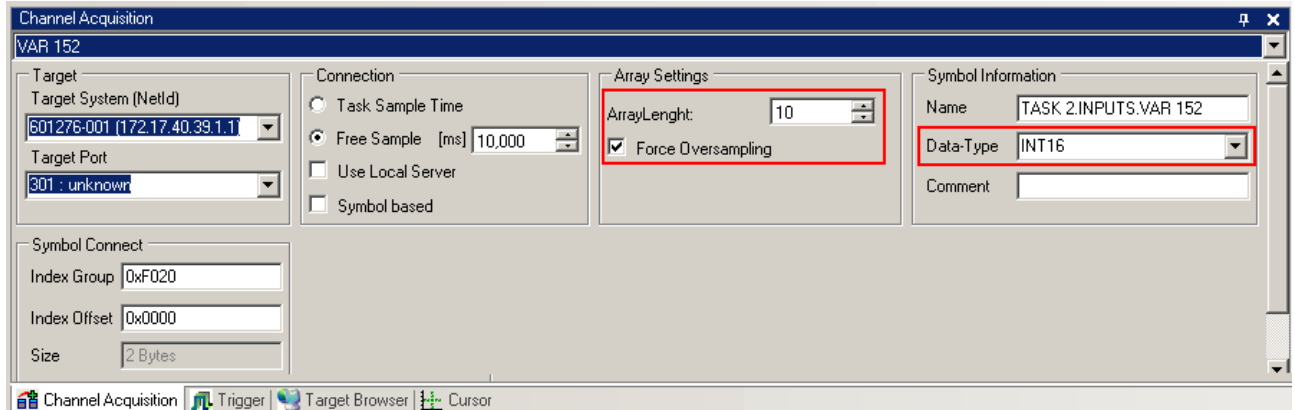


Fig. 175: Channel settings

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.

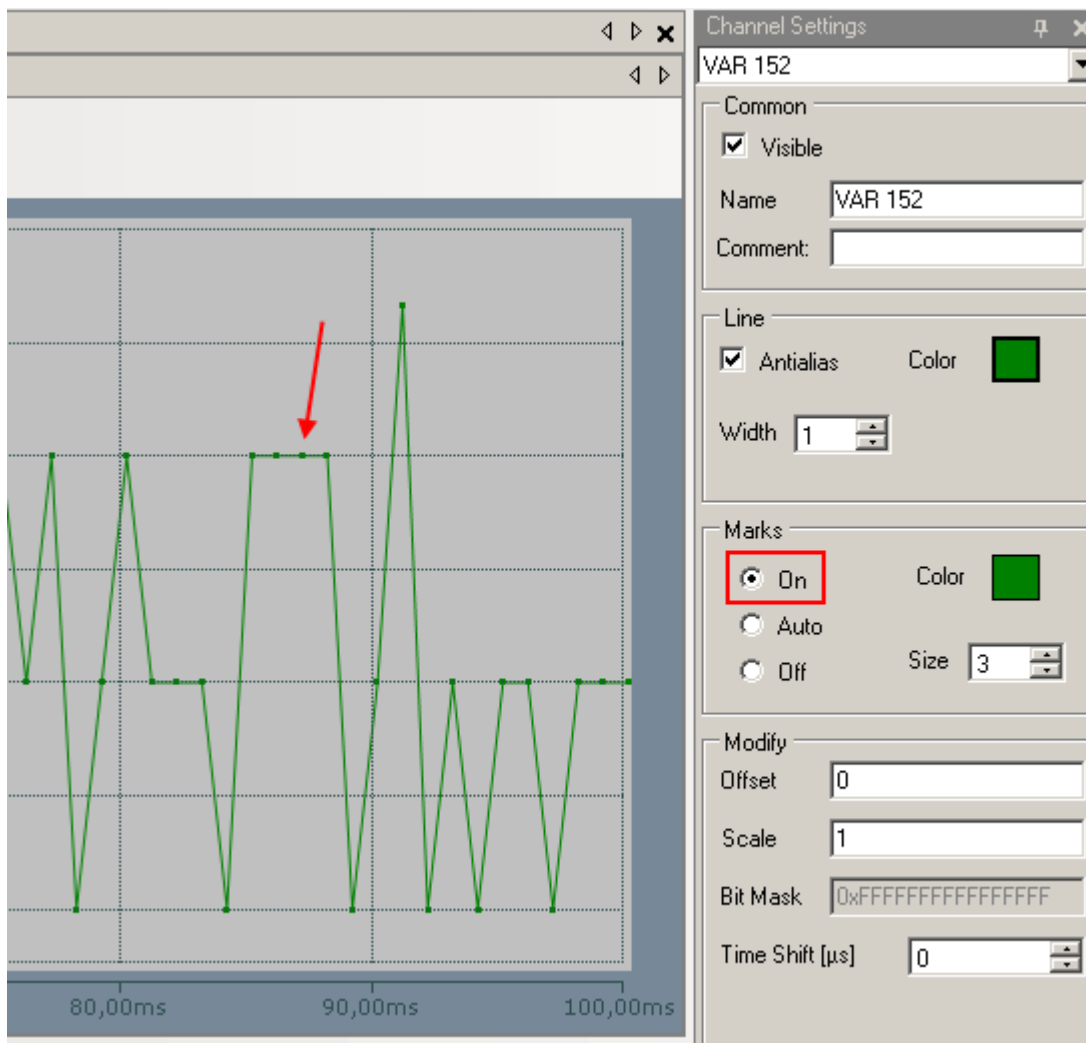


Fig. 176: Activation of the marks

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

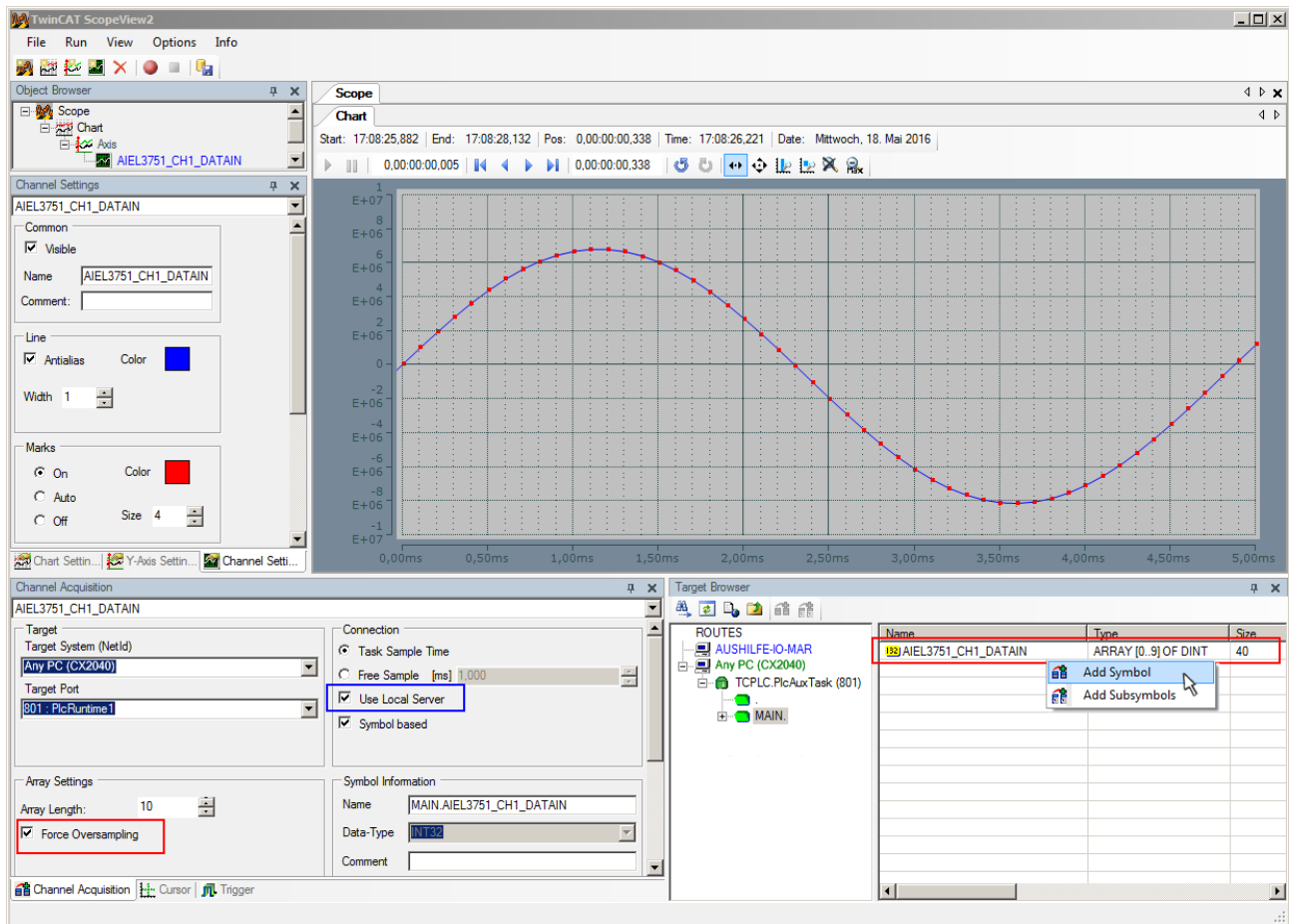


Fig. 177: Illustration of a 10 x oversampling variable of the EL3751 by the Scope2

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.

Proceeding with TwinCAT 2/ alternatively via ADS

In former TwinCAT 2 versions (or a lower revision as specified in the table [▶ 121] above) the oversampling PDO of the respective oversampling able terminal 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 is connected with have to be activated.

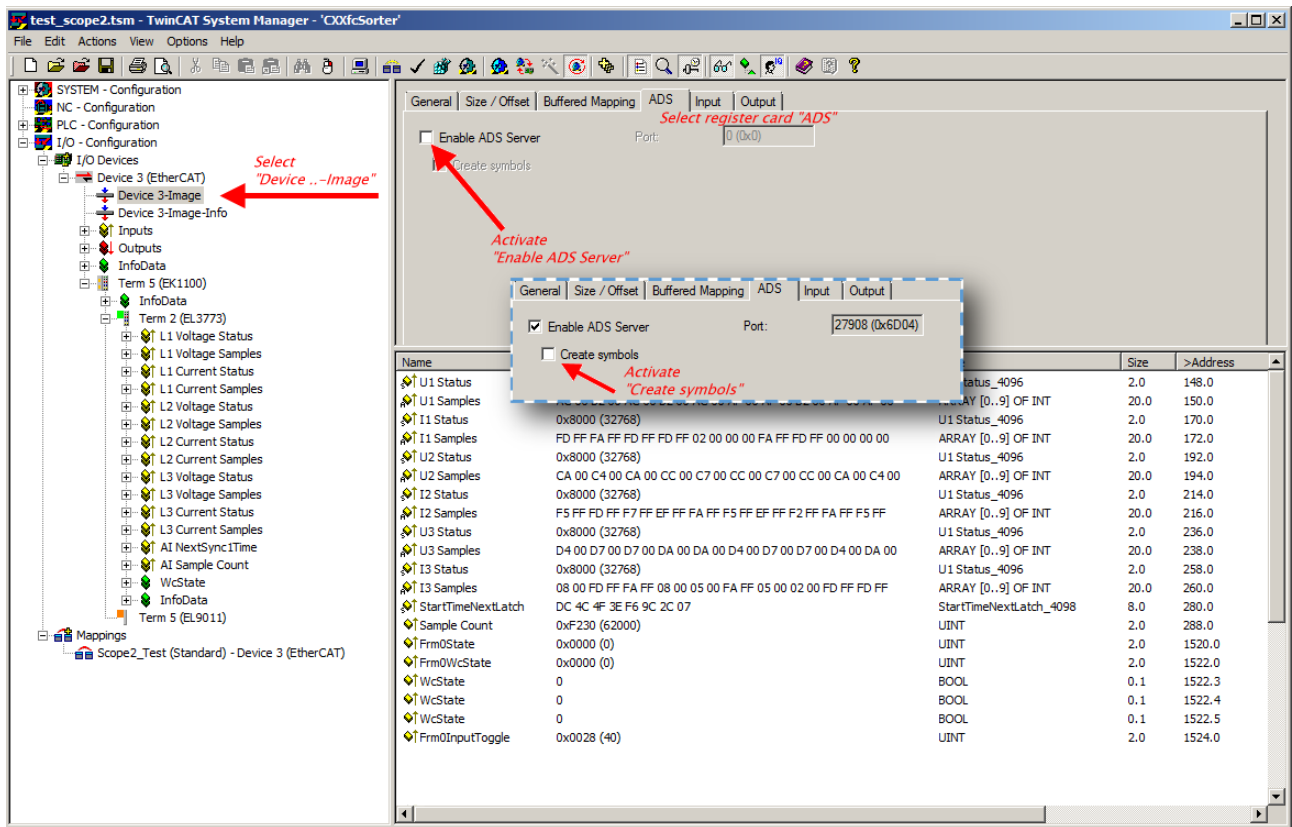


Fig. 178: Activation of the ADS server of the EtherCAT Device (TwinCAT 2)

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.

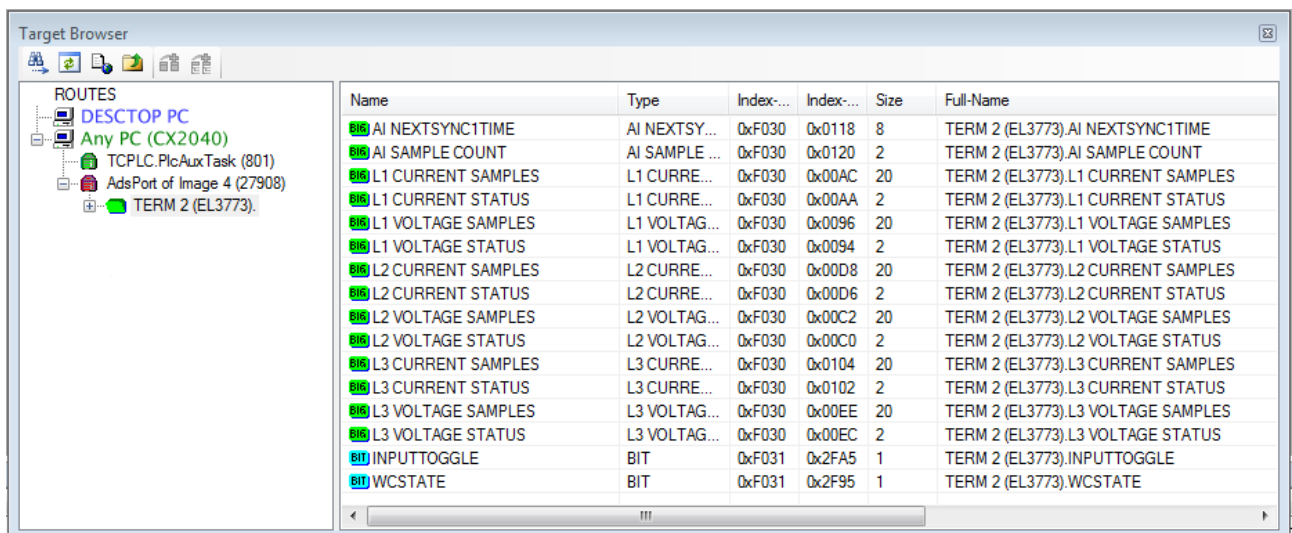
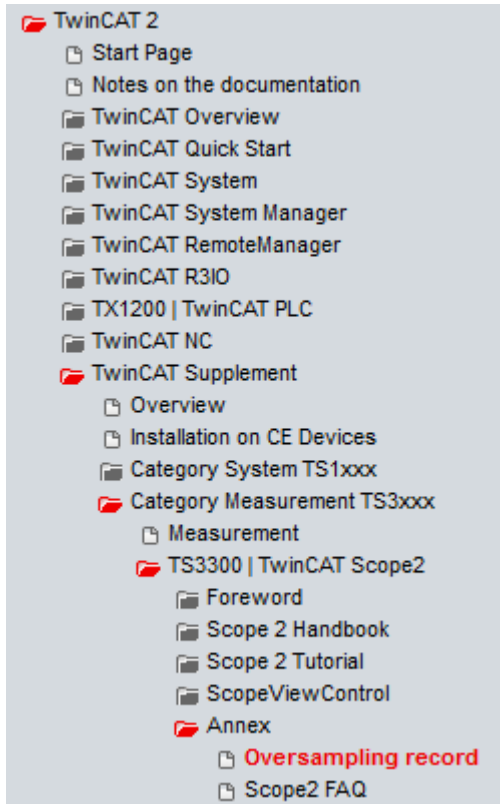


Fig. 179: Direct access of the Scope2 to the terminal's PDOs

Also see Beckhoff Information System

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



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

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

es muss eine ARRAY-Variable vorliegen die über ADS erreichbar ist. Dies kann eine PLC-Variable oder eine im System Manager definierte ARRAY-Variable sein. Diese wird dann vom Scope2 erkannt.

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 device is just activated.

6.5 Process data and configuration

6.5.1 TwinCAT tree

The operating principle of the EL37x2 in conjunction with SYNC0 and SYNC1 pulses was already described in section "Basic Function Principles". This section describes the application of the Beckhoff TwinCAT System Manager for setting the parameters.

Once the terminal has been scanned in, the following tree structure appears (see Fig. *TwinCAT tree EL37x2*).

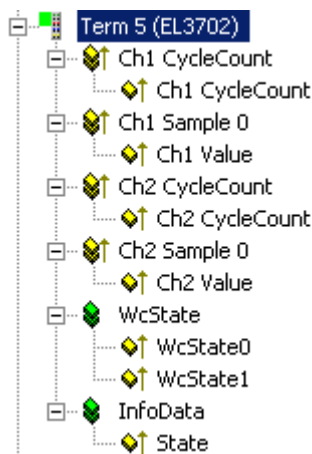


Fig. 181: TwinCAT tree EL37x2

Ch1/Ch2 CycleCount and Ch1/Ch2 Value

Each EL37x2 record contains a CycleCounter for each channel that is incremented with each record, and n analog measured values with a width of 2 bytes each. With an oversampling factor of $n = 25$ a record therefore consists of a 16-bit CycleCounter (overflowing) and 25 16-bit samples per channel. The CycleCounter in the higher-level control system can be used to verify the consistency of the record transfer.

By default the EL37x2 is set to $n = 1$, resulting in the process image shown in Fig. *TwinCAT tree EL37x2*.

WcState 0/1 (Online tab)

Boolean variable indicating the working counter status of the terminal. An invalid working counter (value: 1) is displayed if invalid data were transferred by the EtherCAT slave controller.

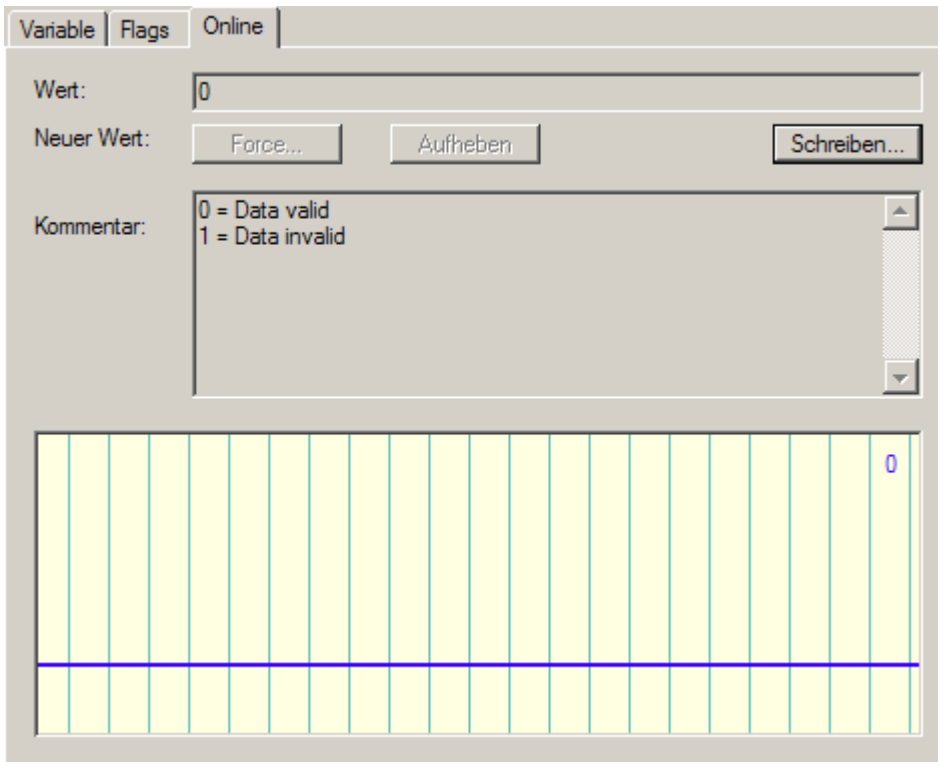


Fig. 182: WCState, Online tab

InfoData, State

Shows the online status (EtherCAT state machine) of the terminal.

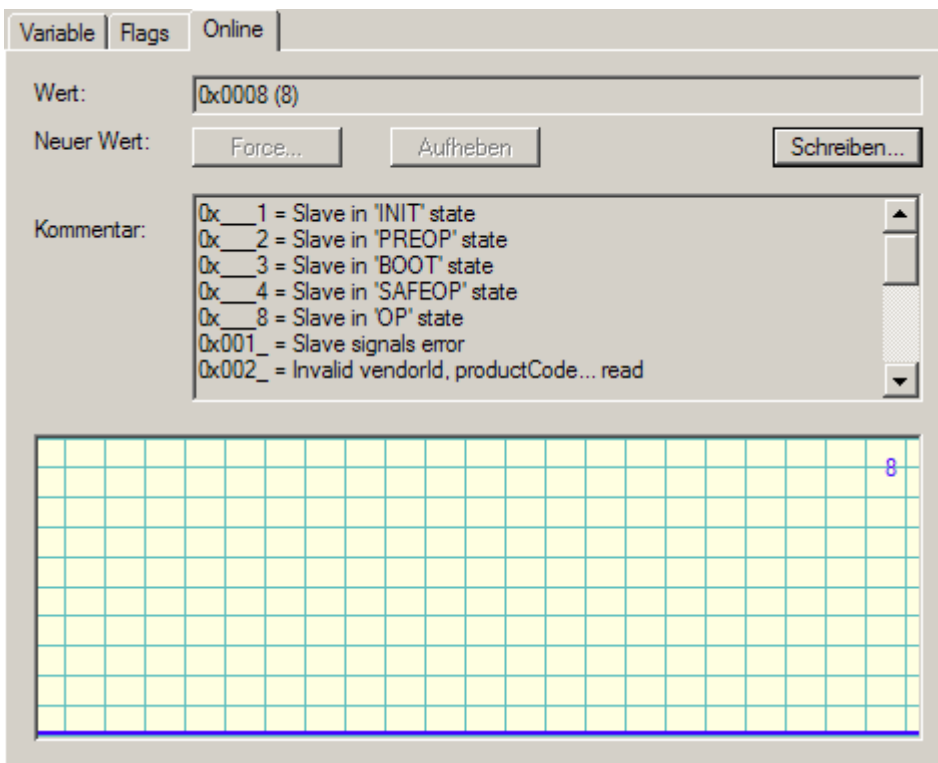


Fig. 183: InfoData, Online tab

Value	Description
0x__1	Slave in 'INIT' state
0x__2	Slave in 'PREOP' state
0x__3	Slave in 'BOOT' state
0x__4	Slave in 'SAFEOP' state
0x__8	Slave in 'OP' state
0x001_	Slave signals error
0x002_	Invalid vendorId, productCode... read
0x004_	Initialization error occurred
0x010_	Slave not present
0x020_	Slave signals link error
0x040_	Slave signals missing link
0x080_	Slave signals unexpected link
0x100_	Communication port A
0x200_	Communication port B
0x400_	Communication port C
0x800_	Communication port D

Table 1: States of the EtherCAT state machine

6.5.2 Extent of process data in delivery state

Process data

The scope of the offered process data and the assignment of the SyncManagers in delivery state can be viewed via the “Process data” tab (see Fig. *Process data tab SM0+SM1+SM2, EL37x2*).

General | EtherCAT | DC | **Process Data** | Online

Sync Manager:

SM	Size	Type	Flags
0	4	Inputs	
1	4	Inputs	
2	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1B00	2.0	Ch1 CycleCount	MF	0	0
0x1A00	2.0	Ch1 Sample 0	MFO	0	0
0x1A01	2.0	Ch1 Sample 1	MFO		0
0x1A02	2.0	Ch1 Sample 2	MFO		0
0x1A03	2.0	Ch1 Sample 3	MFO		0
0x1A04	2.0	Ch1 Sample 4	MFO		0
0x1A05	2.0	Ch1 Sample 5	MFO		0
0x1A06	2.0	Ch1 Sample 6	MFO		0
0x1A07	2.0	Ch1 Sample 7	MFO		0
0x1A08	2.0	Ch1 Sample 8	MFO		0
0x1A09	2.0	Ch1 Sample 9	MFO		0
0x1A0A	2.0	Ch1 Sample 10	MFO		0

PDO Assignment (0x1C10):

- 0x1B00
- 0x1A00
- 0x1A01
- 0x1A02
- 0x1A03
- 0x1A04
- 0x1A05
- 0x1A06

PDO Content (0x1B00):

Index	Size	Offs	Name	Type	Default (hex)
0x6800:01	2.0	0.0	Ch1 CycleCount	UINT	
		2.0			

Download

- PDO Assignment
- PDO Configuration

Load PDO info from device

Sync Unit Assignment...

Fig. 184: Process data tab SM0, EL37x2

General | EtherCAT | DC | **Process Data** | Online

Sync Manager:

SM	Size	Type	Flags
0	4	Inputs	
1	4	Inputs	
2	0	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1B01	2.0	Ch2 CycleCount	MF	1	1
0x1A80	2.0	Ch2 Sample 0	MFO	1	1
0x1A81	2.0	Ch2 Sample 1	MFO		1
0x1A82	2.0	Ch2 Sample 2	MFO		1
0x1A83	2.0	Ch2 Sample 3	MFO		1
0x1A84	2.0	Ch2 Sample 4	MFO		1
0x1A85	2.0	Ch2 Sample 5	MFO		1
0x1A86	2.0	Ch2 Sample 6	MFO		1
0x1A87	2.0	Ch2 Sample 7	MFO		1
0x1A88	2.0	Ch2 Sample 8	MFO		1
0x1A89	2.0	Ch2 Sample 9	MFO		1
0x1A8A	2.0	Ch2 Sample 10	MFO		1

PDO Assignment (0x1C11):

- 0x1B01
- 0x1A80
- 0x1A81
- 0x1A82
- 0x1A83
- 0x1A84
- 0x1A85
- 0x1A86

PDO Content (0x1B01):

Index	Size	Offs	Name	Type	Default (hex)
0x6800:02	2.0	0.0	Ch2 CycleCount	UINT	
		2.0			

Download

- PDO Assignment
- PDO Configuration

Load PDO info from device

Sync Unit Assignment...

Fig. 185: Process data tab SM1, EL37x2

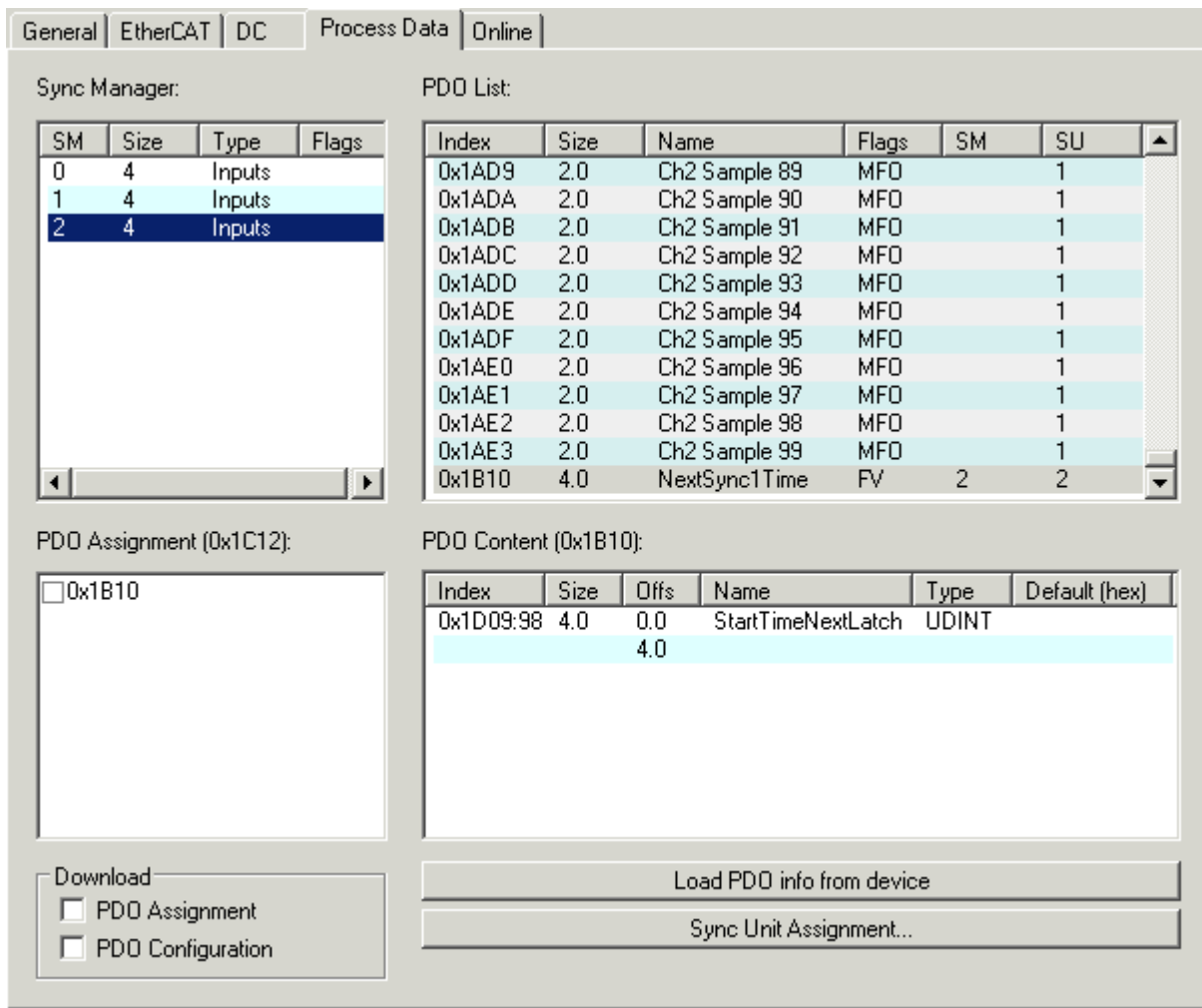


Fig. 186: Process data tab SM2, EL37x2

Sync Manager (SM)

To configure the process data, select the required Sync Manager in the “Sync Manager” field at the top left (Fig. *Process data tab SM0+SM1+SM2, EL37x2*) (only SM2 is editable). The process data assigned to this Sync Manager can then be switched on or off in the “PDO Assignment” box underneath (index 0x1B10). Restarting the EtherCAT system, or reloading the configuration in configuration mode (F4), causes the EtherCAT communication to restart, and the process data is transferred from the terminal.

SM0, PDO Assignment 0x1C10				
Index	Index of excluded PDOs	Size (byte.bit)	Name	PDO content
0x1B00	-	2.0	Ch 1 CycleCount	Index 0x6800:01 - Ch 1 CycleCount
0x1A00 .. 0x1A63	-	2.0	Ch 1 Sample 0 .. Ch 1 Sample 99	Index 0x6000:01 - Ch 1 Value .. Index 0x6630:01 - Ch 1 Value

SM1, PDO Assignment 0x1C11				
Index	Index of excluded PDOs	Size (byte.bit)	Name	PDO content
0x1B01	-	2.0	Ch 2 CycleCount	Index 0x6800:01 - Ch 2 CycleCount
0x1A80 .. 0x1AE3	-	2.0	Ch 2 Sample 0 .. Ch 2 Sample 99	Index 0x6000:01 - Ch 2 Value .. Index 0x6630:01 - Ch 2 Value

SM2, PDO assignment 0x1C12				
Index	Index of excluded PDOs	Size (byte.bit)	Name	PDO content
0x1B10	-	4.0	Ch 2 CycleCount	Index 0x1D09:98:01 - StartNextTimeLatc h

Table 2: PDO assignment of sync managers

6.5.3 Oversampling settings, distributed clocks (DC)

Sample: Setting 2-channel, 2 times oversampling

The oversampling factor can be set in the "DC" tab (see Fig. *Process data tab, sample setting 2-channel, 2 times oversampling*). If only one channel of the EL37x2 is required, the corresponding operating mode can be selected as shown in Fig. *Process data tab, example setting two-channel, 2 x oversampling*, in order to reduce the process data. When an entry is selected in the dialog, the correct sampling ratio at distributed clock level and the correct number of process data to be transferred are set automatically.

● Sampling frequency

i If an oversampling factor is required that does not appear in the list, the user has to specify the ratio between the SYNC0 pulse and the SYNC1 pulse within the permitted limits, based on the information found in "Basic Function Principles". See the [example \[► 150\]](#).

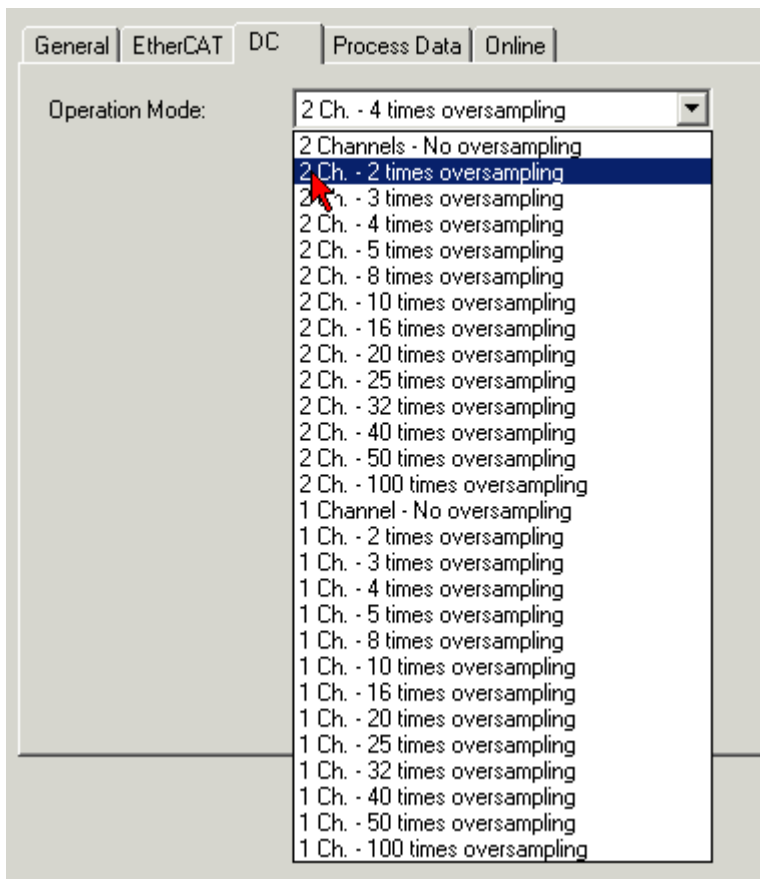


Fig. 187: Process Data tab, example setting two-channel, 2 x oversampling

An oversampling setting of $n = 2$ with two channels in use results in the process image shown in Fig. *TwinCAT tree, example setting two-channel, 2 x oversampling.*

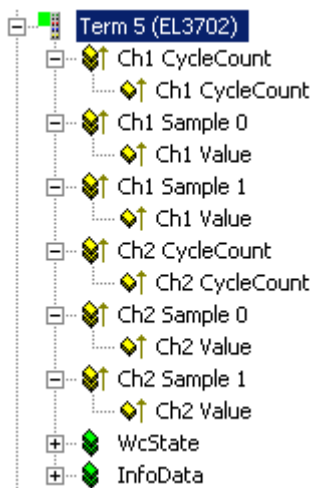


Fig. 188: TwinCAT tree, sample setting 2-channel, 2-times oversampling

"StartTimeNextLatch" activation

The assigned process record "StartTimeNextLatch" for transfer to the EtherCAT bus can be enabled by entering 0x1B10 in the field below, after selecting Sync Manager 2 (Fig. *Process Data tab, activation of index 0x1B10, entry "StartTimeNextLatch", top left.*

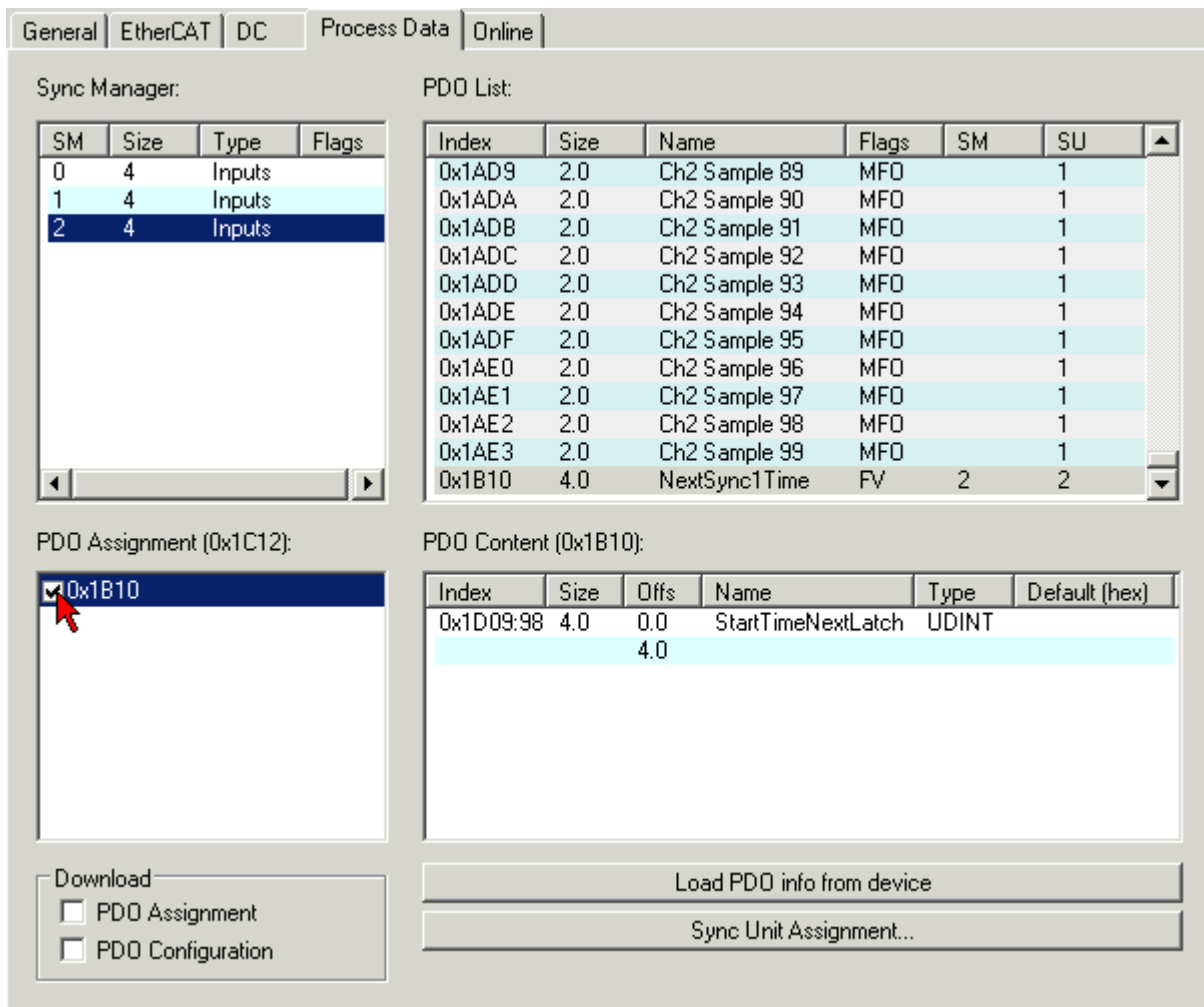


Fig. 189: Process Data tab, activation of index 0x1b10, entry "StartTimeNextLatch"

The StartTimeNextlatch process data is 32 bits wide. During each process data cycle the time is specified at which the *next* SYNC1 pulse and therefore the next block of sample values begins. StartTimeNextLatch 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. The EL37x2 only maps the lower 32 bits of the generally 64-bit distributed clock time.

In this way all samples can be synchronized with other time data within the EtherCAT bus based on the known oversampling factor.

Sample:

With a cycle time of 1 ms (= 1,000,000 ns) and an oversampling factor of 20 in the regarded cycle, the EL37x2 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 5th supplied sample is now to be determined, i.e. the Distributed Clocks time at which the 5th 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 5th sample was determined at the time $6,777,216 + ((5 - 1) * 50,000) = 6,977,216$ ns.

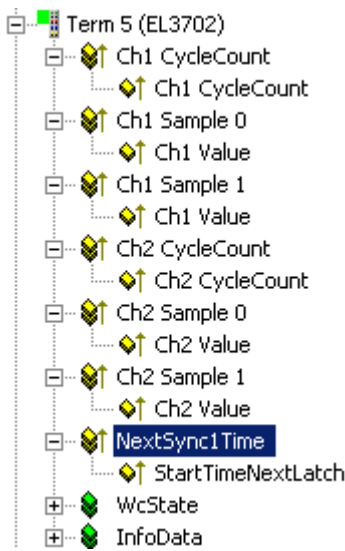


Fig. 190: TwinCAT tree, entry "StartTimeNextLatch"

Special oversampling factor and Shift Time for the SYNC0 pulse

⚠ CAUTION

CAUTION! Risk of device damage!

If these settings are changed in the System Manager, no plausibility checks are carried out on the software side. Correct function of the terminal with all conceivable setting options cannot be guaranteed.

To set an oversampling factor that is not listed in the dialog, the ratio between SYNC0 and SYNC1 pulse can be set manually. To this end open the dialog "Advanced Settings" on the DC tab (Fig. DC tab, activation "Advanced Settings...").

A local shift time for offsetting several EL37x2 devices can also be set here (Fig. DC tab, activation „Advanced Settings...“, „Distributed clocks“).

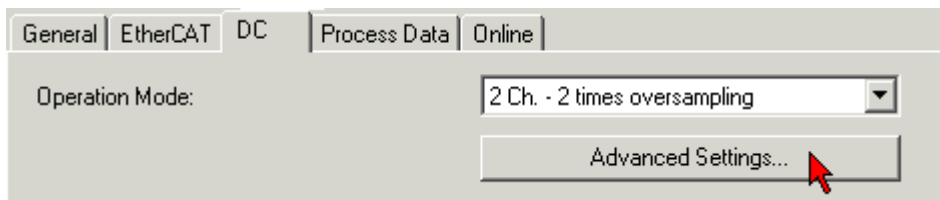


Fig. 191: DC tab, activation "Advanced Settings..."

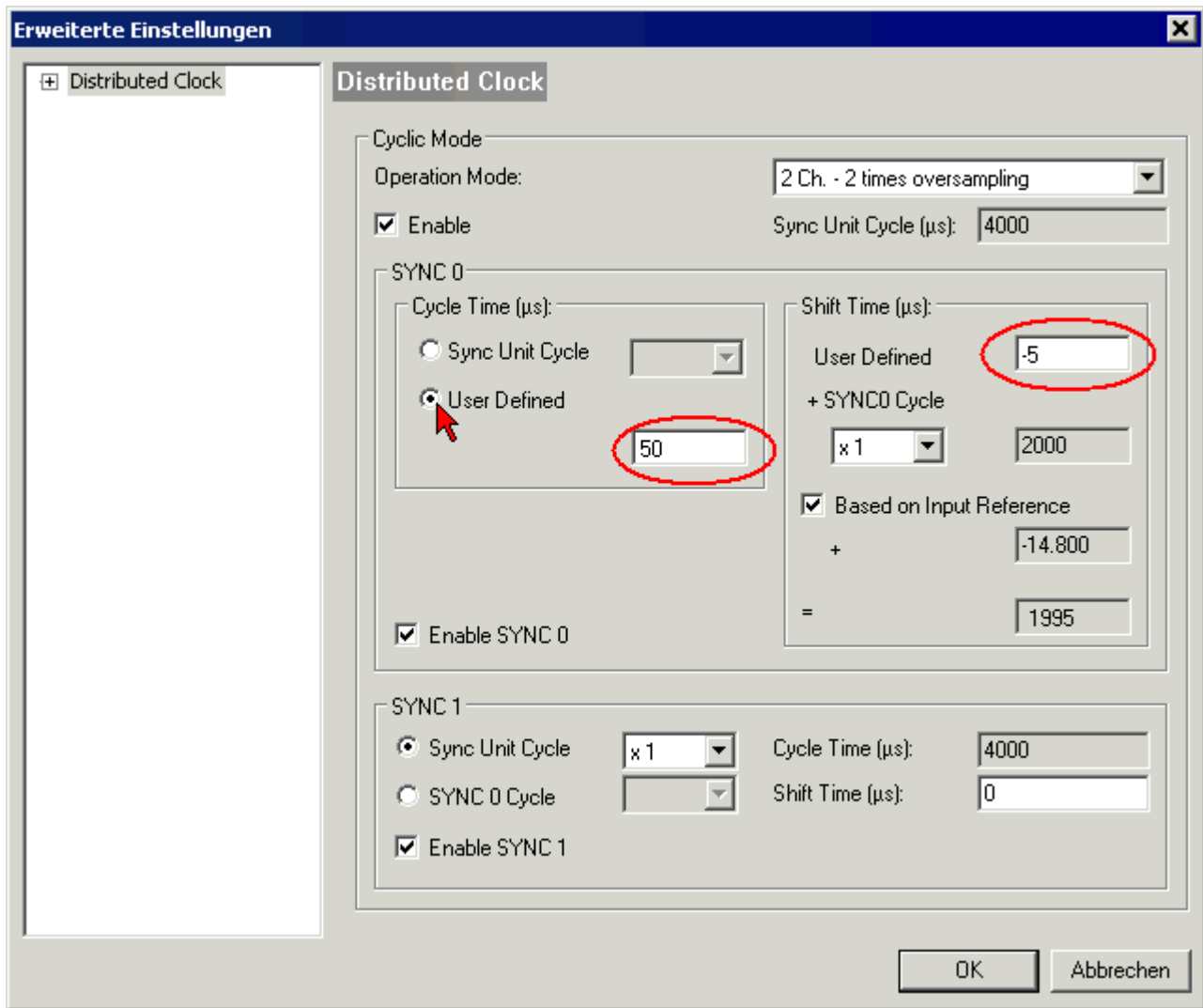


Fig. 192: DC tab, activation “Advanced Settings...”, “Distributed Clocks”

Sample:

Setting the new oversampling factor:

In Fig. *DC tab, activation „Advanced Settings...“, „Distributed Clocks“*, TwinCAT is in Config mode with 4 ms cycle time (4000 µs). The SYNC1 pulse is triggered every 4000 µs. The oversampling factor should be set to 80. This requires a SYNC0 interval of 50 µs. To this end either select a suitable factor for SYNC0 Cycle Time, or manually enter 50 µs via *User Defined*. The ADC is now triggered at the required interval.

In addition, the number of transferred process data has to be adjusted to the new factor. In this example 80 values are required for each channel.

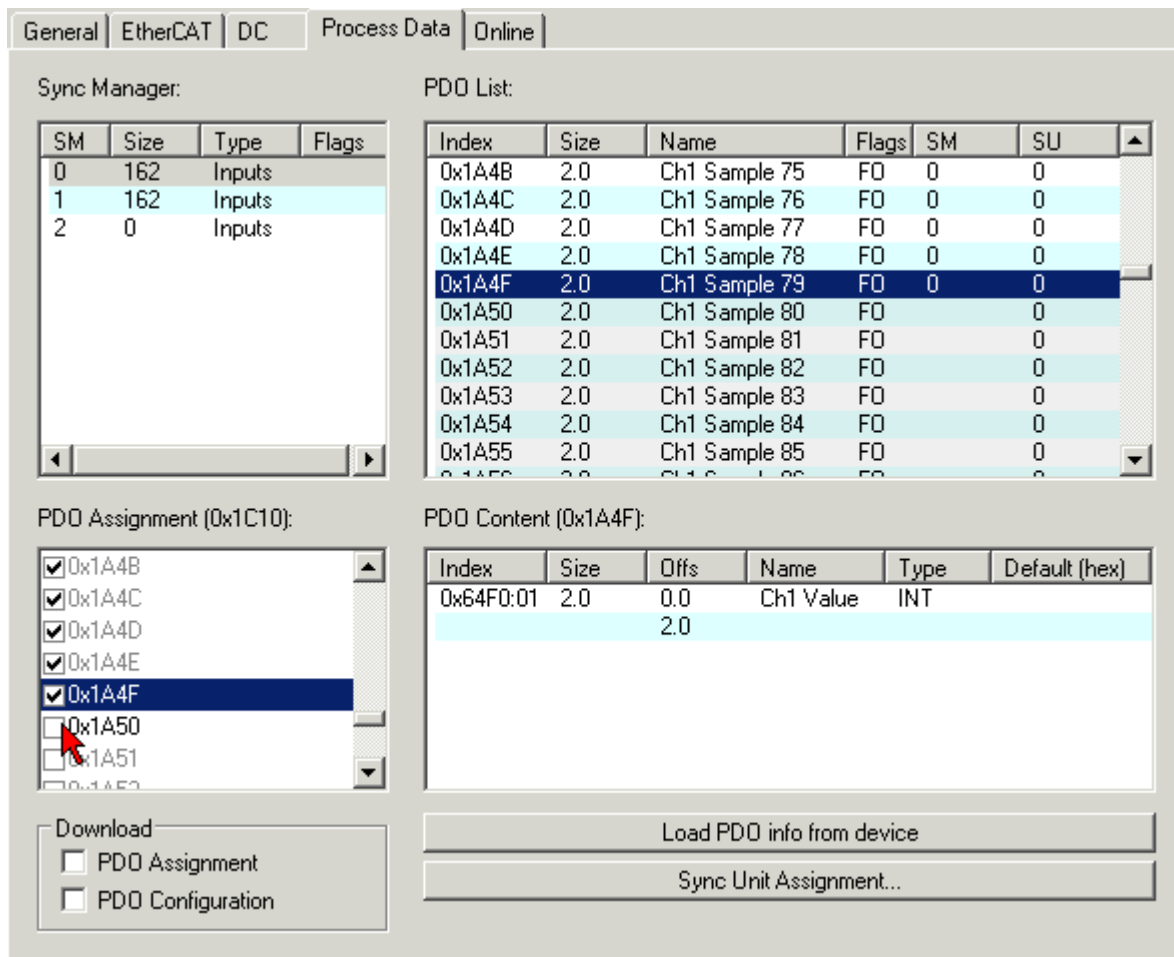


Fig. 193: Process Data tab, adaptation of process data in SM0

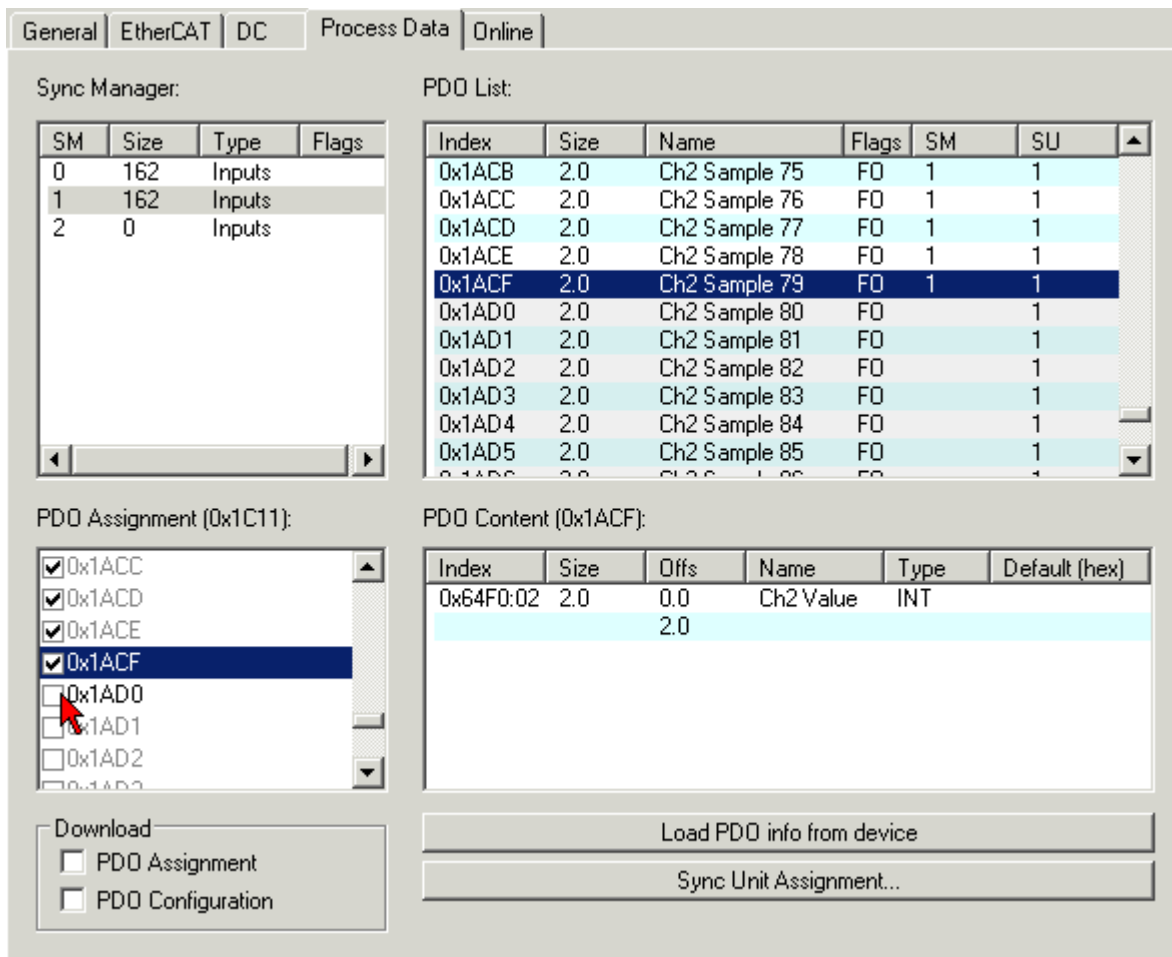


Fig. 194: Process Data tab, adaptation of process data in SM1

Select SM0 and SM1 one after another (Fig. *Process data tab, adjusting the process data in SM0 + SM1*), and in the field below (PDO assignment) select the number of process data corresponding to the oversampling factor, in this example 80_{dec} .

In the PDO list on the right compare the PDO index with the last PDO to be activated under PDO Assignment. In this case all PDOs greater than 0x1A4F (SM0) or 0x1ACF (SM1) have to be deactivated.

● Matching of oversampling factor/process data

i If the number of process data does not match the oversampling factor n , the terminal will reach the OP state, but no process data will be created.

Sample:

Sample for setting a Shift Time:

Under SYNC0 Shift Time User-defined (Fig. *DC tab, activation „Advanced Settings...“, „Distributed Clocks“*), enter a time as required. Any value that may have been created automatically by the EtherCAT master can be overwritten; Although it should be taken into account in the calculation.

Sample: A value of "-5" is preset. An additional shift of $+10 \mu\text{s}$ is required. $-5 + 10 = +5$ is entered as new value.

All time-based/distributed clock-based interrupts (SYNC0/SYNC1) are now offset by this amount relative to other EtherCAT slaves.

6.6 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.

6.6.1 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.

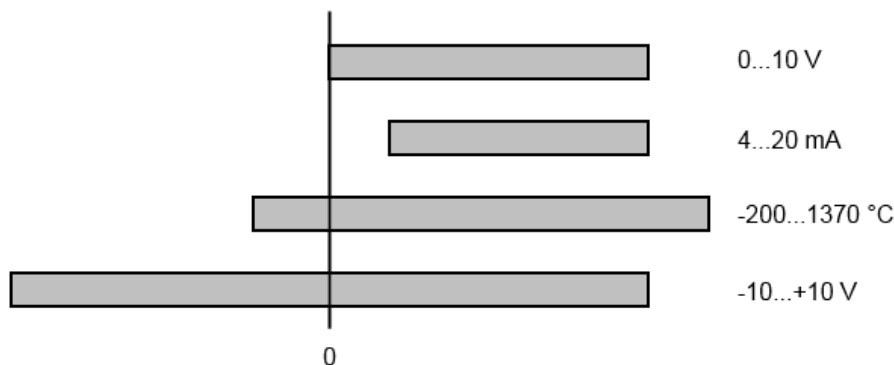


Fig. 195: Full scale value, measuring span

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).

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

The maximum deviation can also be specified directly.

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

● Lower measuring error

i 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.

6.6.3 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

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

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

6.6.4 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.

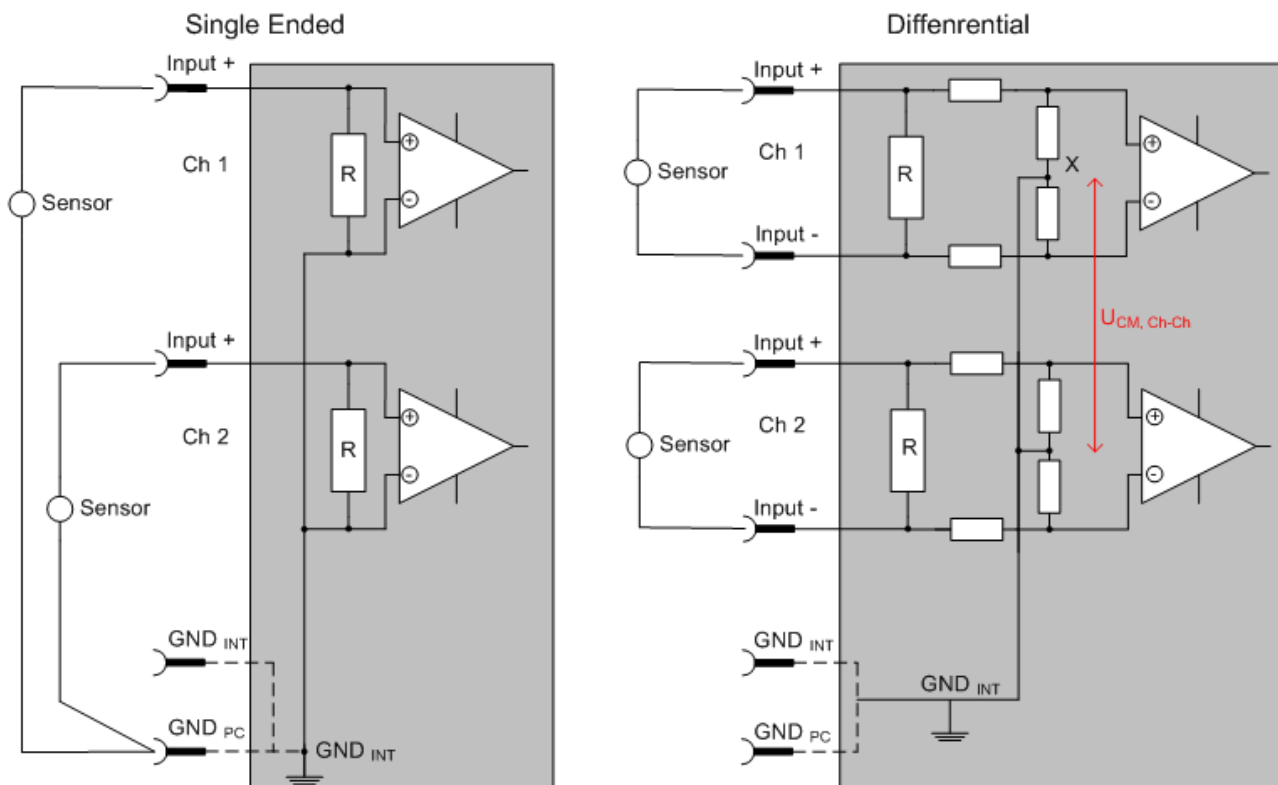


Fig. 196: SE and DIFF module as 2-channel version

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.

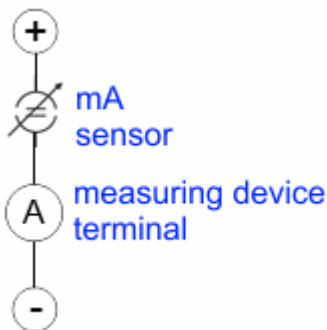


Fig. 197: 2-wire connection

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.

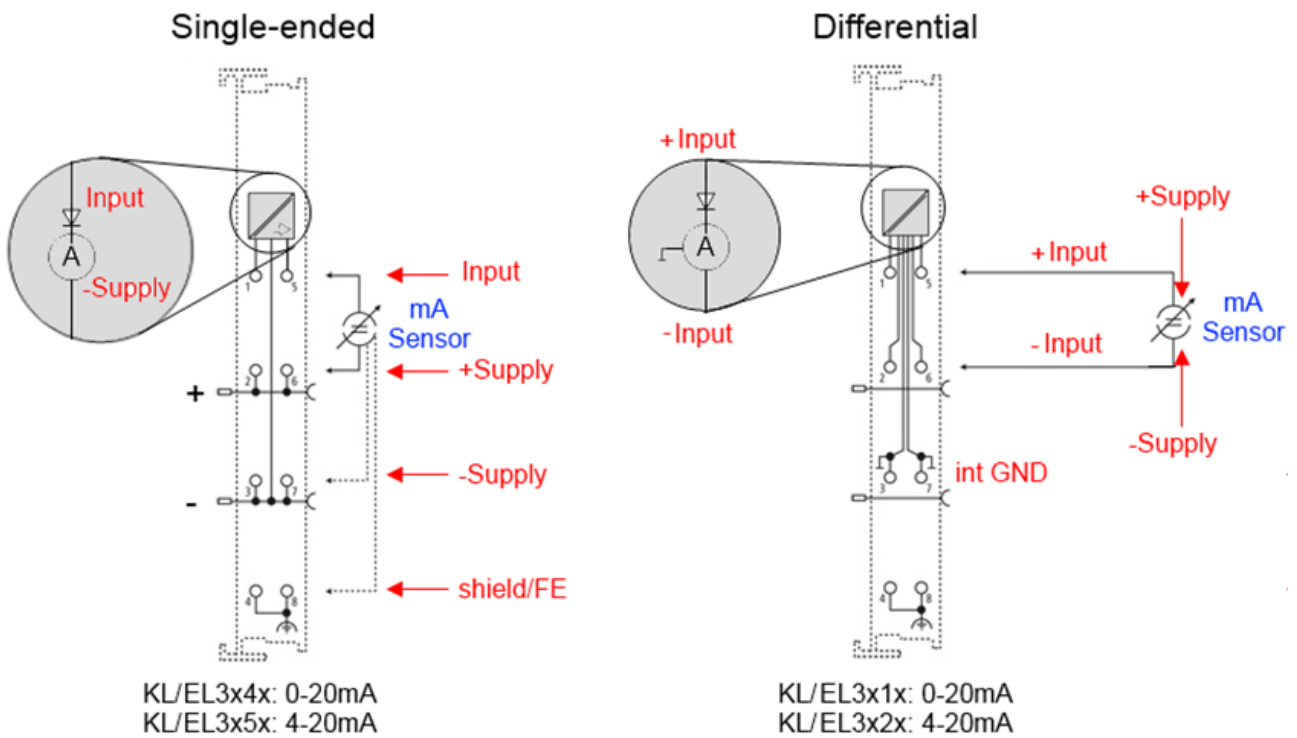


Fig. 198: Connection of externally supplied sensors

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.

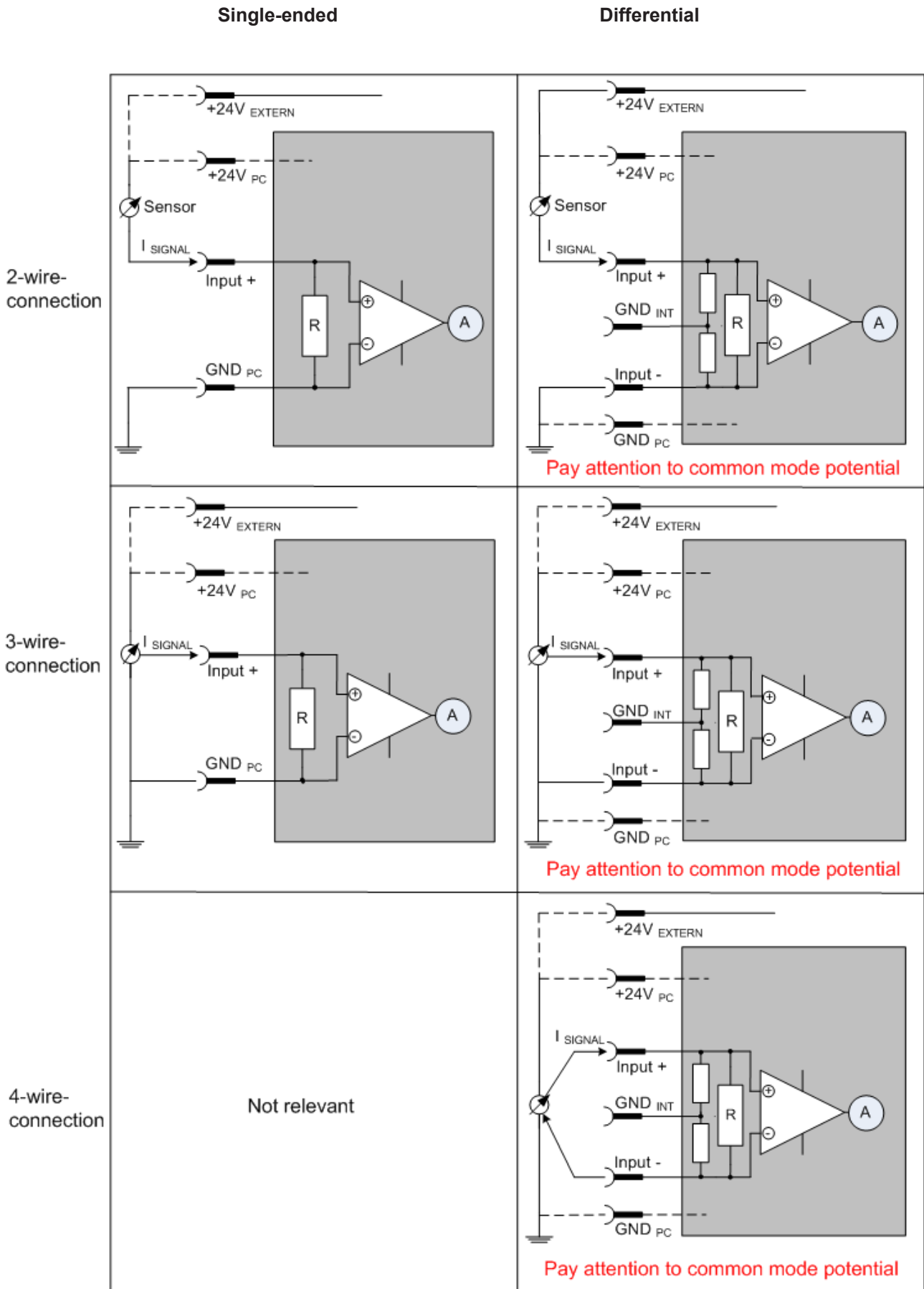


Fig. 199: 2-, 3- and 4-wire connection at single-ended and differential inputs

6.6.5 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.

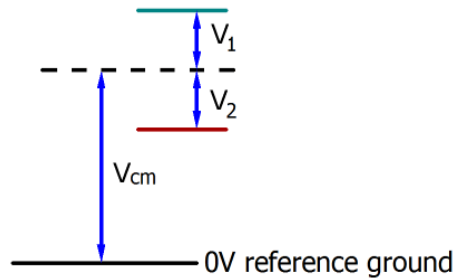


Fig. 200: Common-mode voltage (V_{cm})

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

6.6.6 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

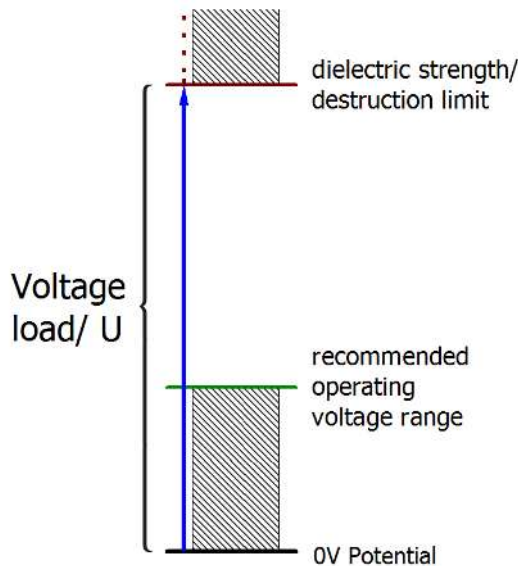


Fig. 201: recommended operating voltage range

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)

6.6.7 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.)

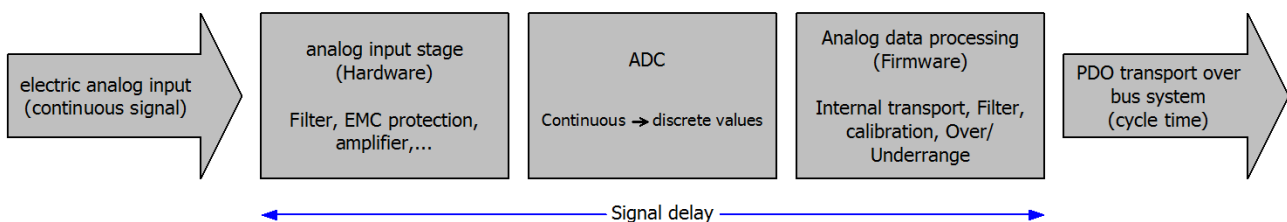


Fig. 202: Signal processing analog input

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, μ s]

= 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, μ s] 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.

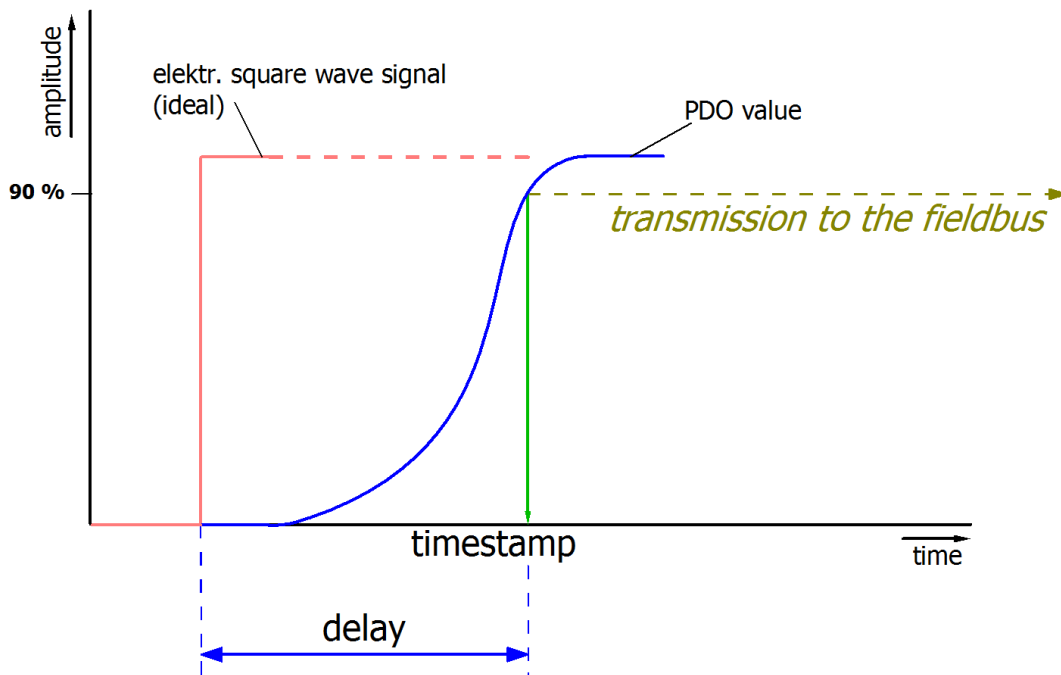


Fig. 203: Diagram signal delay (step response)

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.

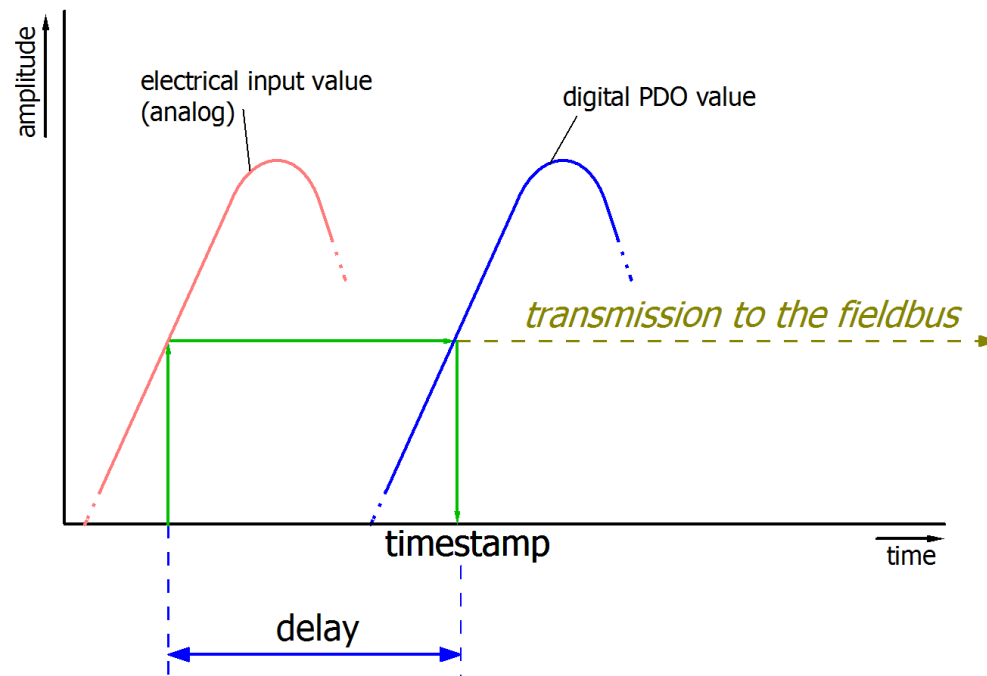


Fig. 204: Diagram signal delay (linear)

3. Additional information:

may be provided in the specification, e.g.

3.1 Actual sampling rate of the ADC (if different from the channel sampling rate)

3.2 Time correction values for run times with different filter settings

...

7 Appendix

7.1 EtherCAT AL Status Codes

For detailed information please refer to the [EtherCAT system description](#).

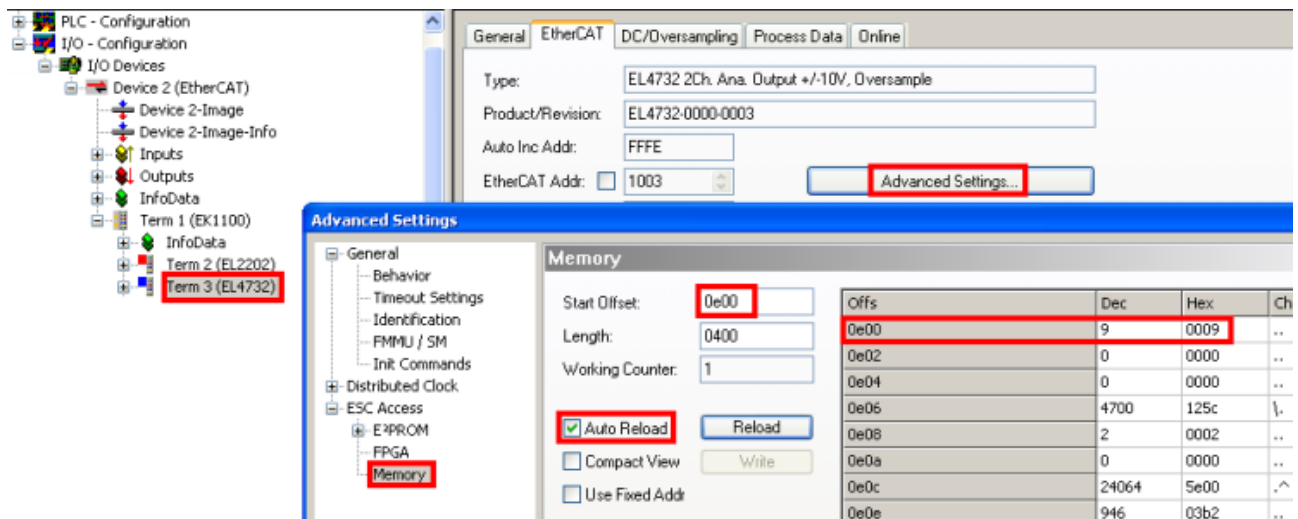
7.2 Firmware information for EL37xx/EL47xx

The firmware version of the EL series EL37xx and EL47xx is shown

- as a laser-engraved serial number on the housing
- in ESC register 0E00_{hex}

To view the register proceed as follows (see Fig. *Firmware version in EL37xx/EL47xx oversampling terminals*):

- EL terminal in the EtherCAT system, install, set to OP state
- WorkingCounter = 0 check
- set "StartOffset" to "0e00" via "Advanced Settings" -> "Memory"
- use "Reload"/AutoReload to ensure that a current online value is displayed
- The firmware version is shown in 0E00_{hex}



The screenshot shows the configuration software interface. On the left, a tree view shows the configuration structure, with 'Term 3 (EL4732)' selected. The main window displays the 'Advanced Settings' dialog for this terminal. The 'Memory' tab is active, showing the 'Start Offset' set to '0e00', 'Length' set to '0400', and 'Working Counter' set to '1'. The 'Auto Reload' checkbox is checked. A table displays memory addresses and their values in decimal and hex.

Offs	Dec	Hex	Ch
0e00	9	0009	..
0e02	0	0000	..
0e04	0	0000	..
0e06	4700	125c	\.
0e08	2	0002	..
0e0a	0	0000	..
0e0c	24064	5e00	..^
0e0e	946	03b2	..

Fig. 205: Firmware version in EL37xx/EL47xx oversampling terminals

7.3 Firmware compatibility

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

Note

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

NOTE			
Risk of damage to the device!			
Pay attention to the instructions for firmware updates on the separate page [▶ 163] . 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!			

EL3702			
Hardware (HW)	Firmware (FW)	Revision no.	Release date
00 - 04	02	EL3702-0000-0002	2007/11
	03		2008/06
	04		2009/12
05 - 15*	05		2010/11
	06		2011/01
	07		2011/02
	08		2011/06
	09	EL3702-0000-0003	2012/12
	10		2014/02
	11*		2015/03

EL3702-0015			
Hardware (HW)	Firmware (FW)	Revision no.	Release date
00 – 01*		EL3702-0000-0003	2016/04
	11*		2016/06

EL3742			
Hardware (HW)	Firmware (FW)	Revision-Nr.	Release date
02 - 05	05	EL3742-0000-0002	2010/11
	06		2011/01
	07		2011/02
	08		2011/06
05 - 11*	09	EL3742-0000-0003	2012/12
	10		2014/02
	11*		2015/03

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

7.4 Firmware Update EL/ES/EM/EPxxxx

This section describes the device update for Beckhoff EtherCAT slaves from the EL/ES, 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.

7.4.1 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:

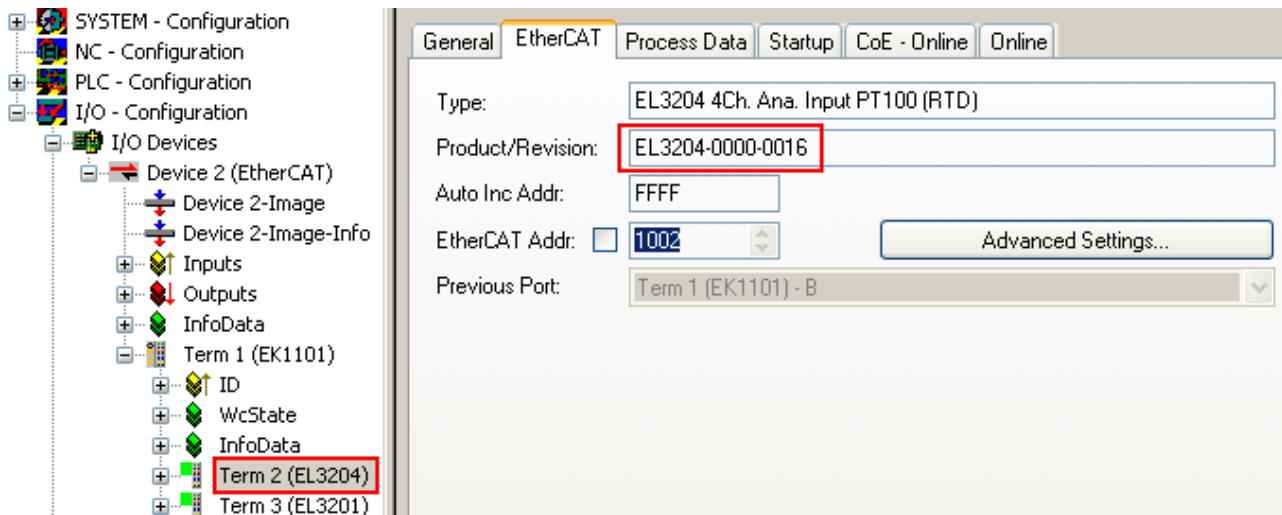


Fig. 206: Device identifier consisting of name EL3204-0000 and revision -0016

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](#).

i Update of XML/ESI description

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

Display of ESI slave identifier

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

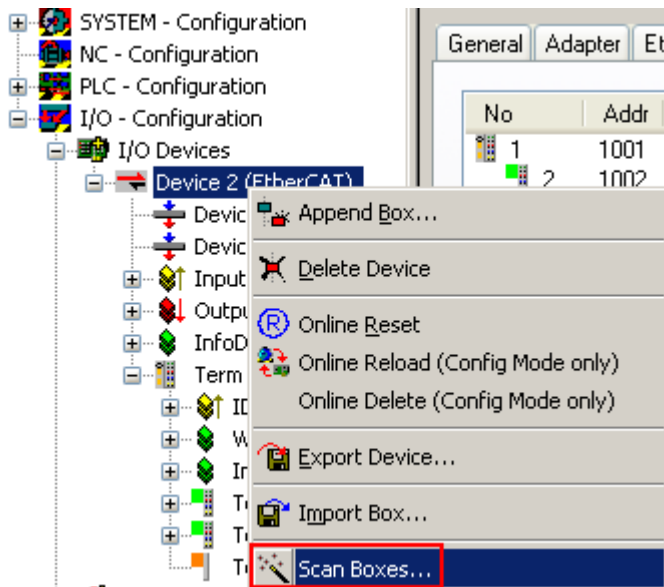


Fig. 207: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 208: Configuration is identical

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

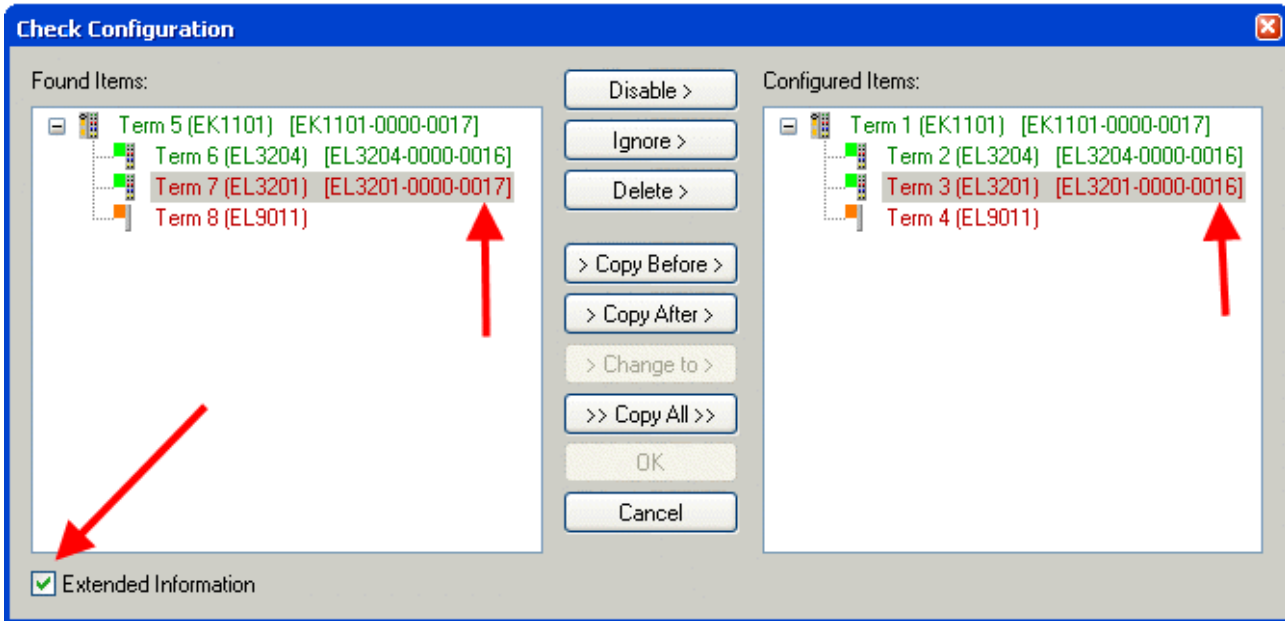


Fig. 209: Change dialog

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*

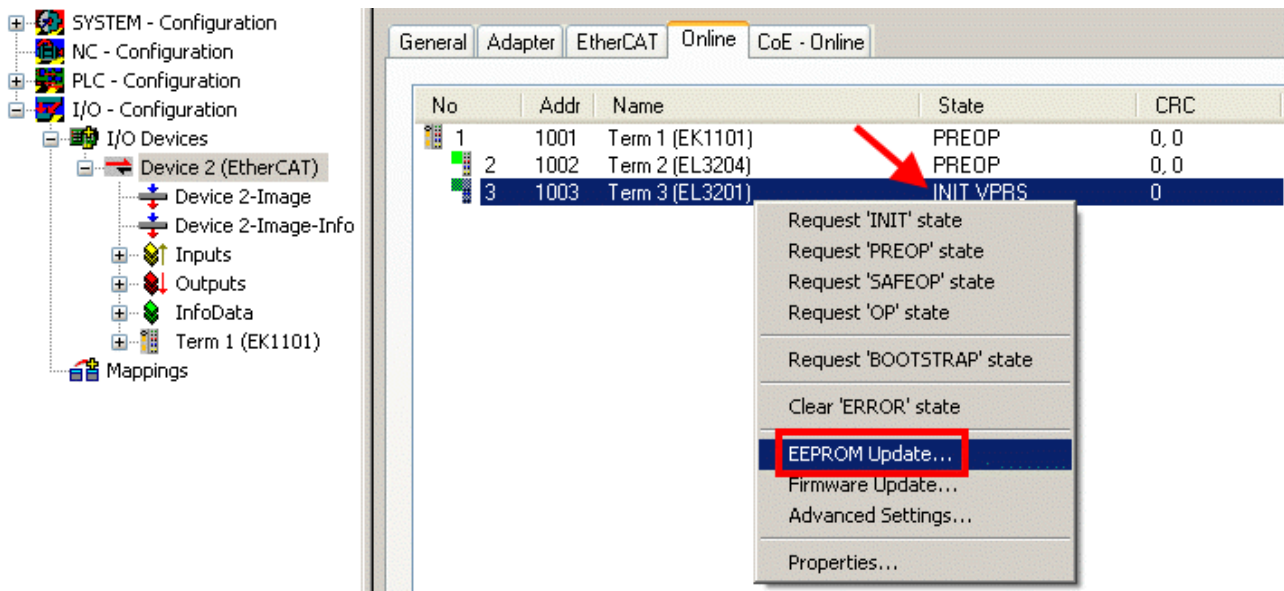


Fig. 210: EEPROM Update

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.

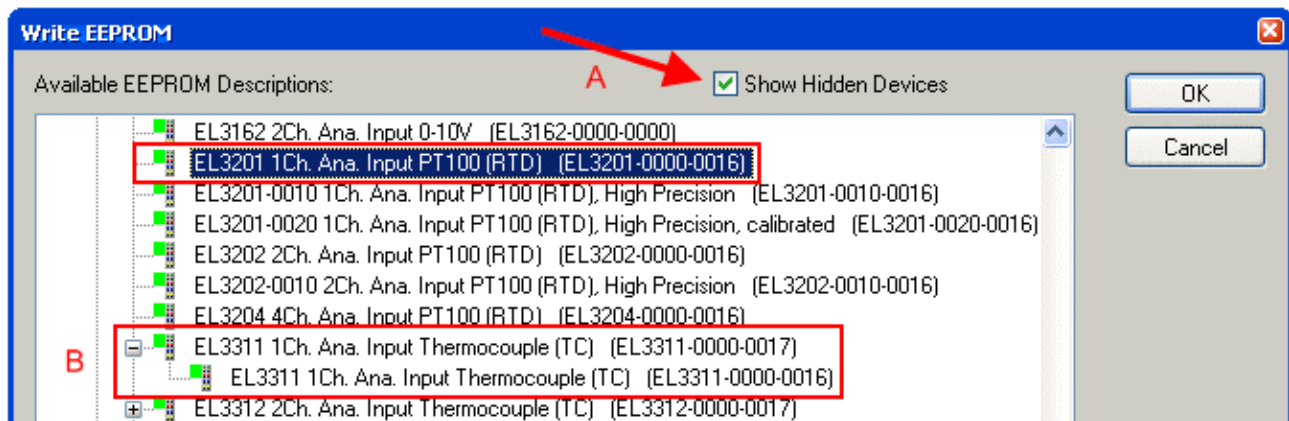


Fig. 211: Selecting the new ESI

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.

7.4.2 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.

The screenshot shows the TwinCAT System Manager interface. On the left, a tree view shows the configuration structure, with 'Term 2 (EL3204)' selected. The main window displays the 'CoE - Online' tab. At the top, there are buttons for 'Update List', 'Advanced...', and 'Online Data'. Below these is a table of CoE entries:

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	HU	> 1 <

Below the table, the 'Advanced Settings' dialog is open, showing the 'Dictionary' section. The 'Online' radio button is selected, and the 'All Objects' option is highlighted in the list below.

Fig. 212: Display of EL3204 firmware version

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

7.4.3 Updating controller firmware *.efw

● CoE directory

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

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

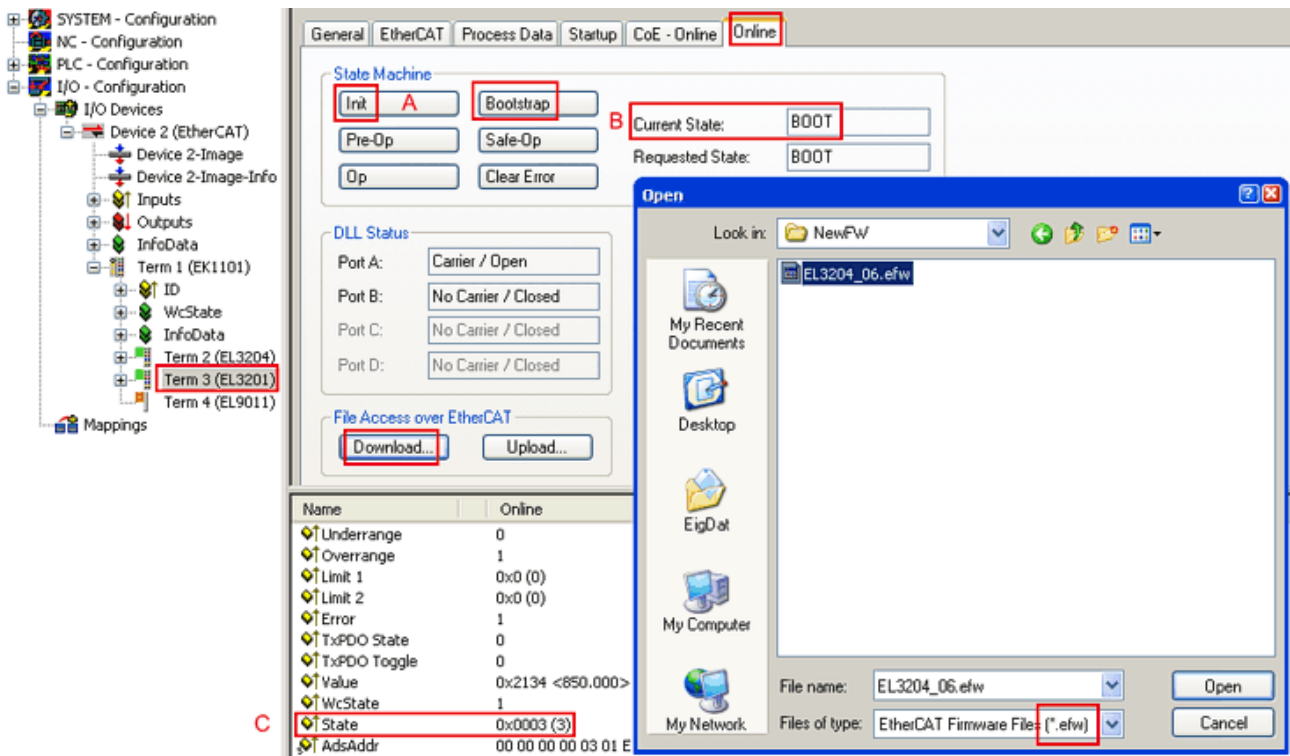
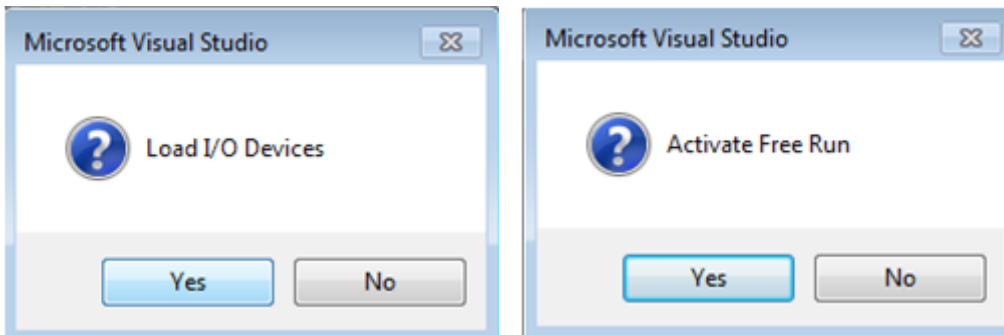


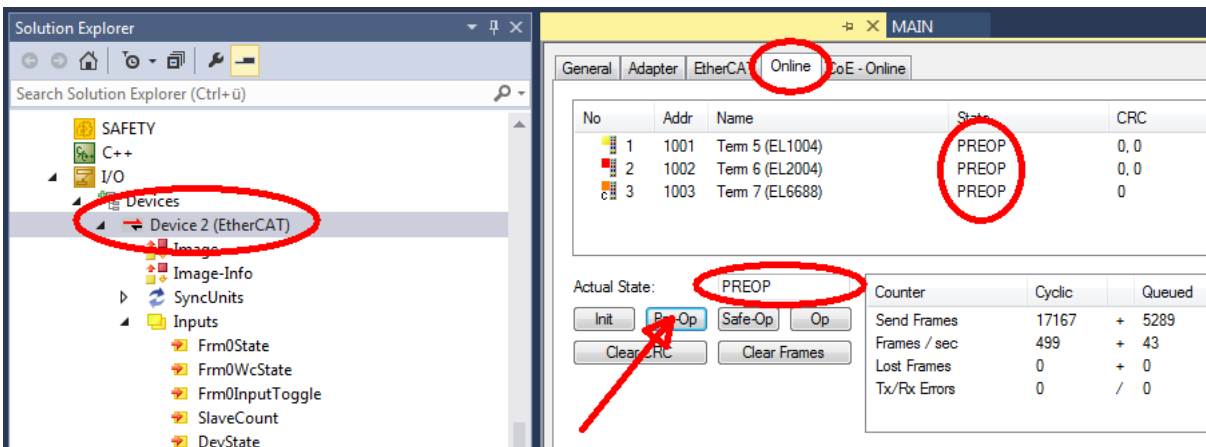
Fig. 213: Firmware Update

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.

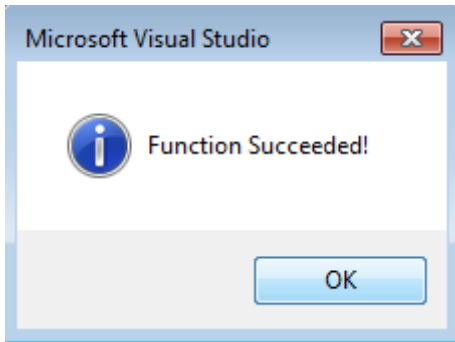


- Switch EtherCAT Master to PreOP



- 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.

7.4.4 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.

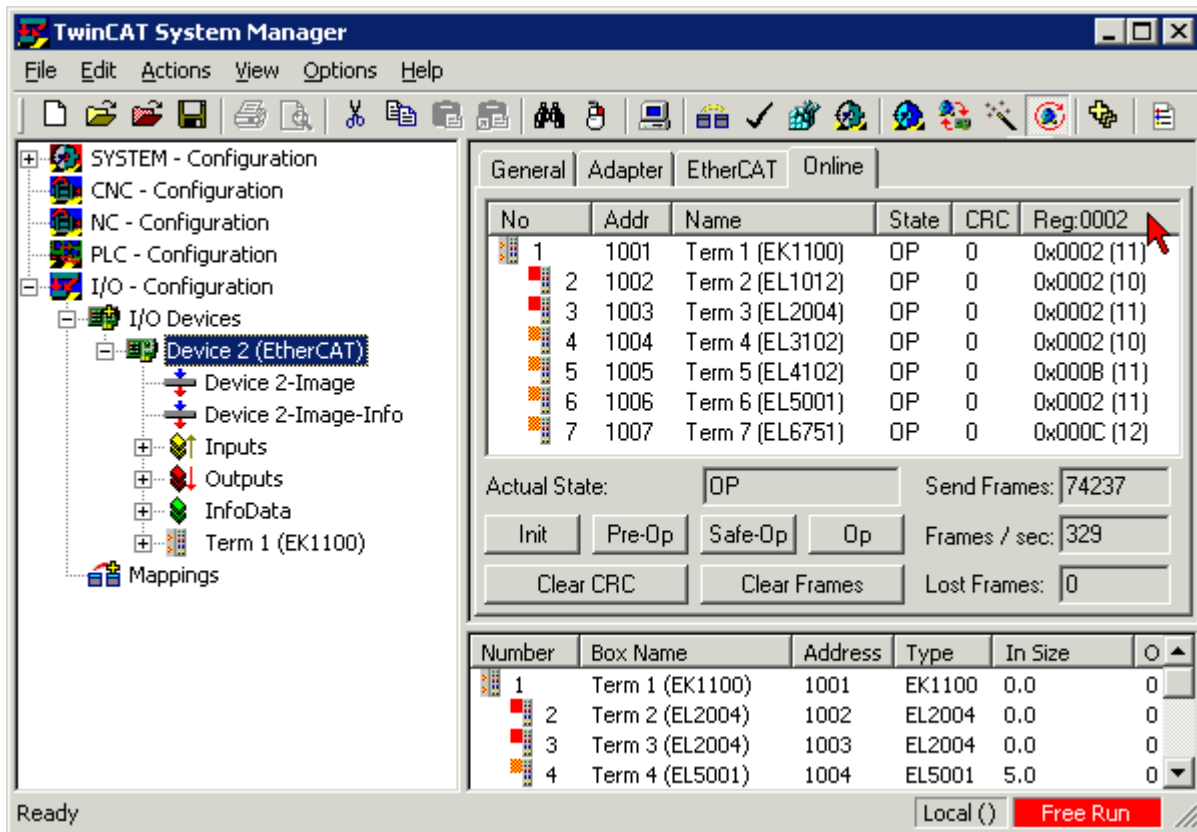


Fig. 214: FPGA firmware version definition

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

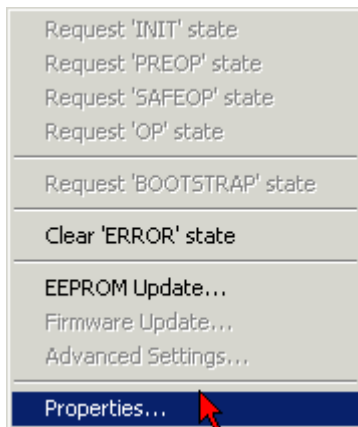


Fig. 215: Context menu *Properties*

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.

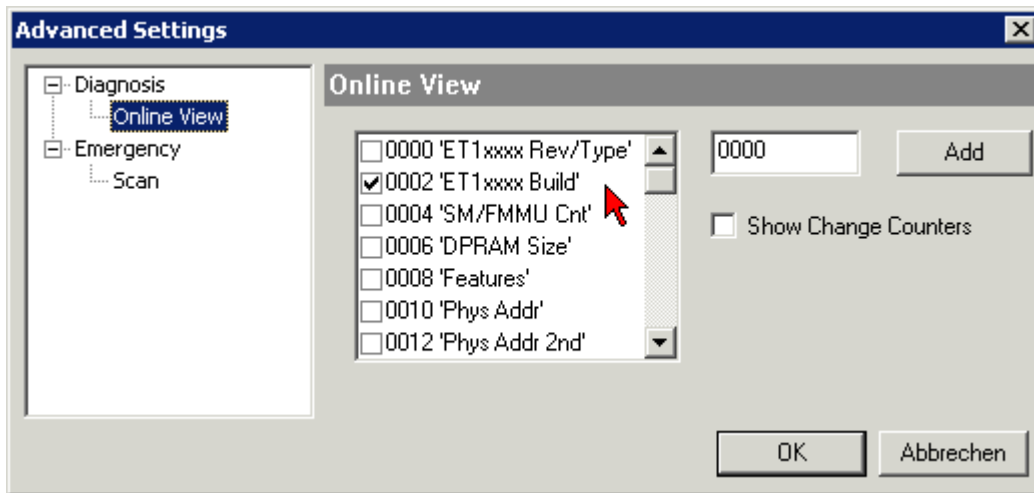


Fig. 216: Dialog *Advanced Settings*

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.

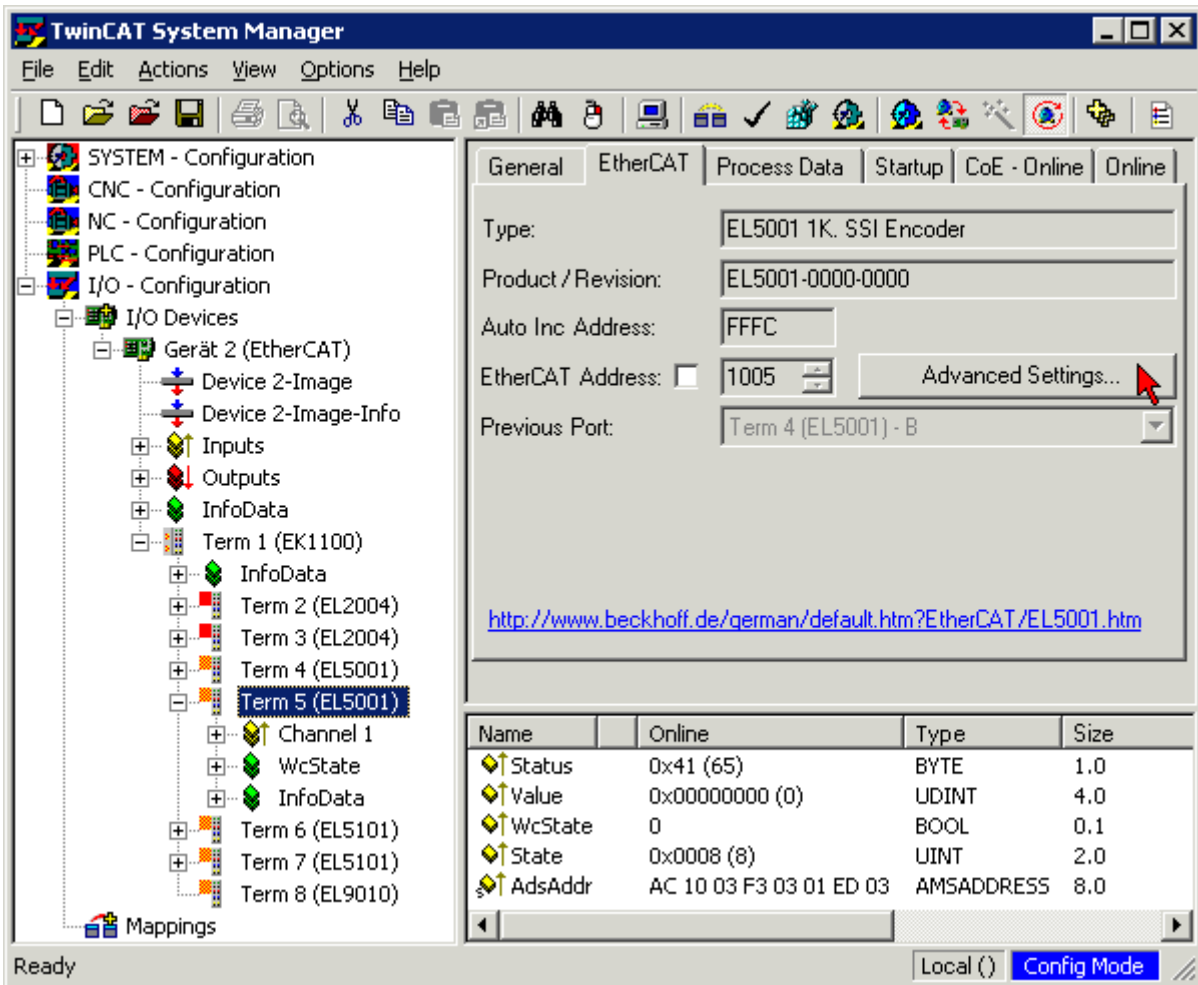
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

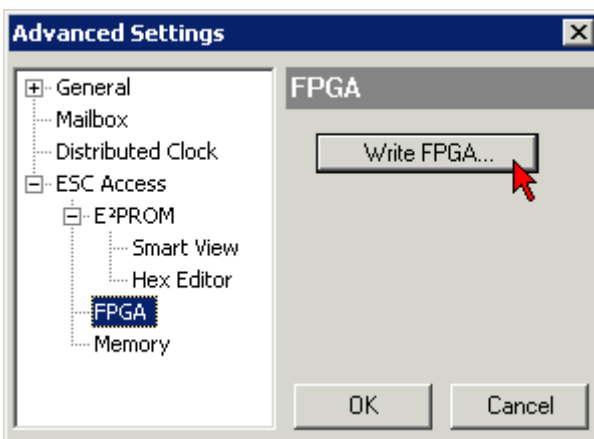
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- 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.

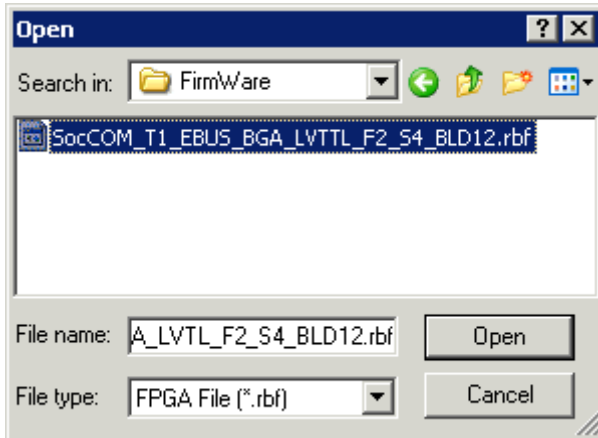
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



- 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!

7.4.5 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.

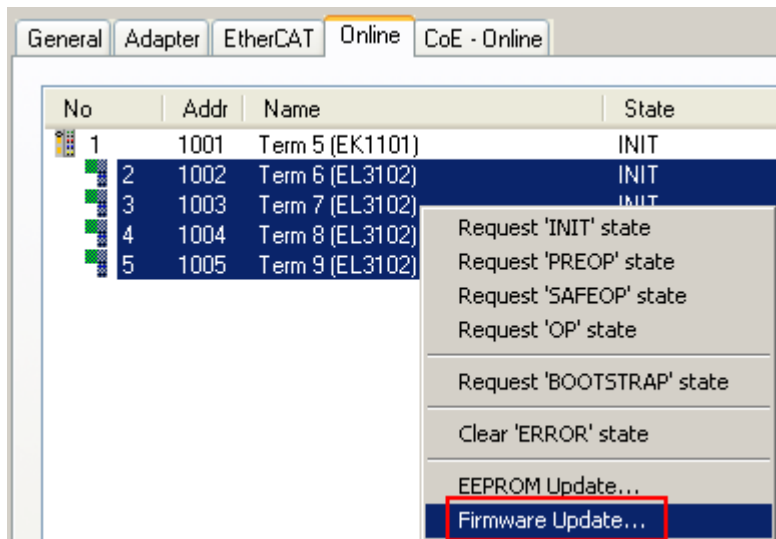


Fig. 217: Multiple selection and firmware update

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

7.5 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).....	11
Fig. 2	EK1100 EtherCAT coupler, standard IP20 IO device with serial/ batch number.....	11
Fig. 3	CU2016 switch with serial/ batch number.....	12
Fig. 4	EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418	12
Fig. 5	EP1258-00001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102.....	12
Fig. 6	EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070	12
Fig. 7	EL2904 IP20 safety terminal with batch number/ date code 50110302 and unique serial number 00331701.....	13
Fig. 8	ELM3604-0002 terminal with unique ID number (QR code) 100001051 and serial/ batch number 44160201.....	13
Fig. 9	EL3702	14
Fig. 10	EL3742	16
Fig. 11	Optional process record StartTimeNextLatch.....	20
Fig. 12	Temporal relationship between SYNC signals and SyncManager interrupt.....	21
Fig. 13	Connection for sample program	23
Fig. 14	Searching the Ethernet adapter.....	24
Fig. 15	Selection and confirmation of the Ethernet adapter.....	24
Fig. 16	Activation of the configuration.....	24
Fig. 17	Confirming the activation of the configuration.....	24
Fig. 18	Generating variable mapping.....	25
Fig. 19	Restarting TwinCAT in RUN mode	25
Fig. 20	Compile project.....	25
Fig. 21	Confirming program start	25
Fig. 22	System manager current calculation	28
Fig. 23	EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog	29
Fig. 24	States of the EtherCAT State Machine.....	31
Fig. 25	Spring contacts of the Beckhoff I/O components.....	34
Fig. 26	Attaching on mounting rail	35
Fig. 27	Disassembling of terminal.....	36
Fig. 28	Power contact on left side.....	37
Fig. 29	Standard wiring.....	38
Fig. 30	Pluggable wiring	39
Fig. 31	High Density Terminals.....	39
Fig. 32	Mounting a cable on a terminal connection	40
Fig. 33	Recommended distances for standard installation position	41
Fig. 34	Other installation positions	42
Fig. 35	Correct positioning.....	43
Fig. 36	Incorrect positioning.....	43
Fig. 37	Internal connection diagram 0/4...20 mA inputs	47
Fig. 38	Internal connection diagram for 0/4...20 mA inputs of a EL3xx2	48
Fig. 39	Example 1: low-side measurement.....	48
Fig. 40	Example 1a, high-side measurement	49
Fig. 41	Example 2, high-side/low-side measurement.....	49

Fig. 42	Invalid EL3xx4 configuration.....	50
Fig. 43	EL37x2 LEDs.....	51
Fig. 44	Pin assignment EL3702, EL3742	52
Fig. 45	Electrical isolation of the inputs	53
Fig. 46	EL3702 - equivalent circuit diagram with variable voltage source	53
Fig. 47	EL3742 - equivalent circuit diagram with variable current source	54
Fig. 48	Relationship between user side (commissioning) and installation.....	56
Fig. 49	Control configuration with Embedded PC, input (EL1004) and output (EL2008)	57
Fig. 50	Initial TwinCAT 2 user interface.....	57
Fig. 51	Selection of the target system	58
Fig. 52	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	58
Fig. 53	Select "Scan Devices..."	59
Fig. 54	Automatic detection of I/O devices: selection the devices to be integrated.....	59
Fig. 55	Mapping of the configuration in the TwinCAT 2 System Manager.....	60
Fig. 56	Reading of individual terminals connected to a device.....	60
Fig. 57	TwinCAT PLC Control after startup	61
Fig. 58	Sample program with variables after a compile process (without variable integration)	62
Fig. 59	Appending the TwinCAT PLC Control project	62
Fig. 60	PLC project integrated in the PLC configuration of the System Manager	63
Fig. 61	Creating the links between PLC variables and process objects	63
Fig. 62	Selecting PDO of type BOOL	64
Fig. 63	Selecting several PDOs simultaneously: activate "Continuous" and "All types".....	64
Fig. 64	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample	65
Fig. 65	Choose target system (remote)	66
Fig. 66	PLC Control logged in, ready for program startup	67
Fig. 67	Initial TwinCAT 3 user interface.....	68
Fig. 68	Create new TwinCAT project.....	68
Fig. 69	New TwinCAT3 project in the project folder explorer	69
Fig. 70	Selection dialog: Choose the target system	69
Fig. 71	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	70
Fig. 72	Select "Scan"	70
Fig. 73	Automatic detection of I/O devices: selection the devices to be integrated.....	71
Fig. 74	Mapping of the configuration in VS shell of the TwinCAT3 environment.....	71
Fig. 75	Reading of individual terminals connected to a device.....	72
Fig. 76	Adding the programming environment in "PLC"	73
Fig. 77	Specifying the name and directory for the PLC programming environment	73
Fig. 78	Initial "Main" program of the standard PLC project.....	74
Fig. 79	Sample program with variables after a compile process (without variable integration)	75
Fig. 80	Start program compilation.....	75
Fig. 81	Creating the links between PLC variables and process objects	76
Fig. 82	Selecting PDO of type BOOL	76
Fig. 83	Selecting several PDOs simultaneously: activate "Continuous" and "All types".....	77
Fig. 84	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample	77
Fig. 85	TwinCAT development environment (VS shell): logged-in, after program startup.....	78
Fig. 86	System Manager "Options" (TwinCAT 2).....	80
Fig. 87	Call up under VS Shell (TwinCAT 3)	80

Fig. 88	Overview of network interfaces	80
Fig. 89	EtherCAT device properties(TwinCAT 2): click on „Compatible Devices...“ of tab “Adapter”	81
Fig. 90	Windows properties of the network interface	81
Fig. 91	Exemplary correct driver setting for the Ethernet port	82
Fig. 92	Incorrect driver settings for the Ethernet port	83
Fig. 93	TCP/IP setting for the Ethernet port	84
Fig. 94	Identifier structure	85
Fig. 95	OnlineDescription information window (TwinCAT 2)	86
Fig. 96	Information window OnlineDescription (TwinCAT 3)	86
Fig. 97	File OnlineDescription.xml created by the System Manager	87
Fig. 98	Indication of an online recorded ESI of EL2521 as an example	87
Fig. 99	Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3).....	87
Fig. 100	Using the ESI Updater (>= TwinCAT 2.11).....	89
Fig. 101	Using the ESI Updater (TwinCAT 3).....	89
Fig. 102	Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)	90
Fig. 103	Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3).....	90
Fig. 104	Selecting the Ethernet port	90
Fig. 105	EtherCAT device properties (TwinCAT 2)	91
Fig. 106	Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3).....	91
Fig. 107	Selection dialog for new EtherCAT device	92
Fig. 108	Display of device revision	92
Fig. 109	Display of previous revisions	93
Fig. 110	Name/revision of the terminal	93
Fig. 111	EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3).....	94
Fig. 112	Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3).....	95
Fig. 113	Scan Devices (left: TwinCAT 2; right: TwinCAT 3).....	95
Fig. 114	Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3).....	95
Fig. 115	Detected Ethernet devices	96
Fig. 116	Example default state	96
Fig. 117	Installing EthetCAT terminal with revision -1018	97
Fig. 118	Detection of EtherCAT terminal with revision -1019	97
Fig. 119	Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)	97
Fig. 120	Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3).....	98
Fig. 121	Scan progressexemplary by TwinCAT 2	98
Fig. 122	Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3).....	98
Fig. 123	Displaying of “Free Run” and “Config Mode” toggling right below in the status bar	98
Fig. 124	TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)	98
Fig. 125	Online display example	99
Fig. 126	Faulty identification	99
Fig. 127	Identical configuration (left: TwinCAT 2; right: TwinCAT 3).....	100
Fig. 128	Correction dialog	100
Fig. 129	Name/revision of the terminal	101
Fig. 130	Correction dialog with modifications	102
Fig. 131	Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3).....	102

Fig. 132 TwinCAT 2 Dialog Change to Alternative Type	102
Fig. 133 Branch element as terminal EL3751	103
Fig. 134 "General" tab.....	103
Fig. 135 „EtherCAT“ tab.....	104
Fig. 136 "Process Data" tab.....	105
Fig. 137 Configuring the process data.....	106
Fig. 138 „Startup“ tab.....	107
Fig. 139 "CoE – Online" tab	108
Fig. 140 Dialog "Advanced settings"	109
Fig. 141 „Online“ tab	109
Fig. 142 "DC" tab (Distributed Clocks).....	110
Fig. 143 Selection of the diagnostic information of an EtherCAT Slave	112
Fig. 144 Basic EtherCAT Slave Diagnosis in the PLC.....	113
Fig. 145 EL3102, CoE directory	115
Fig. 146 Example of commissioning aid for a EL3204	116
Fig. 147 Default behaviour of the System Manager	117
Fig. 148 Default target state in the Slave	117
Fig. 149 PLC function blocks	118
Fig. 150 Illegally exceeding the E-Bus current	119
Fig. 151 Warning message for exceeding E-Bus current	119
Fig. 152 Oversampling PDO of the EL37xx series and in the comparison with EL31xx	120
Fig. 153 Adding a Scope project into an already existing project.....	122
Fig. 154 Representation of a created PLC array variable („aUI_Samples“) to link with oversampling PDOs of EL3773.....	123
Fig. 155 Insertion of a free task	123
Fig. 156 Task property "Create symbols" must be activated.....	124
Fig. 157 Insertion of variable "Var 1" fitting to the oversampling (-factor).....	124
Fig. 158 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)	125
Fig. 159 Select the EL3773 PDO "L1 Voltage Samples" to create a link to the PLC array variable „aUI_Samples“	125
Fig. 160 Selection of the oversampling variable with the target browser.....	126
Fig. 161 Appending the variable "aUI_Samples“ below "axis" within the scope project of the solution explorer.....	126
Fig. 162 Example of recording a sine signal with 10 x oversampling at 1 ms measurement cycle time ...	127
Fig. 163 Activation of the ADS server of the EtherCAT device (TwinCAT 3)	128
Fig. 164 Direct access to PDOs of the terminal by ScopeView	128
Fig. 165 PLC declaration	130
Fig. 166 PLC in the System Manager.....	130
Fig. 167 Add Variable Type	130
Fig. 168 Definition of the variable type	131
Fig. 169 Overview of declared types	131
Fig. 170 Linking	132
Fig. 171 Array variables of an oversampling terminal.....	132
Fig. 172 Settings in the additional task.....	132
Fig. 173 Variable browser up to the array VAR152	133
Fig. 174 AddSymbol on the array	133

Fig. 175	Channel settings	133
Fig. 176	Activation of the marks	134
Fig. 177	Illustration of a 10 x oversampling variable of the EL3751 by the Scope2	135
Fig. 178	Activation of the ADS server of the EtherCAT Device (TwinCAT 2).....	136
Fig. 179	Direct access of the Scope2 to the terminal's PDOs	136
Fig. 180	Automatically calculated array variable (red) in the Scope2.....	137
Fig. 181	TwinCAT tree EL37x2.....	138
Fig. 182	WCState, Online tab	139
Fig. 183	InfoData, Online tab.....	139
Fig. 184	Process data tab SM0, EL37x2	141
Fig. 185	Process data tab SM1, EL37x2	142
Fig. 186	Process data tab SM2, EL37x2	143
Fig. 187	Process Data tab, example setting two-channel, 2 x oversampling	145
Fig. 188	TwinCAT tree, sample setting 2-channel, 2-times oversampling.....	145
Fig. 189	Process Data tab, activation of index 0x1b10, entry "StartTimeNextLatch"	146
Fig. 190	TwinCAT tree, entry "StartTimeNextLatch"	147
Fig. 191	DC tab, activation "Advanced Settings..."	147
Fig. 192	DC tab, activation "Advanced Settings...", "Distributed Clocks"	148
Fig. 193	Process Data tab, adaptation of process data in SM0.....	149
Fig. 194	Process Data tab, adaptation of process data in SM1.....	150
Fig. 195	Full scale value, measuring span	151
Fig. 196	SE and DIFF module as 2-channel version	153
Fig. 197	2-wire connection.....	155
Fig. 198	Connection of externally supplied sensors	156
Fig. 199	2-, 3- and 4-wire connection at single-ended and differential inputs	157
Fig. 200	Common-mode voltage (Vcm).....	158
Fig. 201	recommended operating voltage range	159
Fig. 202	Signal processing analog input.....	159
Fig. 203	Diagram signal delay (step response)	161
Fig. 204	Diagram signal delay (linear)	161
Fig. 205	Firmware version in EL37xx/EL47xx oversampling terminals	162
Fig. 206	Device identifier consisting of name EL3204-0000 and revision -0016	165
Fig. 207	Scan the subordinate field by right-clicking on the EtherCAT device	165
Fig. 208	Configuration is identical	166
Fig. 209	Change dialog	166
Fig. 210	EEPROM Update	167
Fig. 211	Selecting the new ESI.....	167
Fig. 212	Display of EL3204 firmware version	168
Fig. 213	Firmware Update	169
Fig. 214	FPGA firmware version definition	171
Fig. 215	Context menu Properties	171
Fig. 216	Dialog Advanced Settings	172
Fig. 217	Multiple selection and firmware update	174