



Documentation

EL32xx-0xx0

Analog Input Terminals Pt100 (RTD)

Version: 5.6
Date: 2019-06-05

BECKHOFF

Table of contents

1 Foreword	5
1.1 Overview Analog Input Terminals Pt100 (RTD)	5
1.2 Notes on the documentation.....	5
1.3 Safety instructions	7
1.4 Documentation issue status	8
1.5 Version identification of EtherCAT devices	10
2 Product overview.....	14
2.1 EL3201, EL3202, EL3204, EL3214, EL3208 - Introduction.....	14
2.2 EL3214-0090 - Introduction	18
2.3 Technology RTD measuring.....	19
2.4 Basic principles of RTD technology	23
2.5 Technical data	26
2.6 Notes to the factory calibration certificate.....	31
2.7 Start	32
3 Basics communication	33
3.1 EtherCAT basics.....	33
3.2 EtherCAT cabling – wire-bound.....	33
3.3 General notes for setting the watchdog.....	34
3.4 EtherCAT State Machine	36
3.5 CoE Interface.....	38
3.6 Distributed Clock	43
4 Installation.....	44
4.1 Safety instructions	44
4.2 Environmental conditions	44
4.3 Transport / storage	44
4.4 Control cabinet / terminal box.....	44
4.5 Instructions for ESD protection.....	45
4.6 Installation on mounting rails	46
4.7 Installation instructions for enhanced mechanical load capacity	49
4.8 Positioning of passive Terminals	50
4.9 Installation positions	51
4.10 Connection	53
4.10.1 Connection system	53
4.10.2 Wiring.....	55
4.10.3 Shielding	56
4.11 ATEX - Special conditions (standard temperature range).....	57
4.12 ATEX - Special conditions (extended temperature range)	58
4.13 ATEX Documentation	59
4.14 UL notice	59
4.15 LEDs and connection	60
4.16 Connection of analog RTD signal lines	69
5 Commissioning.....	72
5.1 TwinCAT Quick Start	72

5.1.1	TwinCAT 2	75
5.1.2	TwinCAT 3	85
5.2	TwinCAT Development Environment	97
5.2.1	Installation of the TwinCAT real-time driver	97
5.2.2	Notes regarding ESI device description	103
5.2.3	TwinCAT ESI Updater	107
5.2.4	Distinction between Online and Offline	107
5.2.5	OFFLINE configuration creation	108
5.2.6	ONLINE configuration creation	113
5.2.7	EtherCAT subscriber configuration	121
5.3	General Notes - EtherCAT Slave Application	130
5.4	Settings and application notes for EL32xx-0010/0020	138
5.5	Adjustments and application notes for EL3204-0200	141
5.6	Process data and operating modes	149
5.7	TwinSAFE SC	154
5.7.1	TwinSAFE SC - operating principle	154
5.7.2	TwinSAFE SC - configuration	154
5.8	TwinSAFE SC process data EL3214-0090	158
5.9	CoE object description EL32xx-00xx	158
5.10	CoE object description EL3204-0200	168
5.11	Status word	176
5.12	Notices on analog specifications	181
5.12.1	Full scale value (FSV)	181
5.12.2	Measuring error/ measurement deviation	181
5.12.3	Temperature coefficient tK [ppm/K]	182
5.12.4	Single-ended/differential typification	183
5.12.5	Common-mode voltage and reference ground (based on differential inputs)	188
5.12.6	Dielectric strength	188
5.12.7	Temporal aspects of analog/digital conversion	189
6	Appendix	193
6.1	EtherCAT AL Status Codes	193
6.2	Calibration certificate	193
6.3	Firmware compatibility	194
6.4	Firmware Update EL/ES/EM/ELM/EPxxxx	196
6.4.1	Device description ESI file/XML	197
6.4.2	Firmware explanation	200
6.4.3	Updating controller firmware *.efw	201
6.4.4	FPGA firmware *.rbf	202
6.4.5	Simultaneous updating of several EtherCAT devices	206
6.5	Restoring the delivery state	207
6.6	Support and Service	208

1 Foreword

1.1 Overview Analog Input Terminals Pt100 (RTD)

EL3201, EL3202, EL3204 [▶ 14]	1, 2, 4 Channel Input Terminals Pt100 (RTD)
EL3201-0010, EL3202-0010 [▶ 14]	1, 2 Channel Input Terminals Pt100 (RTD), high precision
EL3201-0020, EL3202-0020 [▶ 14]	1, 2 Channel Input Terminals Pt100 (RTD), high precision with calibration certificate
EL3204-0200 [▶ 14]	4 Channel Input Terminal for RTD up to 240 kΩ, NTC 20 k, 16 bit
EL3214 [▶ 16]	4 Channel HD Input Terminal Pt100 for 3-wire connection
EL3214-0090 [▶ 18]	4 Channel HD Input Terminal Pt100, 3-wire connection, TwinSAFE Single Channel
EL3208 [▶ 17]	8 Channel HD Input Terminal Pt100 (RTD)

1.2 Notes on the documentation

Intended audience

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

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

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

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

Disclaimer

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

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

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

Trademarks

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH.

Other designations used in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owners.

Patent Pending

The EtherCAT Technology is covered, including but not limited to the following patent applications and patents: EP1590927, EP1789857, DE102004044764, DE102007017835 with corresponding applications or registrations in various other countries.

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



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

Copyright

© Beckhoff Automation GmbH & Co. KG, Germany.

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

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

1.3 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

DANGER

Serious risk of injury!

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

WARNING

Risk of injury!

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

CAUTION

Personal injuries!

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

NOTE

Damage to environment/equipment or data loss

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



Tip or pointer

This symbol indicates information that contributes to better understanding.

1.4 Documentation issue status

Version	Comment
5.6	<ul style="list-style-type: none"> • Update chapter "Technical data" • Update structure
5.5	<ul style="list-style-type: none"> • Update chapter "Technical data" • Update UL notice
5.4	<ul style="list-style-type: none"> • Update chapter "Technical data" • Update chapter " Object description EL32xx-00xx" • Update structure • Revision version updated
5.3	<ul style="list-style-type: none"> • Update chapter "Technical data" • Update chapter "EL3214-0090 - Introduction" and chapter " Object description EL32xx-00xx" • Update chapter "TwinSAFE SC" • Update structure • Revision version updated
5.2	<ul style="list-style-type: none"> • Update chapter "Object description", "Function" • Structural update • Revision version updated
5.1	<ul style="list-style-type: none"> • Update chapter "Technical data" • Structural update • Revision version updated
5.0	<ul style="list-style-type: none"> • Addenda EL3214-0090 • Update chapter "Technical data" • Structural update • Revision version updated
4.9	<ul style="list-style-type: none"> • Update chapter "Technical data" • Addenda chapter "Instructions for ESD protection" • Chapter "Analog technical notices - specifications" replaced by chapter "Notices on analog specifications"
4.8	<ul style="list-style-type: none"> • Update chapter "Technology RTD measuring" • Correction revision status
4.7	<ul style="list-style-type: none"> • Update chapter "Notes on the documentation" • Correction of Technical data • Update chapter "TwinCAT 2.1x" -> "TwinCAT Development Environment" and "TwinCAT Quick Start" • Correction revision status
4.6	<ul style="list-style-type: none"> • "Technical data" section updated • Revision version updated
4.5	<ul style="list-style-type: none"> • Update section "Settings and application notes for EL32xx-0010/0020" • "Notes to the factory calibration certificate" inserted within section "Product overview"
4.4	<ul style="list-style-type: none"> • "Technical data" section updated • Structural update • Revision version updated
4.3	<ul style="list-style-type: none"> • "Technology RTD measuring" section updated • Structural update • Revision version updated
4.2	<ul style="list-style-type: none"> • "Technical data" section updated • Structural update • Revision version updated
4.1	<ul style="list-style-type: none"> • "LED and connection" section updated

Version	Comment
4.0	<ul style="list-style-type: none"> • Transfer SCHEMA ST4 • EL3208 added • EL3214 added • "Technical data" section updated • Structural update • Revision version updated
3.1	<ul style="list-style-type: none"> • "Technical data" section updated • "Assembly instructions with increased mechanical load capacity" section supplemented • Structural update • Revision version updated
3.0	<ul style="list-style-type: none"> • Structural update
2.9	<ul style="list-style-type: none"> • EL3204-0200 supplemented • Structural update
2.8	<ul style="list-style-type: none"> • "Technology" section updated
2.7	<ul style="list-style-type: none"> • "RTD measurement technology" section updated, EL3204-0200 supplemented
2.6	<ul style="list-style-type: none"> • "Technology" section updated
2.5	<ul style="list-style-type: none"> • "RTD technology" section added
2.4	<ul style="list-style-type: none"> • "Technology" section updated
2.3	<ul style="list-style-type: none"> • Note on the connection of analog signal conductors supplemented
2.2	<ul style="list-style-type: none"> • Note on calibration certificate supplemented • Structural update
2.1	<ul style="list-style-type: none"> • EL3202-0020 added
2.0	<ul style="list-style-type: none"> • Data flow schematic supplemented, technical notes supplemented
1.9	<ul style="list-style-type: none"> • Technical notes supplemented
1.8	<ul style="list-style-type: none"> • Technical notes supplemented
1.7	<ul style="list-style-type: none"> • Technical data supplemented and corrected
1.6	<ul style="list-style-type: none"> • Technical corrections, EL3202-0010 supplemented
1.5	<ul style="list-style-type: none"> • KT/KTY characteristics added, limit supplemented
1.4	<ul style="list-style-type: none"> • Technical changes added
1.3	<ul style="list-style-type: none"> • Technical changes added
1.2	<ul style="list-style-type: none"> • EL3201-0010, EL3201-0020 added
1.1	<ul style="list-style-type: none"> • Technical data corrected
1.0	<ul style="list-style-type: none"> • Technical data supplemented, first publication

1.5 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

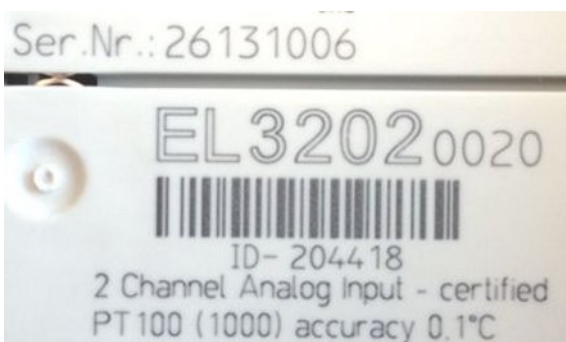


Fig. 4: EL3202-0020 with serial/ batch number 26131006 and unique ID-number 204418

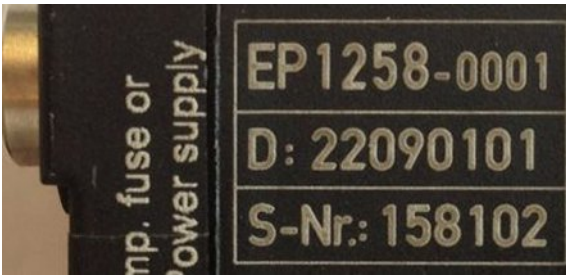


Fig. 5: EP1258-0001 IP67 EtherCAT Box with batch number/ date code 22090101 and unique serial number 158102



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

2 Product overview

2.1 EL3201, EL3202, EL3204, EL3214, EL3208 - Introduction

1-, 2- and 4-channel analog input terminals Pt100 (RTD)

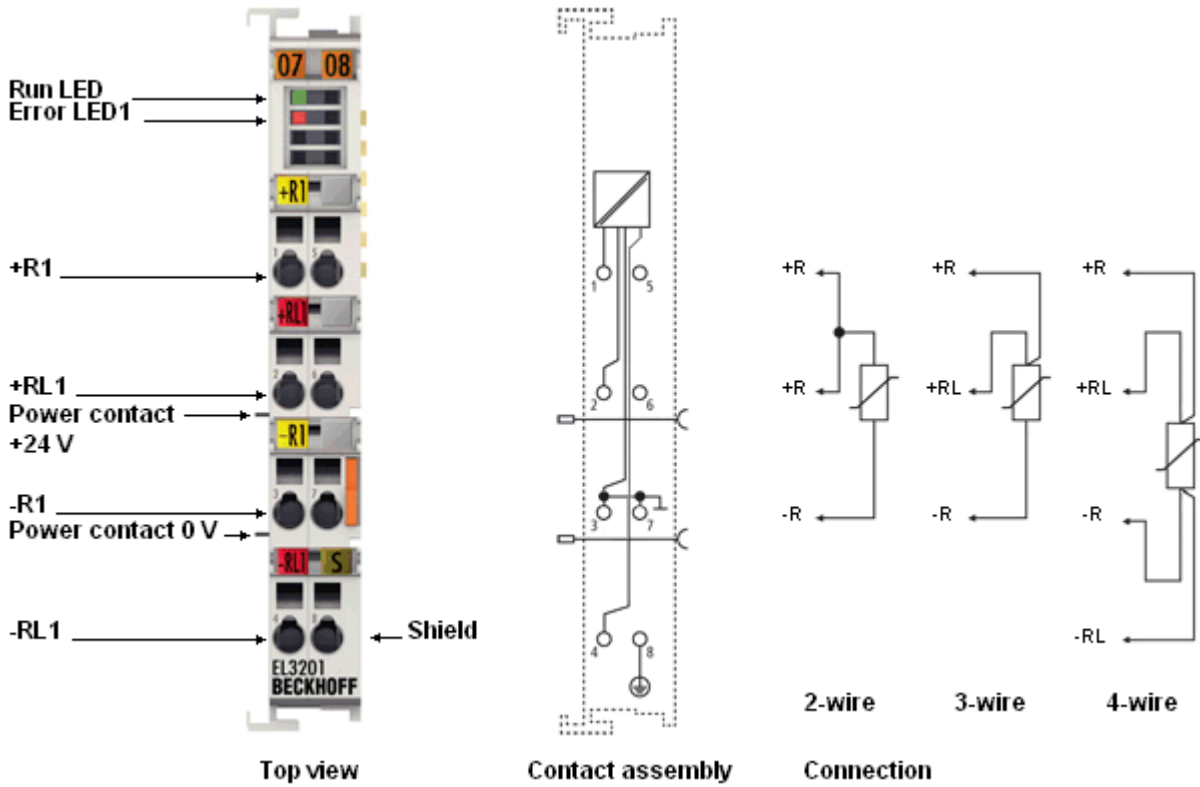


Fig. 9: EL3201

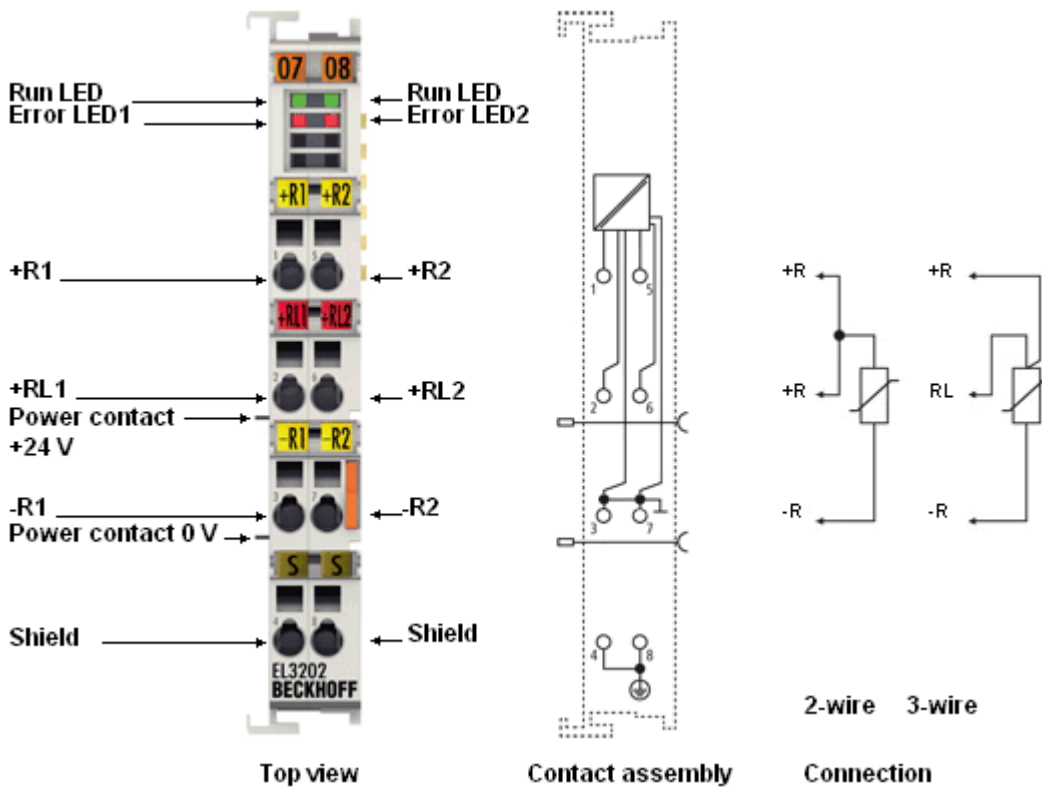


Fig. 10: EL3202

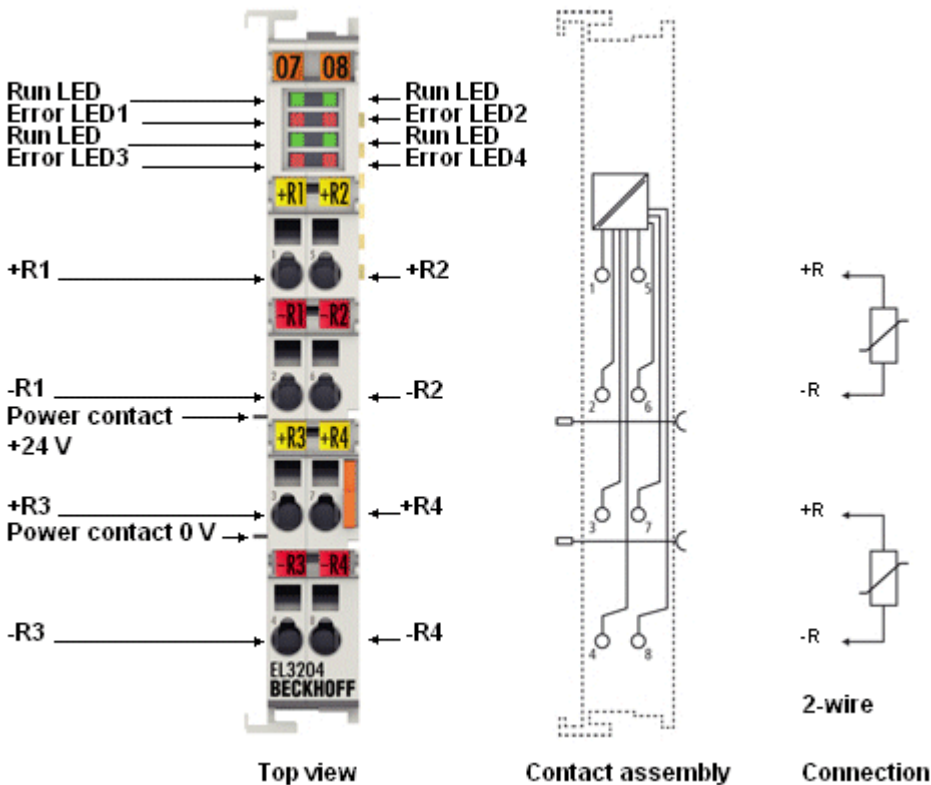


Fig. 11: EL3204

The EL320x analog input terminals allow resistance sensors to be connected directly. The EL3204 circuitry can operate 2-wire sensors, the EL3202 2 and 3-wire sensors, and the EL3201 additionally can operate 4-wire sensors. Several characteristic sensor curves (Pt100, Pt1000, NI120, NI1000, KTY types and others) are supported.

The EL3201-0010 is the high-precision version of the single-channel terminal (EL3202-0010, two-channel) with a measuring error $< \pm 0.1 \text{ }^\circ\text{C}$ using Pt sensors (ambient temperature of $40 \text{ }^\circ\text{C}$). These terminals can operate 4-wire sensors. The high-precision EL3201-0020 and EL3202-0020 will be delivered with a calibration certificate [► 193] in addition.

The terminals from the EL320x-xxxx series can measure the temperature at the measuring point or directly output the resistance values of the sensors. For temperature measurements the temperature value is calculated via the characteristic curves stored in the terminal.

The EL3204-0200 can be used with any RTD in the range of 100 Ohm to 240 kOhm. The resistance values are converted (linearization) in the terminal either based on preset characteristic curves, conversion formulas with specific material parameters (e.g. according to IEC 60751, Steinhart-Hart equation, B-parameter equation), or a freely programmable conversion table.

The EtherCAT Terminals indicate their signal state by means of light emitting diodes. Sensor malfunctions such as broken wires are indicated by error LEDs.

Four-channel analog HD input terminal for Pt100 (RTD) for 3-wire connection

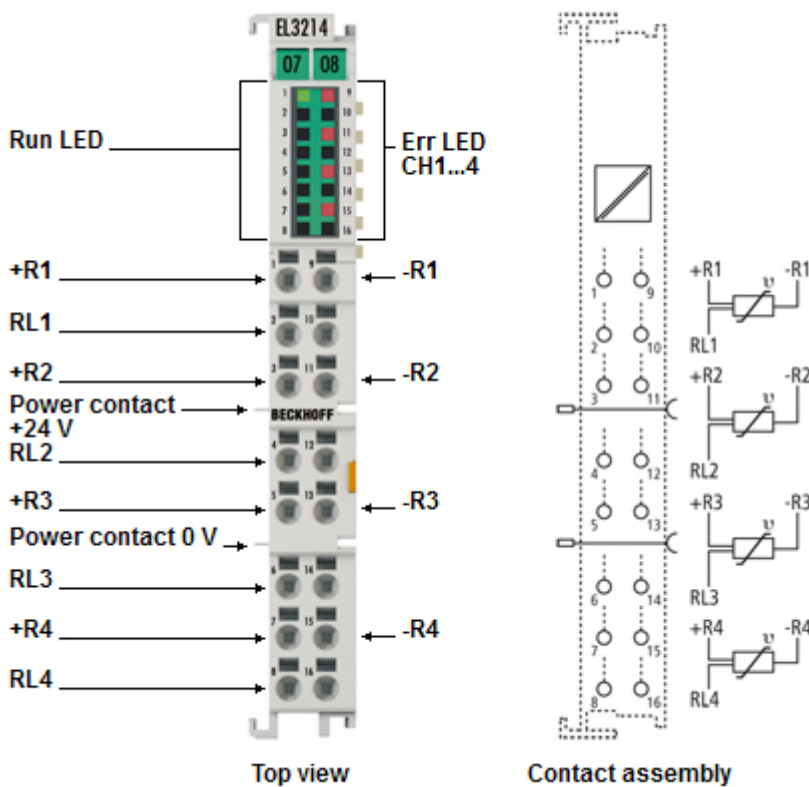


Fig. 12: EL3214

The EL3214 analog input terminals allows four resistance sensors to be connected directly on a width of 12 mm.

The EtherCAT Terminal circuitry can operate 3-wire sensors. A microprocessor handles linearization across the whole temperature range, which is freely selectable.

The EtherCAT Terminals standard settings are: resolution $0.1 \text{ }^\circ\text{C}$. Sensor malfunctions such as broken wires are indicated by error LEDs.

Eight-channel analog HD input terminal for Pt100 (RTD)

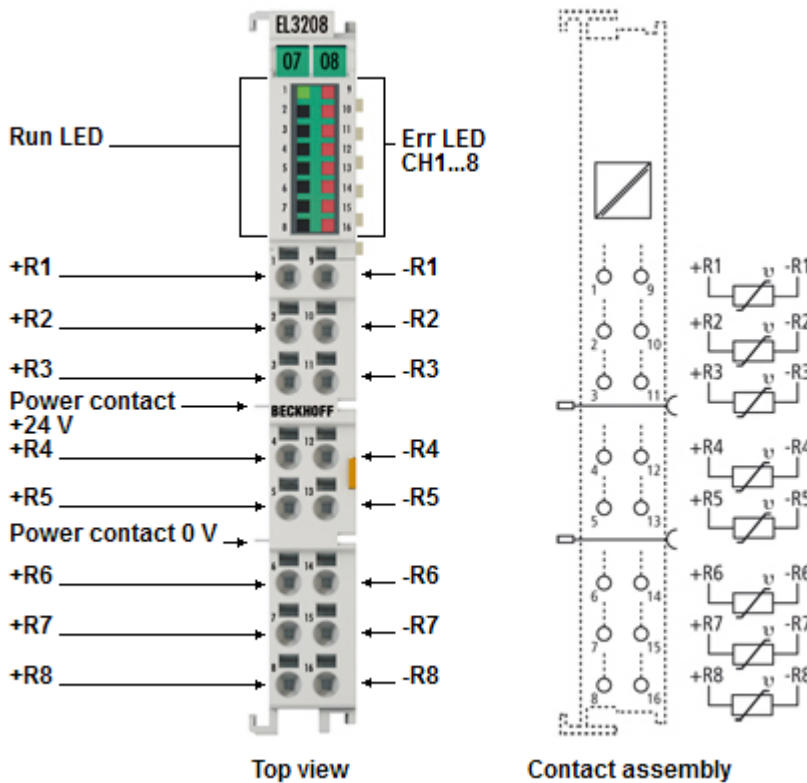


Fig. 13: EL3208

The EL3208 analog input terminals allow eight resistance sensors to be connected directly on a width of 12 mm (High Density housing).

The EtherCAT Terminal circuitry can operate 2-wire sensors. A microprocessor handles linearization across the whole temperature range, which is freely selectable.

The EtherCAT Terminals standard settings are: resolution 0.1 °C. Sensor malfunctions such as broken wires are indicated by error LEDs.

Quick links

- [EtherCAT function principles](#)
- [EL32xx technology \[► 19\]](#)
- [TwinCAT configuration creation \[► 72\]](#)
- [CoE object description and parameterization \[► 158\]](#)
- [Status word \[► 176\]](#)
- [Process data and operating modes \[► 149\]](#)
- [EL32xx-00xx settings and application notes \[► 138\]](#)
- [EL3204-0200 settings and application notes \[► 141\]](#)

2.2 EL3214-0090 - Introduction

4-channel input terminal PT100 (RTD), 2/3-wire connection, TwinSAFE Single Channel

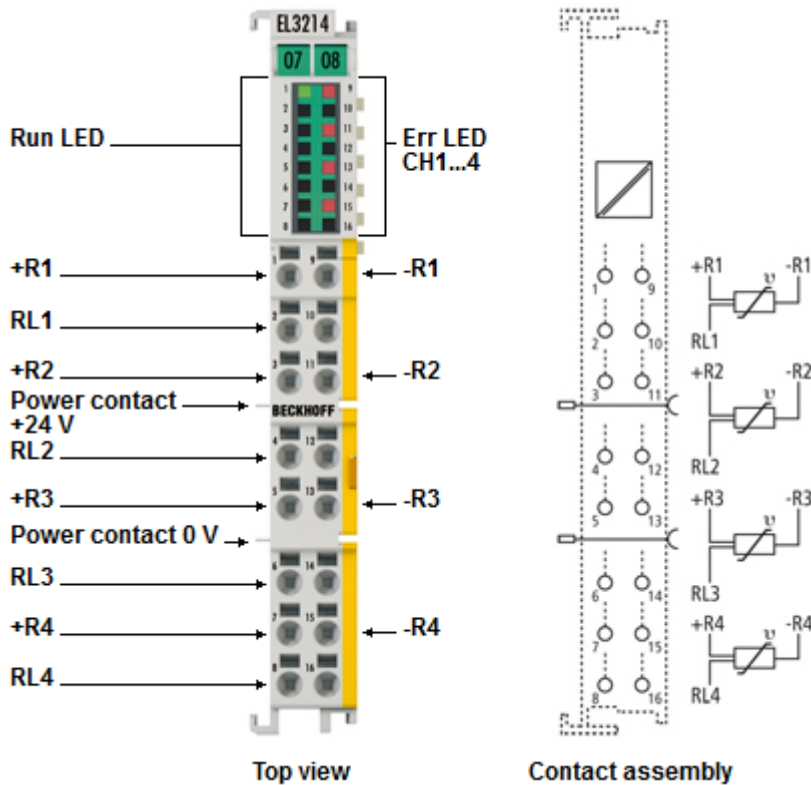


Fig. 14: EL3214-0090

The EL3214-0090 analog input terminal allows four resistance sensors to be connected directly. The EtherCAT Terminal circuit can operate sensors using the 3-wire technique (from firmware 02 also in 2-wire technology). A microprocessor handles linearization across the whole temperature range, which is freely selectable.

With the aid of the TwinSAFE SC technology (TwinSAFE Single Channel) it is possible to make use of standard signals for safety tasks in any network or fieldbus. The standard functions and features of the I/Os remain available. The data from these TwinSAFE SC I/Os is fed to the TwinSAFE Logic, where they undergo safety-related multi-channel processing. In the Safety Logic the data originating from different sources is analyzed, checked for plausibility and submitted to a "voting". This is done by certified function blocks such as Scale, Compare/Voting (1oo2, 2oo3, 3oo5), Limit, etc. For safety reasons, however, at least one of the data sources must be a TwinSAFE SC component. The remainder of the data can originate from other standard Bus Terminals, drive controllers or measuring transducers.

With the aid of the TwinSAFE SC technology it is typically possible to achieve a safety level equivalent to PL d/Cat. 3 in accordance with EN ISO 13849-1 or SIL 2 in accordance with EN 62061.

Quick links

- [EtherCAT function principles](#)
- [EL32xx technology \[► 19\]](#)
- [TwinCAT configuration creation \[► 72\]](#)
- [CoE object description and parameterization \[► 158\]](#)
- [Status word \[► 176\]](#)
- [Process data and operating modes \[► 149\]](#)

2.3 Technology RTD measuring

Table of contents

- [Function](#) [[▶ 19](#)]
- [Notes regarding EL320x-0010/-0020 series \(high-precision\)](#) [[▶ 20](#)]
- [Ratiometric voltage measurement](#) [[▶ 20](#)]
- [Connection techniques](#) [[▶ 20](#)]
- [Example of line compensation in the 2-wire mode](#) [[▶ 21](#)]
- [Overview of suitable resistance sensors](#) [[▶ 21](#)]

Function

The EL32xx-xxxx analog input terminals allow resistance sensors in the range 0 - 4096 Ohm (EL3204-0200: up to 240 kOhm, see special notes) to be connected directly.

Functions:

- Resistance measurement
 - Measuring range 0 to 1024 Ω : Resolution 1/64 Ohm
 - Measuring range 0 to 4096 Ω : Resolution 1/16 Ohm
 - The use of the terminal in the range from 0 to 10 Ohm is not recommended due to the relatively low measuring accuracy. Resistance measurement terminals such as the [EL3692](#) with 4-wire measurement are to be used for such measurements.
 - With the EL3201 and EL3202 the external bridge must be inserted between +R and +RL in 3-wire mode
- Temperature measurement

The measured sensor resistance is converted directly into a temperature by the internal μC via the desired linearization characteristic curve

 - Standard resolution 1/10 $^{\circ}\text{C}$ (1 digit = 0.1 $^{\circ}\text{C}$) according to a theoretically representable temperature range [- 3276.7 to 3276.8 $^{\circ}\text{C}$]
The physically specified temperature range for the respective sensor is to be observed!
 - Various PTC sensor characteristic curves are implemented over their complete measuring range for selection in the EL32xx series: Pt/Ni xxxx, KTY xx
 - Scaling and presentation can be changed
 - If a sensor is to be used that is not already implemented in the EL320x, the EL3204-0200 can be used
Due to its free programmability, the EL3204-0200 can be adjusted to any sensor characteristic curve in the permissible resistance range.

Additional notes:

- The resistance is determined by means of ratiometric voltage measurement, [see below](#) [[▶ 20](#)].
- The error state 'broken wire' is detected as overrange, signaled as an error to the controller and indicated by the ERROR LED.
- The error state 'short-circuit' is detected as underrange if the resistance is smaller than the smallest resistance of the measuring range, signaled as an error to the controller and indicated by the ERROR LED.
- Characteristic curves for KT/KTY sensors are implemented from firmware 06 (EL3201, EL3202-xxxx, EL3204) and selectable via the CoE directory.
- In the delivery state, the measured value is displayed in increments of 1/10 $^{\circ}\text{C}$ in two's complement format (integer).
- Other methods of display, e.g. high resolution with 1/100 $^{\circ}\text{C}$ can be selected via CoE [0x80n0:02](#) [[▶ 159](#)].
When using the high resolution a temperature range of -320 to +320 $^{\circ}\text{C}$ (-32566 to 32567) is measurable by the 2-byte PDO.

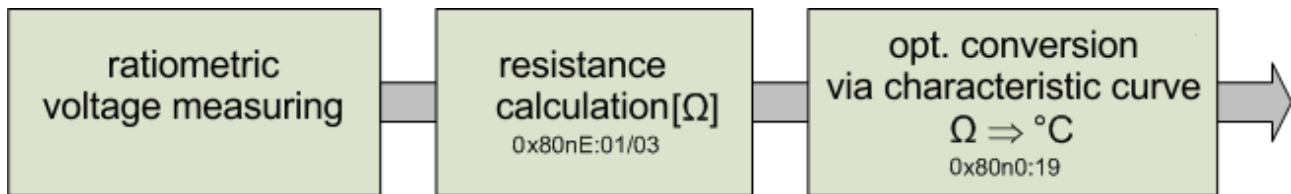


Fig. 15: Display of the measurement and calculation of the resistance/temperature

Notes regarding EL320x-0010/-0020 series (high-precision)

Thanks to its special hardware and software properties the EL320x-0010/0020 is able to measure the resistance value and thus the temperature with high accuracy. Please note:

● Increased accuracy EL320x-00x0

i In the case of the high-precision terminals the settings in the object 0x80n0 [▶ 158] / RTD-Settings must not be changed! The high-precision resistance/temperature measurement can only be performed in the range 10 to 220 Ohm, corresponding to a Pt100 measuring range of -200 to 320 °C.

In order to be able to display the high accuracy, the 0.01 °C/digit display is preset. The increased accuracy of the EL320x-00x0 is therefore valid only for the following settings:

- Resistance measurement or Pt100 characteristic curve (0.00385 Ω/Ω/°C, IEC60751 characteristic curve Pt385) – other characteristic curves are not guaranteed!
- Ambient temperature of 40 °C
- 4-wire connection technique
- Measuring range -200 to 320 °C
- 50 Hz filter

Temperatures or resistances outside of the range specified above cannot be measured with high precision. Outside these benchmark figures the high-precision terminal behaves like a terminal with standard accuracy.

Ratiometric voltage measurement

The EL32xx-xxxx measures resistance ratiometrically by means of voltage comparison, see fig. “4-wire and 3-wire connection technique”:

- A constant voltage of 2.5 V is applied across a highly precise known reference resistance R_{ref} and the sensor R_t connected in series.
 - EL3204-0200: 120 kOhm reference resistor
 - all other EL32xx: 5 kOhm reference resistor
- The sensor resistance can be determined by comparing the two voltages, V1 at reference and V2 at the sensor
- The measuring current through the sensor thus depends on the sensor resistance; this must be considered in questions of sensor self-heating
Example: at 0°C and thus 1000 Ohm internal resistance, a Pt1000 thus causes a measuring current of 0.1 mA on an EL3204.

● Wiring of the input channels

i Due to this measurement principle (resistive temperature sensor), a sensor may not be connected in parallel to two or more input channels!

Connection techniques

The electrical connection of a resistance sensor to the EL32xx can take place using the 2-wire, 3-wire or 4-wire (EL3201 only) methods. Since the measuring method is a resistance measurement, the sensor supply cables with their internal resistance can falsify the measurement. The following are available for this:

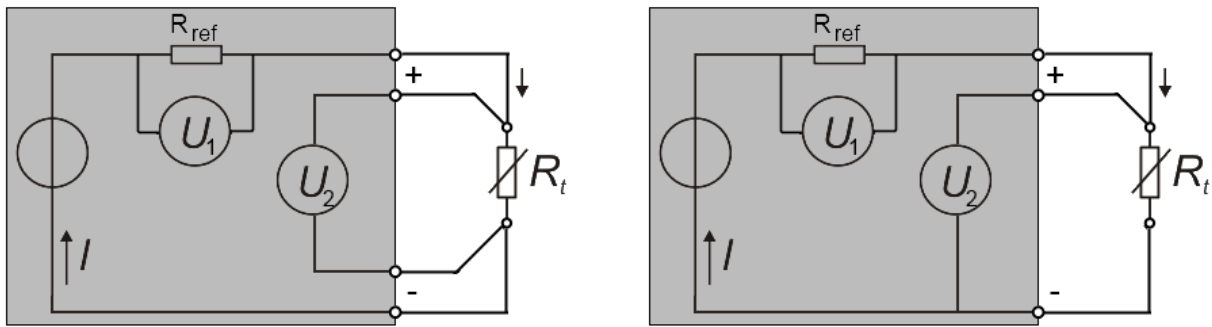


Fig. 16: 4-wire and 3-wire connection technology

- 4-wire sensors: current conduction and voltage measurement take place via different cores. Since the voltage is measured by the EL32xx with high impedance, the cores for the voltage measurement are subject to virtually no voltage drop.
- 3-wire sensors: this simplified connection reduces wiring costs and compensates cable resistances to a considerable degree.
- 2-wire sensors: very simple connection, recommended only for short supply cables
The supply line resistances can be eliminated from the calculation in 2-wire mode if they are made known to the EL320x in the CoE object [0x80n0:1B](#) [[▶ 159](#)] (unit [1/32 Ω]). The determination of the supply line resistance is possible on the application side either by measurement or by comparison.

● Two-wire connection



If the EL3201/EL3202 is operated with a two-wire connection, the inputs +R and +RL must be bridged by the user.

Example of line compensation in the 2-wire mode

Example of line compensation in the 2-wire mode. The cross-sectional area of the 50 meter long copper wire connecting cable is 0.5 mm². The specific resistance of copper is 0.0175 Ω mm² m⁻¹.

Determination of the total resistance of the feed cable:

$$R_{\text{tot}} = 0.0175 \Omega \text{ mm}^2 \text{ m}^{-1} * (2 * 50 \text{ m} / 0.5 \text{ mm}^2) = 3.5 \Omega$$

For a resistance change of the Pt100 of ≈0.39 Ω/K, the resulting temperature deviation is:

$$3.5 \Omega / (0.39 \Omega/^{\circ}\text{C}) = 8.97 \text{ }^{\circ}\text{C}$$

if the line resistance is not taken into account. If the 3.5 Ohm are now entered as

$$3.5 \Omega / (1/32) \Omega = 112_{\text{dec}}$$

in [0x8000:1B](#) [[▶ 159](#)] this is subtracted from the measured value and the temperature is corrected accordingly.

● Plug variant ES32xx with Pt100 / Ni100 sensors



When using Pt100 / Ni100 sensors we recommend the use of the EL32xx in order to avoid losses of accuracy due to transition resistances with the plug variant ES32xx.

Overview of suitable resistance sensors

Overview of suitable resistance sensors. The following resistance sensors are suitable for temperature measurement with the EL32xx-0000 (table 1) and can be selected via the object [0x80n0:19](#) [[▶ 159](#)].

Special or deviating characteristics (e.g. Pt375) can be measured with the freely parameterizable EL3204-0200.

Type	Resistance range	Implemented temperature range
Pt100 (0.00385 $\Omega/\Omega/^\circ\text{C}$, IEC60751 characteristic curve Pt385)	~ 18 ... ~390 Ohm	-200 °C to 850 °C (for high-precision terminals see Note [► 20] !)
Ni100		-60°C to 250°C
Pt1000 (0.00385 $\Omega/\Omega/^\circ\text{C}$, IEC60751 characteristic curve Pt385)	~180 ... ~3900 Ohm	-200°C to 850°C
Pt500		-200°C to 850°C
Pt200		-200°C to 1370°C
Ni1000		-60°C to 250°C
Ni1000 TK5000 100°C: 1500 ohm		-30 to 160°C
Ni120		-60°C to 320°C
KT100/110/130/210/230 KTY10/11/13/16/19 KTY81/82-110,120,150 KTY81-121 KTY81-122 KTY81-151 KTY81-152 KTY81/82-210,220,250 KTY81-221 KTY81-222 KTY81-251 KTY81-252	~500 ... ~2200 Ohm	-55 ...150°C
KTY83-110,120,150 KTY83-121 KTY83-122 KTY83-151 KTY83-152	~500 ... ~2500 Ohm	-50...175°C
KTY84-130,150 KTY84-151	~350 ... ~2500 Ohm	-40...300°C
KTY21/23-6 KTY1x-5 KTY1x-7 KTY21/23-5 KTY21/23-7	~500 ... ~4000 Ohm	-50...150°C

2.4 Basic principles of RTD technology

Certain materials change their electrical resistance if the temperature of the material changes. Thanks to this property they can be used as sensors for the measurement of temperature. Such an RTD element (Resistance Temperature Device) or thermistor then exhibits a well-known material-dependent property, i.e. how the resistance changes in relation to the temperature – the so-called characteristic. In an initial approximation this characteristic can be taken to be a linear equation:

$$\Delta R = k\Delta T$$

The factor k can be positive or negative and must be specified by the sensor manufacturer:

- positive coefficient (PTC): resistance increases with increasing temperature, i.e. it conducts less well; the sensor is then referred to as a PTC thermistor
- negative coefficient (NTC): resistance increases with decreasing temperature, i.e. it conducts better; the sensor is then referred to as an NTC thermistor

The larger the coefficient, the higher the sensitivity of the sensor.

● Temperature measurement

i This kind of temperature measurement is to be distinguished from that using thermocouple sensors: these spontaneously generate a (small) voltage across the conductor, which is measured at the contact points.

Within a very small measuring range nearly all materials can be described by such a linear characteristic. However, it is often necessary to measure over a large measuring range, e.g. several tens or hundreds of °K. Within such ranges the characteristic must be described for many materials by non-linear equations of a higher order or by using exponential components. Examples of such equations are

- Platinum/PT sensors (PTC thermistor) according to IEC 60751:
 - for the range -200.. 0 °C:

$$R(T) = R_0(1 + AT + BT^2 + C(T - 100^\circ\text{C})T^3)$$

- for the range 0 °C..850 °C:

$$R(T) = R_0(1 + AT + BT^2)$$

The coefficients A , B and C are to be specified by the sensor manufacturer or taken from the standard. The parameter R_0 indicates the resistance in ohm of the platinum sensor at $T = 0$ °C. The sensor designations are based on these characteristics, e.g. for PT100 $R_0 = 100 \Omega$ at $T = 0$ °C.

- Steinhart-Hart (for NTC thermistor)

$$\frac{1}{T} = a + b \ln(R) + c \ln^3(R)$$

The coefficients a , b , c should be specified by the sensor manufacturer, or they can be determined by measuring the resistance at three known temperatures.

- B-parameter equation (for NTC thermistor)

$$R_T = R(T) = A \cdot e^{\frac{B}{T}} = R_{T_0} \cdot e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)}$$

The coefficients R_{T_0} , B , T_0 should be specified by the sensor manufacturer, or they can be determined by measuring the resistance at two known temperatures.

The B-parameter equation is a simplified version of the Steinhart-Hart equation. The B-parameter itself is only constant in a small range, e.g. between 25°C.. 50°C or 25°C.. 85°C, which is identified as follows: $B_{25/50}$ or $B_{25/85}$. The accuracy of the equation strongly depends on the B-parameter. The larger the measuring range, the lower the accuracy. If a larger measuring range required, it is preferable to use the Steinhart-Hart equation.

- and others

A typical characteristic is shown for each of the NTC and PTC families in following figure:

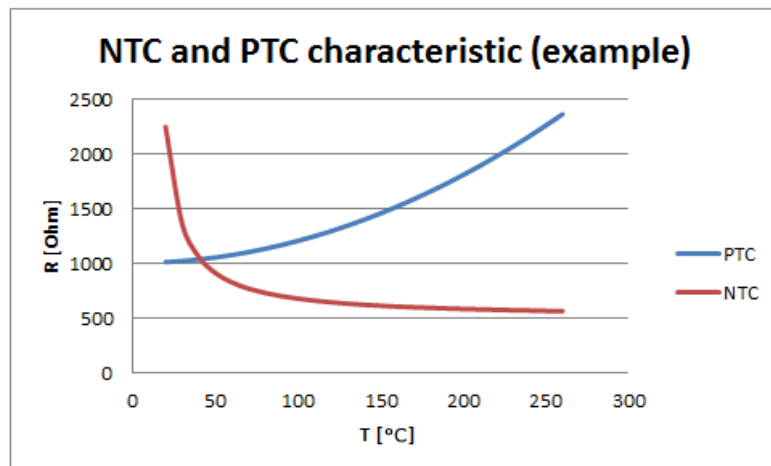


Fig. 17: Examples for temperature-dependent resistance values

Hence, there is no such *thing* as a general NTC or PTC sensor – in fact, they are names for sensor families with a certain behavior.

For common sensors/characteristics such as PT100, these characteristics are already permanently implemented into the measuring devices. The user must check whether a sensor that he intends to use is supported by the measuring device. The following criteria apply here:

- Temperature range: does the sensor support the intended temperature range?
- Measuring range: can the sensor resistance be measured in the intended temperature range?
- Characteristic curve: can the measured resistance be converted accordingly into temperature? (base point, gradient/coefficient)
- Velocity: how often is the resistance measured?

In a quite basic way, a sensor manufacturer can of course also publish the characteristic of its sensor as a value table.

i Resistance measurement

To determine the resistance it is usual to pass a measuring current in the mA range (< 5 mA) through the sensor and to measure the resulting voltage. Three effects must be taken into consideration when doing this:

- the measuring current can lead to self-heating of the sensor. However, this usually has only a minimal effect on the measuring accuracy.

Special sensors tend to be used for cryogenic applications.

- the sensor supply lines also always have a resistance and add a (usually) constant additional resistance to the measurement. This can be compensated by

- 3 or 4-wire connection of the sensor
- manually accounting for the known wire resistance in the calculation
- using a sensor with a higher nominal resistance – the supply line effects are then of less consequence
- Insulation faults or thermovoltages can affect the measurement.

For classification there follows an overview of the NTC/PTC properties of various sensors:

NTC and PTC properties

NTC	PTC
many semiconductors	many metals
various ceramics	various ceramics

NTC	PTC
NTC20, NTC100, etc.	Pt100, Pt1000, ..
	KTY ..
	Ni100, Ni1000, ..
	FeT

● Sensor exchange

i Please note that 1:1 exchangeability is not always guaranteed, especially in the case of manufacturer-specified sensors. If necessary the new sensor must be recalibrated in the system.

2.5 Technical data

EL3201 - Technical data

Technical data	EL3201	EL3201-0010	EL3201-0020
Number of inputs	1		1
Sensor types	Pt100, Pt200, Pt500, Pt1000, Ni100, Ni120, Ni1000 KT/KTY from firmware 06 [▶ 194] Resistance measurement 10 Ω to 1 kΩ or 10 Ω to 4 kΩ (e.g. for potentiometer connection)		
Connection method	2-, 3-, 4-wire	4 wire	
Temperature range	Range-dependent: -200...+850°C (Pt sensors); -60...+250°C (Ni sensors)	Range-dependent: -200...+850°C (Pt sensors); -60...+250°C (Ni sensors) High-precision measurement for Pt100 sensors only, Measuring range -200...+320°C, see note [▶ 20!]	
Resolution (default)	0.1°C per digit	0.01°C per digit	
Conversion time	approx. 4 - 500 ms (configurable), depending on configuration and filter setting approx. 24 ms, preset		
Measuring current (depending on the sensor element and temperature)	typ. < 0.5 mA		
Measuring error	< ±0.5 °C for Pt sensors < ±1.5 °C (extended temperature range)	< ±0.1 °C (0°C ... +55°C) < ±0.3 °C (extended temperature range) (for Pt100 sensors, ambient temperature of 40 °C, 4-wire connection technology, Measuring range -200 to 320°C), 50 Hz filter	< ±0.1 °C (for Pt100 sensors, ambient temperature of 40 °C, 4-wire connection technology, Measuring range -200 to 320°C), 50 Hz filter
Width in the process image	max. 4-byte input		
Power supply for electronics	via the E-Bus		
Current consumption from the E-bus	typ. 190 mA		
Electrical isolation	500 V (E-bus/field voltage)		
Configuration	via TwinCAT System Manager		
Weight	approx. 60 g		
Permissible ambient temperature range during operation	-25°C to +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 (connected width: 12 mm)		
Installation [▶ 46]	on 35 mm mounting rail according to EN 60715		
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [▶ 49] for terminals with increased 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 [▶ 58] cULus [▶ 59]		CE ATEX [▶ 57] cULus [▶ 59]

EL3202 - Technical data

Technical data	EL3202	EL3202-0010, EL3202-0020
Number of inputs	2	2
Sensor types	Pt100, Pt200, Pt500, Pt1000, Ni100, Ni120, Ni1000 KT/KTY from firmware 06 [▶ 194] Resistance measurement 10 Ω to 1 kΩ or 10 Ω to 4 kΩ (e.g. for potentiometer connection)	
Connection method	2-, 3-wire (Preset: 3-wire)	4-wire
Temperature range	Range-dependent: -200...+850°C (Pt sensors); -60...+250°C (Ni sensors)	Range-dependent: -200...+850°C (Pt sensors) -60...+250°C (Ni sensors) High-precision measurement for Pt100 sensors only, Measuring range -200...+320°C, see note [▶ 20]!
Resolution (default)	0.1°C per digit	0.01°C per digit
Conversion time	approx. 800 ms - 2 ms (configurable), depending on configuration and filter setting approx. 85 ms, preset	
Measuring current (depending on the sensor element and temperature)	typ. < 0.5 mA	
Measuring error	< ±0.5 °C for Pt sensors < ±1.5 °C (extended temperature range)	< ±0.1 °C (0°C ... +55°C) < ±0.3 °C (extended temperature range) (for Pt100 sensors, ambient temperature of 40 °C, 4-wire connection technology, Measuring range -200 to 320°C), 50 Hz filter
Width in the process image	max. 8-byte input	
Power supply for electronics	via the E-Bus	
Current consumption from the E-bus	typ. 190 mA	
Electrical isolation	500 V (E-bus/field voltage)	
Configuration	via TwinCAT System Manager	
Weight	approx. 60 g	
Permissible ambient temperature range during operation	-25°C to +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 (connected width: 12 mm)	
Installation [▶ 46]	on 35 mm mounting rail according to EN 60715	
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [▶ 49] for terminals with increased 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 [▶ 58] cULus [▶ 59]	

EL3204 - Technical data

Technical data	EL3204	EL3204-0200
Number of inputs	4	4
Sensor types	Pt100, Pt200, Pt500, Pt1000, Ni100, Ni120, Ni1000 KT/KTY from firmware 06 [▶ 194] Resistance measurement 10 Ω to 1 kΩ or 10 Ω to 4 kΩ (e.g. for potentiometer connection)	any RTD within the range 100 Ω to 240 kΩ
Connection method	2-wire	
Temperature range	Range-dependent: -200...+850°C (Pt sensors); -60 ... +250°C (Ni sensors)	Range-dependent
Resolution (default)	0.1°C per digit	default: 0.1 °C per digit for temperature measurement
Conversion time	approx. 800 ms - 2 ms (configurable), depending on configuration and filter setting approx. 85 ms, preset	approx. 4 - 500 ms (configurable), depending on configuration and filter setting approx. 24 ms, preset
Measuring current (depending on the sensor element and temperature)	typ. < 0.5 mA	typ. < 0.03 mA
Measuring error	< ±0.5 °C for Pt sensors < ±1.5 °C (extended temperature range)	< ±0.3% of the respective full scale value during resistance measurement
Width in the process image	max. 16-byte input	max. 16-byte input
Power supply for electronics	via the E-Bus	
Current consumption from the E-bus	typ. 190 mA	typ. 150 mA
Electrical isolation	500 V (E-bus/field voltage)	
Configuration	via TwinCAT System Manager	
Weight	approx. 60 g	
Permissible ambient temperature range during operation	-25°C to +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 (connected width: 12 mm)	
Installation [▶ 46]	on 35 mm mounting rail according to EN 60715	
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [▶ 49] for terminals with increased 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 [▶ 58] cULus [▶ 59]	CE ATEX [▶ 57] cULus [▶ 59]

EL3208 - Technical data

Technical data	EL3208
Number of inputs	8
Sensor types	Pt100, Pt200, Pt500, Pt1000, Ni100, Ni120, Ni1000 KT/KTY, resistance measurement 10 Ω to 1 kΩ or 10 Ω to 4 kΩ (e.g. for potentiometer connection)
Connection method	2-wire
Temperature range	Range-dependent: -200...+850°C (Pt sensors); -60...+250°C (Ni sensors)
Resolution (default)	0.1°C per digit
Conversion time	approx. 3 ms - 1600 ms (configurable), depending on configuration and filter setting
Measuring current (depending on the sensor element and temperature)	typ. < 0.5 mA
Measuring error	< ±0.5 °C for Pt sensors < ±1.5 °C (extended temperature range)
Width in the process image	max. 32-byte input
Power supply for electronics	via the E-Bus
Current consumption from the E-bus	typ. 140 mA
Electrical isolation	500 V (E-bus/field voltage)
Configuration	via TwinCAT System Manager
Weight	approx. 60 g
Permissible ambient temperature range during operation	-25°C to +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 (connected width: 12 mm)
Installation [► 46]	on 35 mm mounting rail according to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [► 49] for terminals with increased mechanical load capacity
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE cULus [► 59]

EL3214 - Technical data

Technical data	EL3214
Number of inputs	4
Sensor types	Pt100, Pt200, Pt500, Pt1000, Ni100, Ni120, Ni1000 KT/KTY, resistance measurement 10 Ω to 1 kΩ or 10 Ω to 4 kΩ (e.g. for potentiometer connection)
Connection method	3-wire
Temperature range	Range-dependent: -200...+850°C (Pt sensors); -60...+250°C (Ni sensors)
Resolution (default)	0.1°C per digit
Conversion time	approx. 5 ms - 1600 ms (configurable), depending on configuration and filter setting
Measuring current (depending on the sensor element and temperature)	typ. < 0.5 mA
Measuring error	< ±0.5 °C with Pt sensors, 4 x 3-wire connection < ±1.5 °C (extended temperature range)
Width in the process image	max. 16-byte input
Power supply for electronics	via the E-Bus
Current consumption from the E-bus	typ. 140 mA
Electrical isolation	500 V (E-bus/field voltage)
Configuration	via TwinCAT System Manager
Weight	approx. 60 g
Permissible ambient temperature range during operation	-25°C to +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 (connected width: 12 mm)
Installation [► 46]	on 35 mm mounting rail according to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27, see also installation instructions [► 49] for terminals with increased mechanical load capacity
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE cULus [► 59]

EL3214-0090 - Technical data

Technical data	EL3214-0090
Number of inputs	4
Sensor types	Pt100, Pt200, Pt500, Pt1000, Ni100, Ni120, Ni1000 KT/KTY, resistance measurement 10 Ω to 1 kΩ or 10 Ω to 4 kΩ (e.g. for potentiometer connection)
Connection method	2-wire (from firmware 02), 3-wire
Temperature range	Range-dependent: -200...+850°C (Pt sensors); -60...+250°C (Ni sensors)
Resolution (default)	0.1°C per digit
Conversion time	approx. 170 ms (preset)
Measuring current (depending on the sensor element and temperature)	typ. < 0.5 mA
Measuring error	< ±0.5 °C with Pt sensors, 4 x 3-wire connection < ±1.5 °C (extended temperature range)
MTBF (55°C)	>890.000 h
Power supply for electronics	via the E-Bus
Current consumption from the E-bus	typ. 140 mA
Electrical isolation	500 V (E-bus/field voltage)
Configuration	via TwinCAT System Manager
Weight	approx. 60 g
Permissible ambient temperature range during operation	-25°C to +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 (connected width: 12 mm)
Installation [▶ 46]	on 35 mm mounting rail according to EN 60715
Vibration / shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Approval	CE cULus [▶ 59]

2.6 Notes to the factory calibration certificate

Basically every Beckhoff analogue device (input or output) will be justified i.e. will be calibrated during production. This procedure won't be documented unique but will be carried out in the scope of the calibration certificate by the respective "-0020" types. Whether a calibration certificate is existing/ deliverable can be requested by Beckhoff sales or be gathered by the current price list ("-0020" version exists e.g. for EL3202 means its type is: EL3202-0020).

The factory calibration certificate (German: "Werkskalibrierschein") entitles the residual error after compensation/adjustment to the used Beckhoff factory standard. The factory calibration certificate (pdf formatted) can be explicit allocated by the ID number of the terminal. It is ready to be downloaded by the Beckhoff website.

By its nature this certificate documents the accuracy of measurement at the time of certificate creation, it contains no assertion about the behavior or change of the measurement accuracy in the future. A calibration certificate acts as a backtracking view to the previous time of usage. By reiterated certification procedures over years (without justification) it allows making conclusions about its ageing behavior, so called calibrate history.

A device having a calibration certificate can be used within an application if the master conditions/ framework requirements approve that. If a national certificate (German: DAkkS, Deutsche Akkreditierungsstelle GmbH) will be requested, corresponding service providers can execute a DAkkS calibration on every Beckhoff product within their own laboratories. *Beckhoff Automation GmbH & Co. KG is not able to execute a DAkkS calibration at the moment (2016).*

i **Linguistic note**

In American English, "calibration" or "alignment" is understood to mean compensation/adjustment, thus a modifying effect on the device. "Verification", on the other hand, refers to observational determination and documentation of the residual error, referred in German language use as "*Kalibrierung*".

2.7 Start

For commissioning:

- Install the EL32xx as described in section [Installation \[▶ 44\]](#).
- Configure the EL32xx in TwinCAT as described in section [Commissioning \[▶ 72\]](#).

3 Basics communication

3.1 EtherCAT basics

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

3.2 EtherCAT cabling – wire-bound

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

Cables and connectors

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

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

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

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

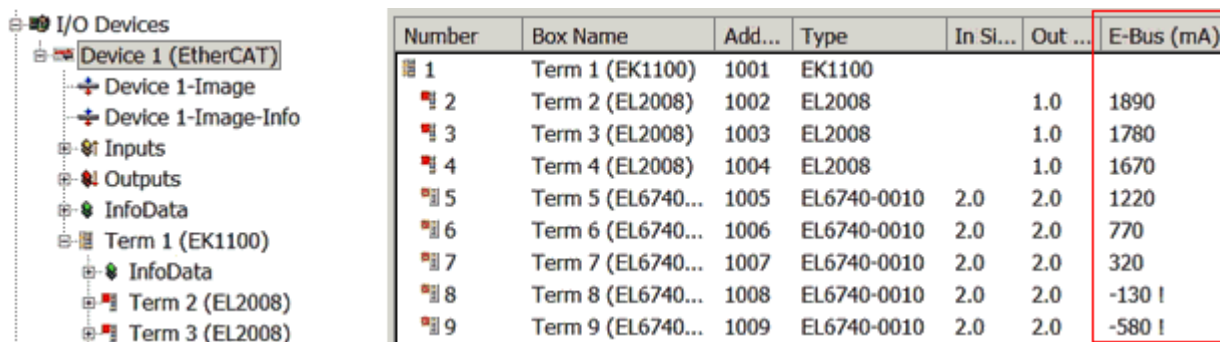
Recommended cables

Suitable cables for the connection of EtherCAT devices can be found on the [Beckhoff website!](#)

E-Bus supply

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

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



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

Fig. 18: System manager current calculation

NOTE**Malfunction possible!**

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

3.3 General notes for setting the watchdog

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

The EtherCAT slave controller (ESC) in the EL2xxx terminals features 2 watchdogs:

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

SM watchdog (SyncManager Watchdog)

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

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

PDI watchdog (Process Data Watchdog)

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

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

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

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

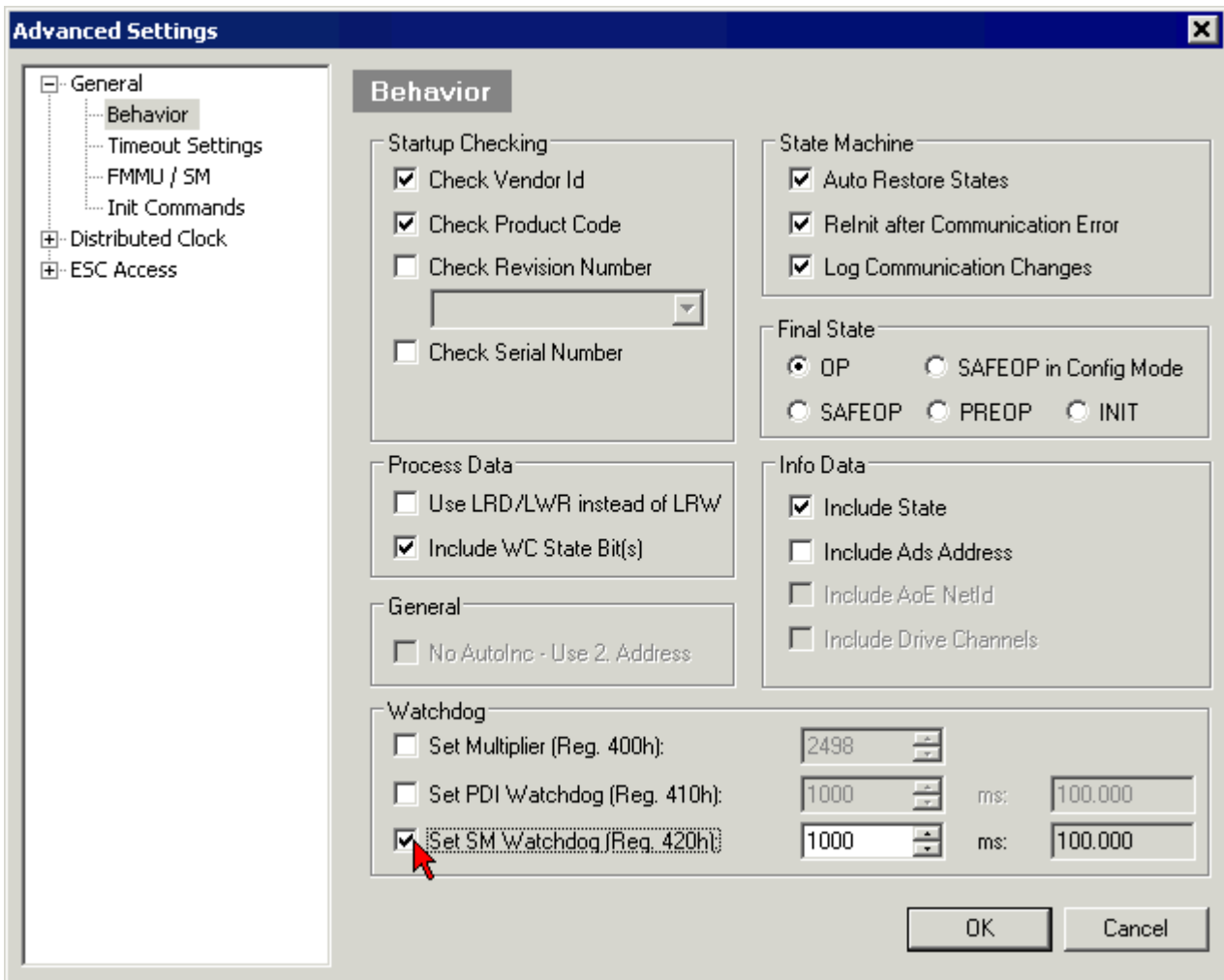


Fig. 19: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

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.

3.4 EtherCAT State Machine

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

A distinction is made between the following states:

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

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

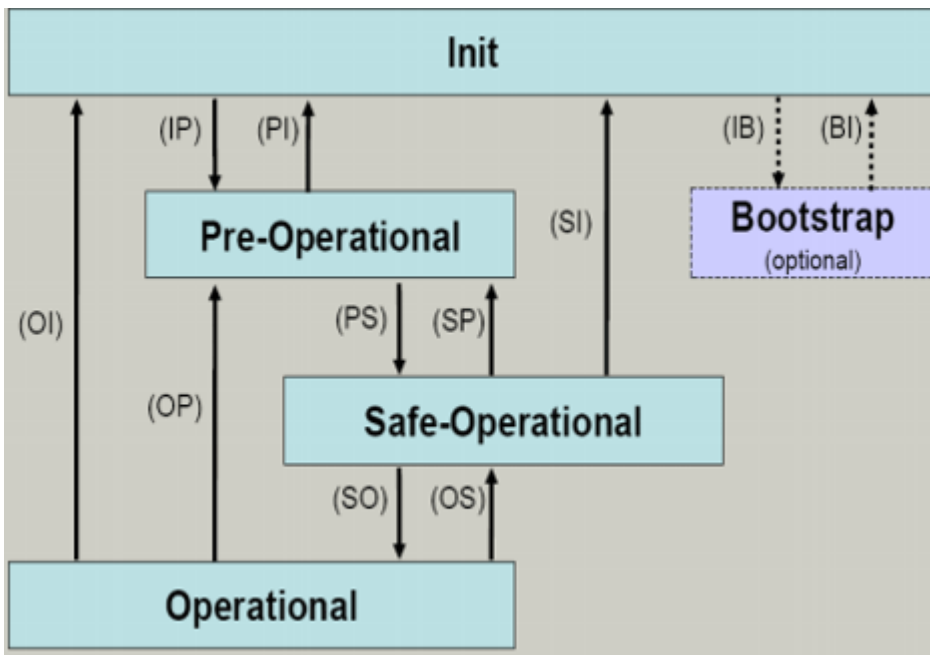


Fig. 20: States of the EtherCAT State Machine

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

Operational (Op)

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

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

Boot

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

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

3.5 CoE Interface

General description

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

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

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

The order is specified in 2 levels via hexadecimal numbering: (main)index, followed by subindex. The value ranges are

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

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

The relevant ranges for EtherCAT fieldbus users are:

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

Other important ranges are:

- 0x4000: In some EtherCAT devices the channel parameters are stored here (as an alternative to the 0x8000 range).
- 0x6000: Input PDOs ("input" from the perspective of the EtherCAT master)
- 0x7000: Output PDOs ("output" from the perspective of the EtherCAT master)

● Availability



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

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

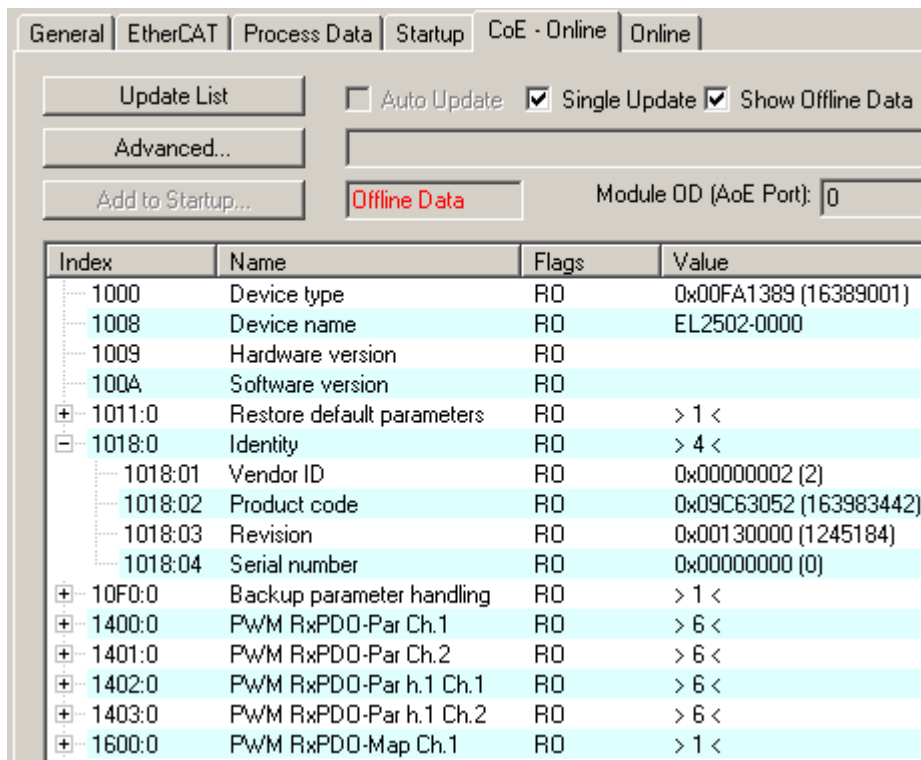


Fig. 21: "CoE Online " tab

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

Data management and function "NoCoeStorage"

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

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

Data management

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

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

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

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

Startup list

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

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

Recommended approach for manual modification of CoE parameters

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

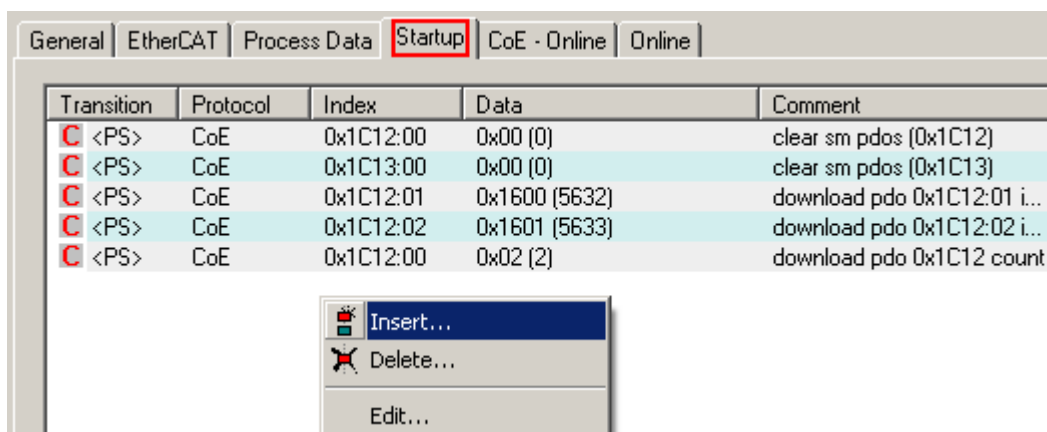


Fig. 22: Startup list in the TwinCAT System Manager

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

Online/offline list

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

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

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

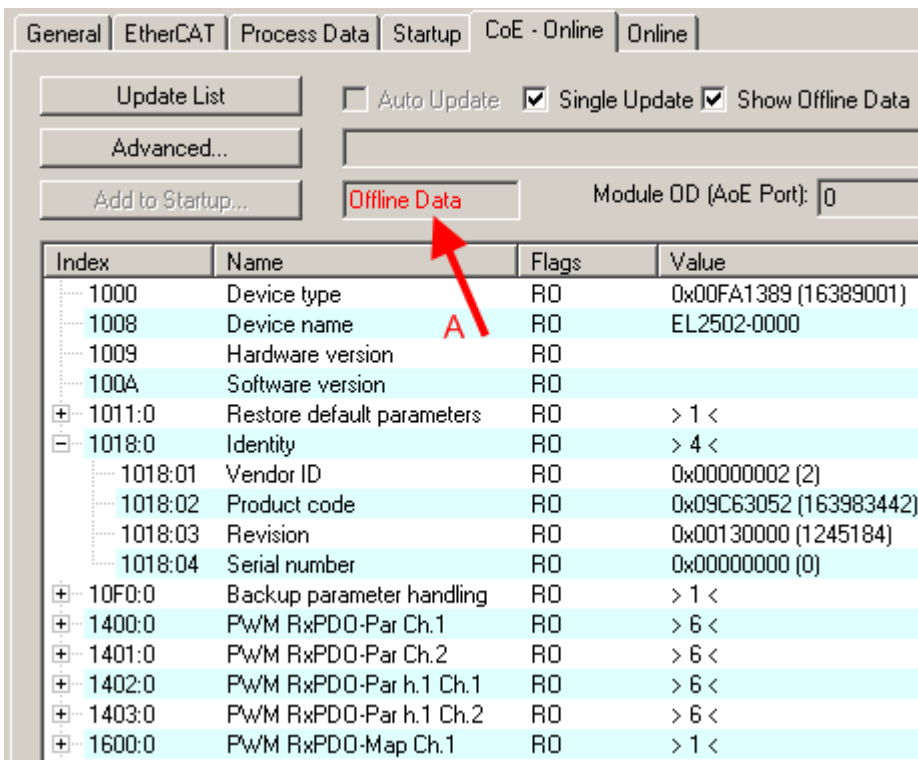


Fig. 23: Offline list

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

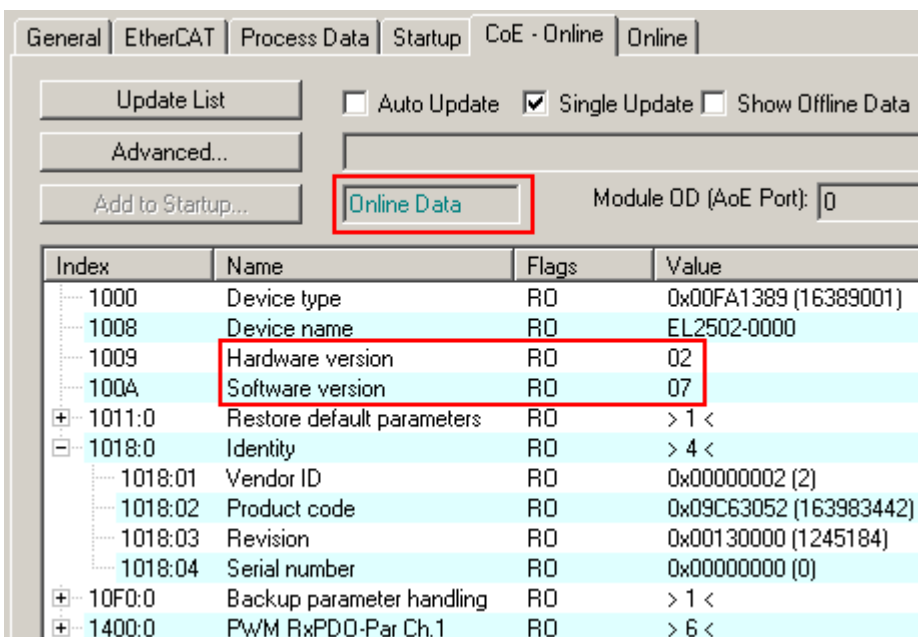


Fig. 24: Online list

Channel-based order

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

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

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

This is generally written as 0x80n0.

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

3.6 Distributed Clock

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

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

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

4 Installation

4.1 Safety instructions

Before installing and commissioning the TwinSAFE components please read the safety instructions in the foreword of this documentation.

4.2 Environmental conditions

Please ensure that the TwinSAFE components are only transported, stored and operated under the specified conditions (see technical data)!

WARNING

Risk of injury!

The TwinSAFE components must not be used under the following operating conditions.

- under the influence of ionizing radiation (that exceeds the level of the natural environmental radiation)
- in corrosive environments
- in an environment that leads to unacceptable soiling of the TwinSAFE component

NOTE

Electromagnetic compatibility

The TwinSAFE components comply with the current standards on electromagnetic compatibility with regard to spurious radiation and immunity to interference in particular.

However, in cases where devices such as mobile phones, radio equipment, transmitters or high-frequency systems that exceed the interference emissions limits specified in the standards are operated near TwinSAFE components, the function of the TwinSAFE components may be impaired.

4.3 Transport / storage

Use the original packaging in which the components were delivered for transporting and storing the TwinSAFE components.

CAUTION

Note the specified environmental conditions

Please ensure that the digital TwinSAFE components are only transported and stored under the specified environmental conditions (see technical data).

4.4 Control cabinet / terminal box

The TwinSAFE terminals must be installed in a control cabinet or terminal box with IP54 protection class according to IEC 60529 as a minimum.

4.5 Instructions for ESD protection

NOTE

Destruction of the devices by electrostatic discharge possible!

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

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

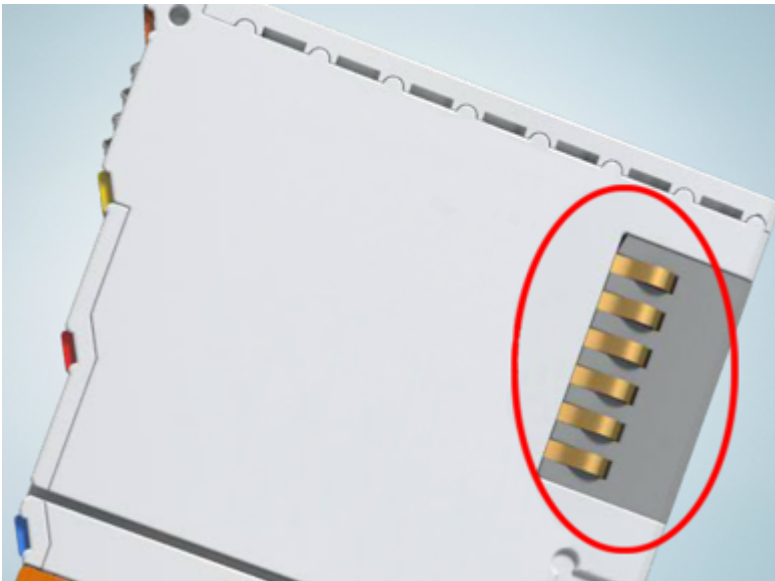


Fig. 25: Spring contacts of the Beckhoff I/O components

4.6 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

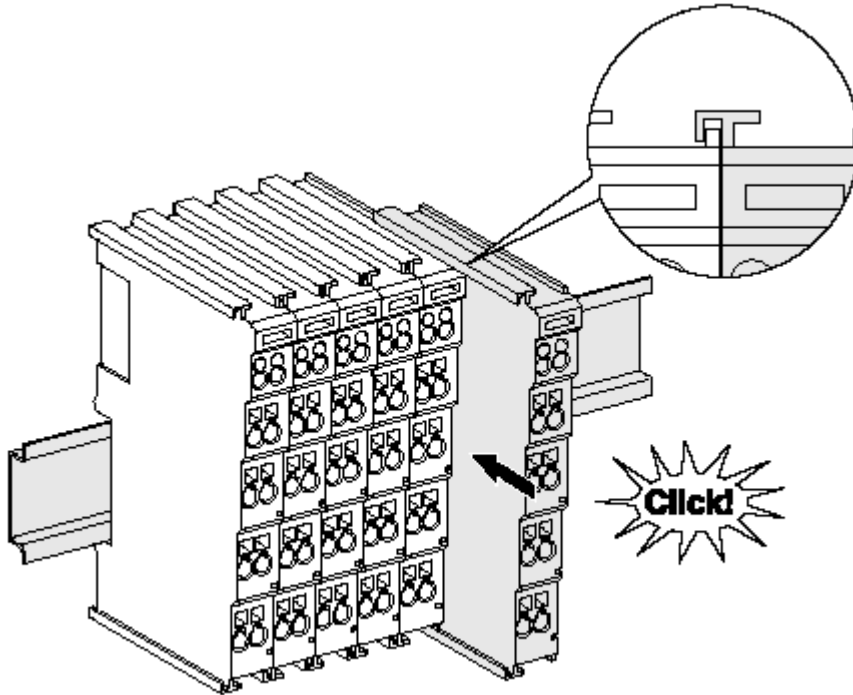


Fig. 26: Attaching on mounting rail

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.

i Fixing of mounting rails

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

Disassembly

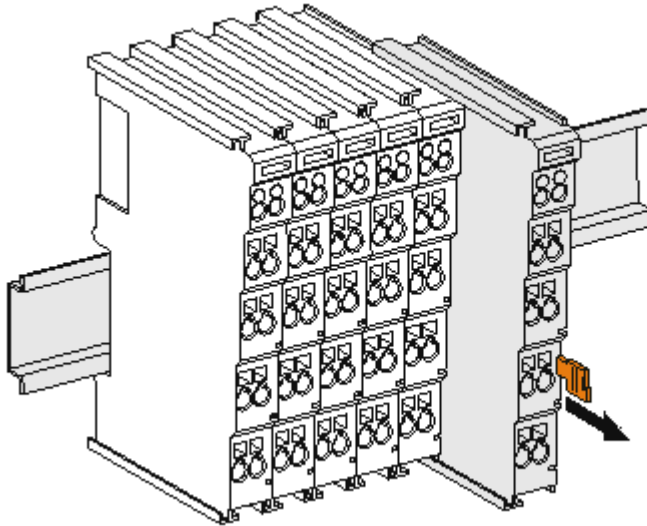


Fig. 27: Disassembling of terminal

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.

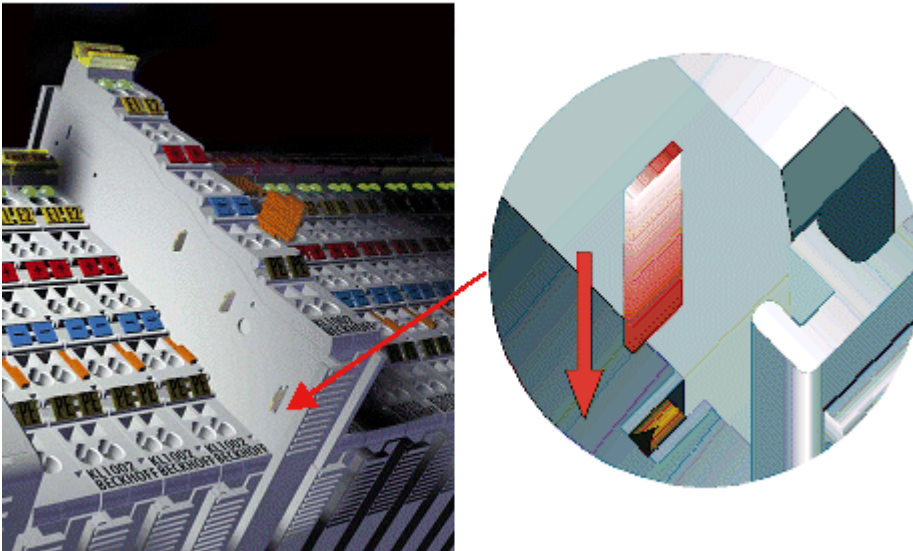


Fig. 28: Power contact on left side

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!

4.7 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.

4.8 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)

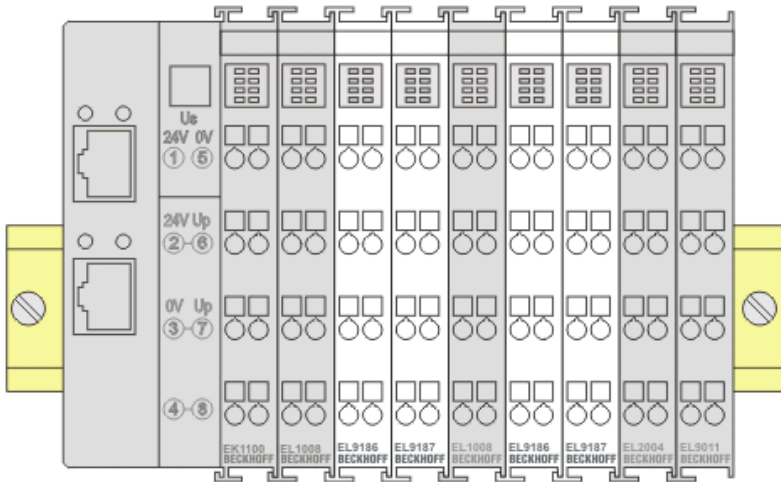


Fig. 29: Correct positioning

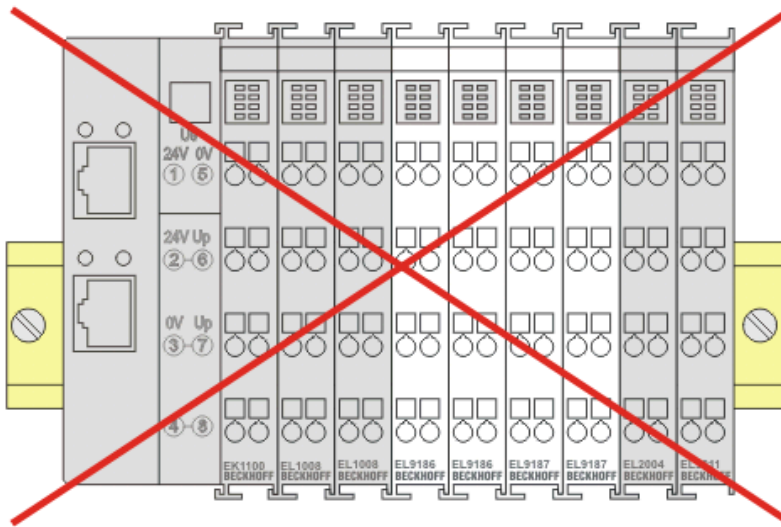


Fig. 30: Incorrect positioning

4.9 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.

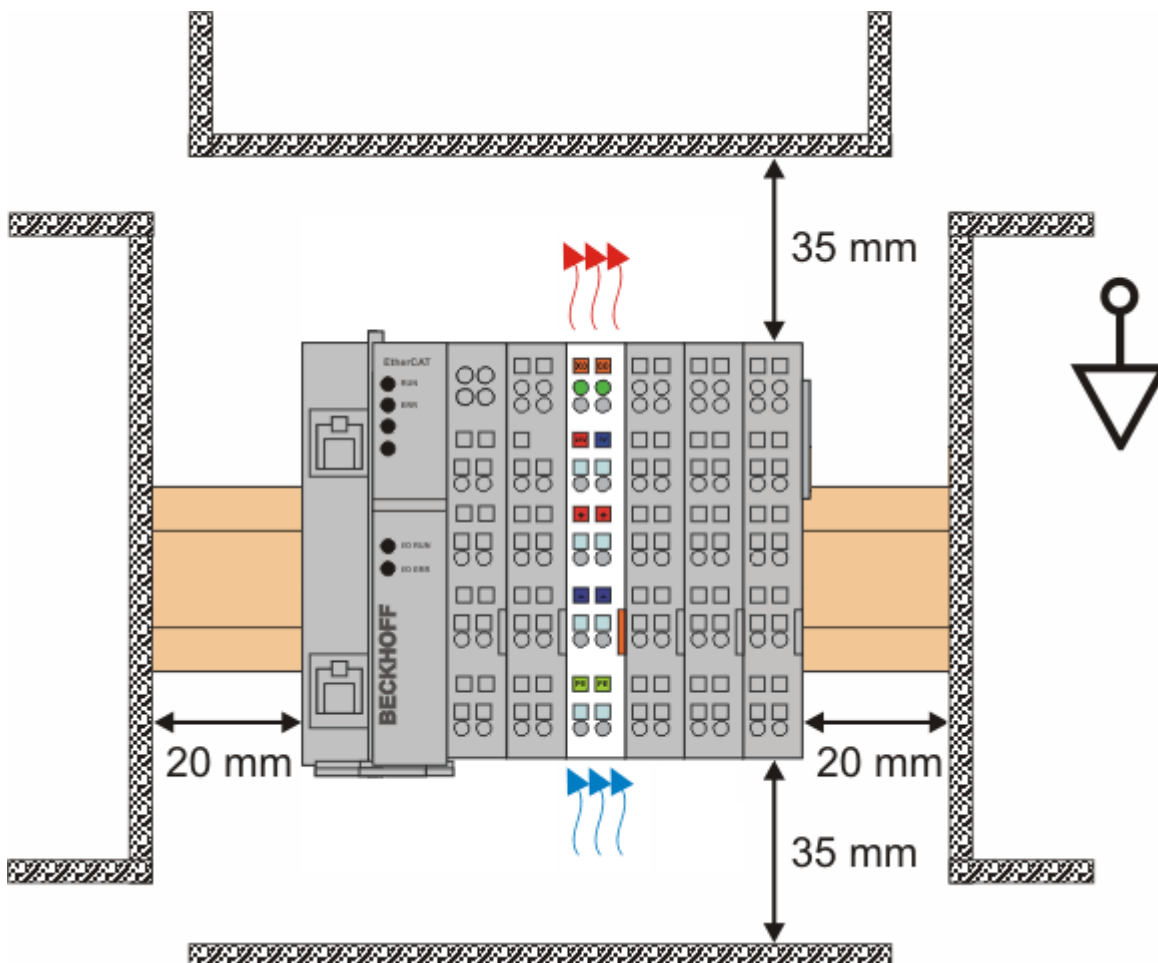


Fig. 31: Recommended distances for standard installation position

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.

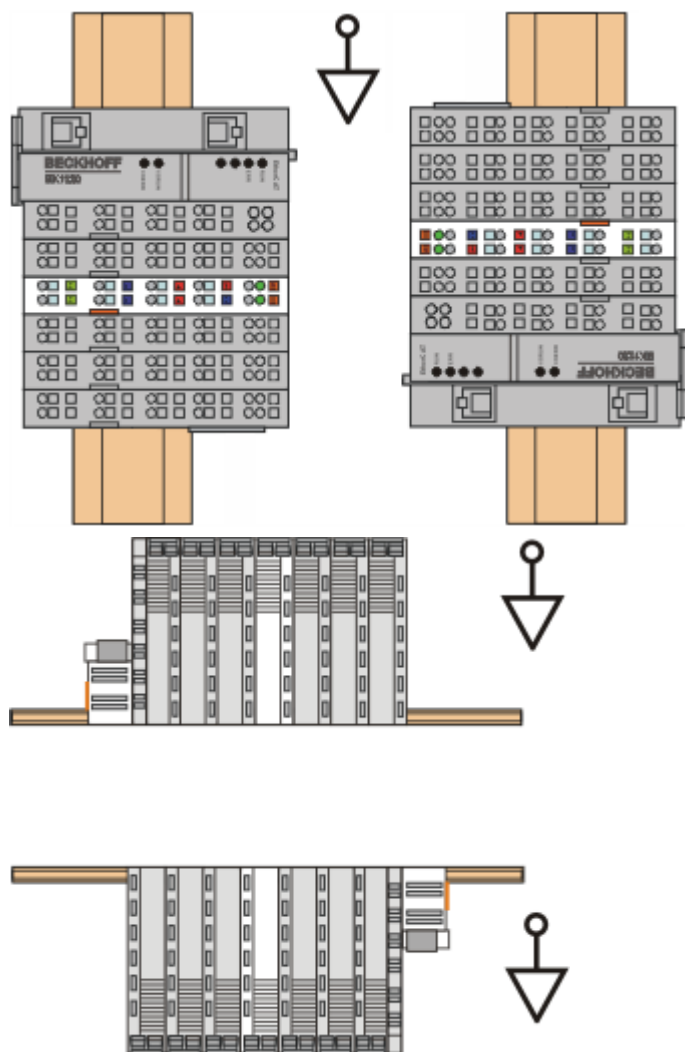


Fig. 32: Other installation positions

4.10 Connection

4.10.1 Connection system

⚠ WARNING

Risk of electric shock and damage of device!

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

Overview

The Bus Terminal system offers different connection options for optimum adaptation to the respective application:

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

Standard wiring (ELxxxx / KLxxxx)



Fig. 33: Standard wiring

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

Pluggable wiring (ESxxxx / KSxxxx)



Fig. 34: Pluggable wiring

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

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

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

Conductor cross sections between 0.08 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 ESxxxx and KSxxxx series has been retained as known from ELxxxx and KLxxxx series.

High Density Terminals (HD Terminals)



Fig. 35: High Density Terminals

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

● Wiring HD Terminals

i The High Density (HD) Terminals of the ELx8xx and KLx8xx series doesn't support pluggable 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 below!

4.10.2 Wiring

⚠ WARNING

Risk of electric shock and damage of device!
 Bring the bus terminal system into a safe, powered down state before starting installation, disassembly or wiring of the Bus Terminals!

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

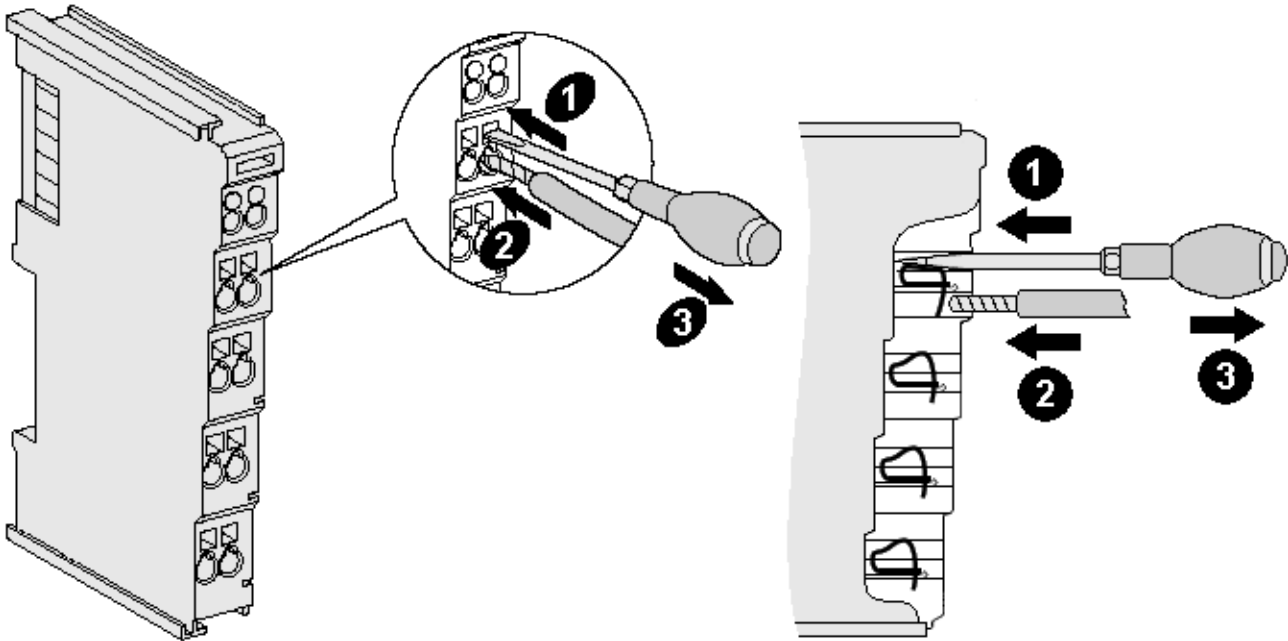


Fig. 36: Connecting a cable on a terminal point

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

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

See the following table for the suitable wire size width.

Terminal housing	ELxxxx, KLxxxx	ESxxxx, KSxxxx
Wire size width (single core wires)	0.08 ... 2.5 mm ²	0.08 ... 2.5 mm ²
Wire size width (fine-wire conductors)	0.08 ... 2.5 mm ²	0,08 ... 2.5 mm ²
Wire size width (conductors with a wire end sleeve)	0.14 ... 1.5 mm ²	0.14 ... 1.5 mm ²
Wire stripping length	8 ... 9 mm	9 ... 10 mm

High Density Terminals ([HD Terminals](#) [▶ 54]) with 16 terminal points

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

Terminal housing	High Density Housing
Wire size width (single core wires)	0.08 ... 1.5 mm ²
Wire size width (fine-wire conductors)	0.25 ... 1.5 mm ²
Wire size width (conductors with a wire end sleeve)	0.14 ... 0.75 mm ²
Wire size width (ultrasonically "bonded" conductors)	only 1.5 mm ²
Wire stripping length	8 ... 9 mm

4.10.3 Shielding



Shielding

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

4.11 ATEX - Special conditions (standard temperature range)

⚠ WARNING

Observe the special conditions for the intended use of Beckhoff fieldbus components with standard temperature range in potentially explosive areas (directive 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 0 to 55°C for the use of Beckhoff fieldbus components standard temperature range in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

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

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

Marking

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



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

or



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

4.12 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

4.13 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!>

4.14 UL notice

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

Basic principles

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



4.15 LEDs and connection

EL3201, EL3202

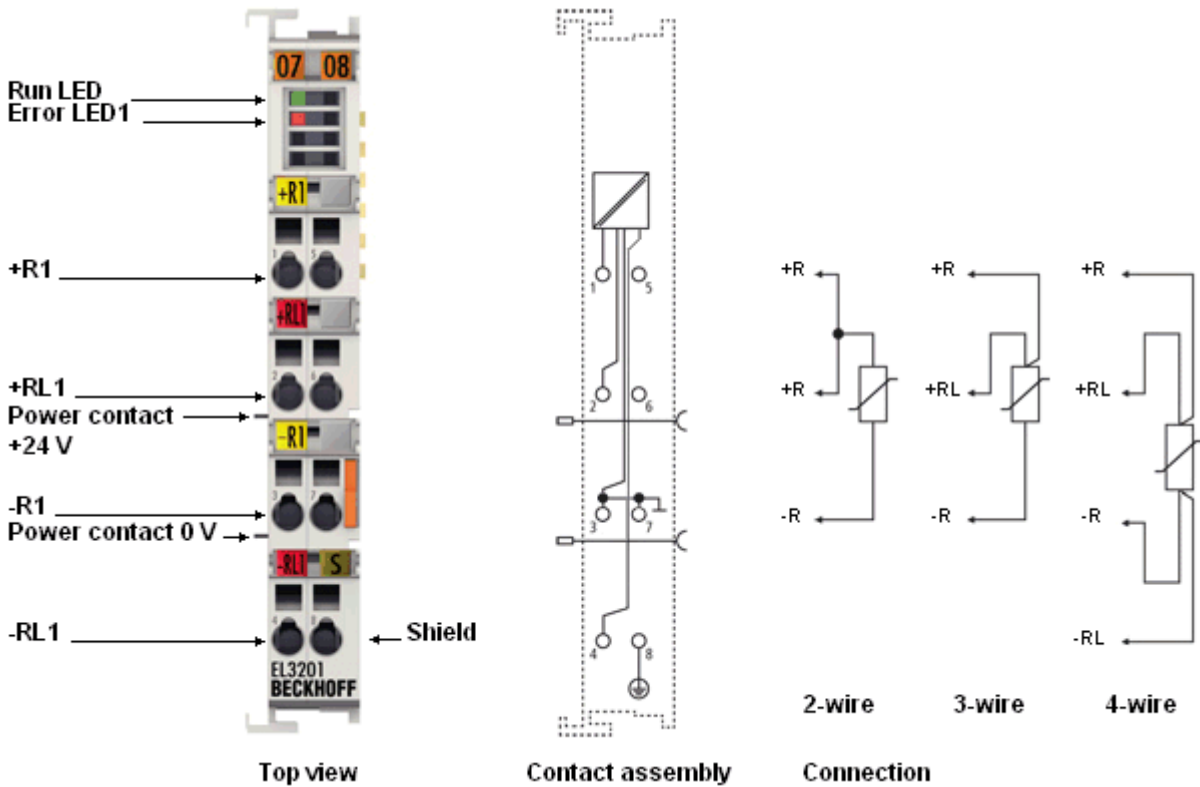


Fig. 37: EL3201 pin assignment

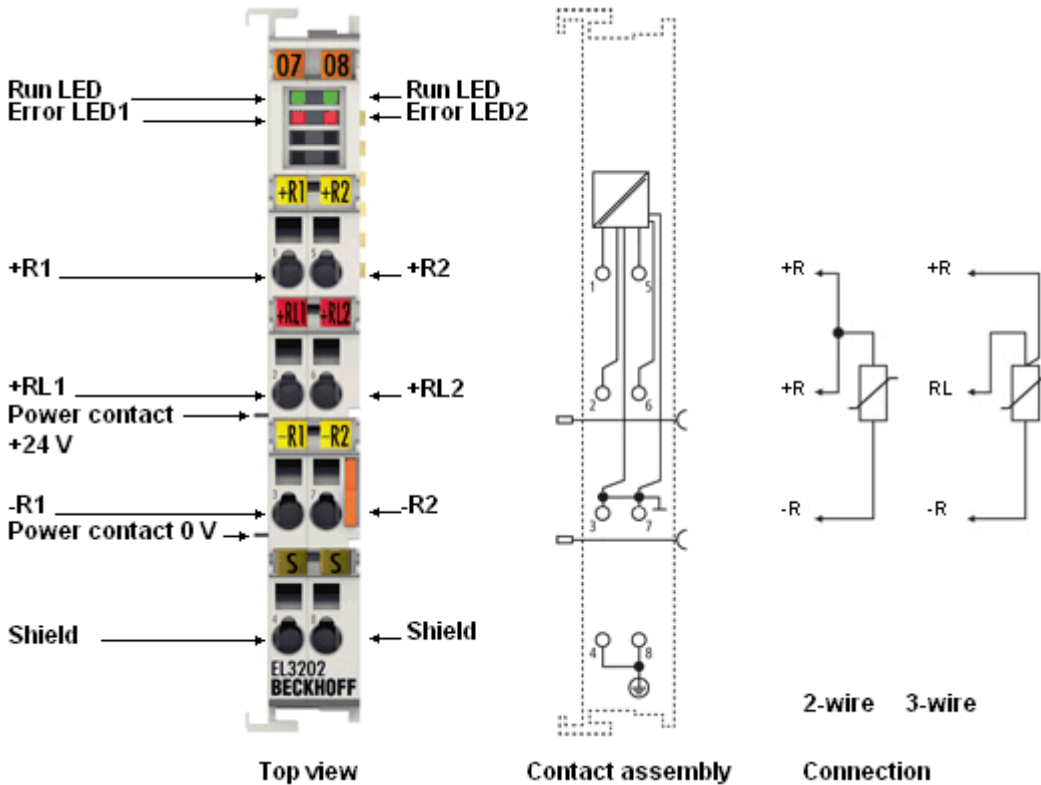


Fig. 38: EL3202 pin assignment

i Two-wire connection EL3201/EL3202

If the EL3201/EL3202 is operated with a two-wire connection, the inputs +R and +RL must be bridged by the user.

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT state machine: INIT = initialization of the terminal or BOOTSTRAP = function for firmware updates of the terminal
		flashing	State of the EtherCAT state machine: PREOP = function for mailbox communication and different standard-settings set
		Single flash	State of the EtherCAT state machine: SAFEOP = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
on	State of the EtherCAT state machine: OP = normal operating state; mailbox and process data communication is possible		
ERROR1, ERROR2**	red	Short circuit or wire breakage. The resistance is in the invalid range of the characteristic curve	

** EL3202 only

EL3201 pin assignment

Terminal point	No.	Comment
+R1	1	Input +R1
+RL1	2	Input +RL1
-R1	3	Input -R1
-RL1	4	Input -RL1
...	5	not used
...	6	not used
...	7	not used
Shield	8	Shield

EL3202 pin assignment

Terminal point	No.	Comment
+R1	1	Input +R1
+RL1	2	Input +RL1
-R1	3	Input -R1
Shield	4	Shield (internally connected to terminal point 8)
+R2	5	Input +R2
+RL2	6	Input +RL2
-R2	7	Input -R2
Shield	8	Shield (internally connected to terminal point 4)

EL3201-00x0, EL3202-00x0

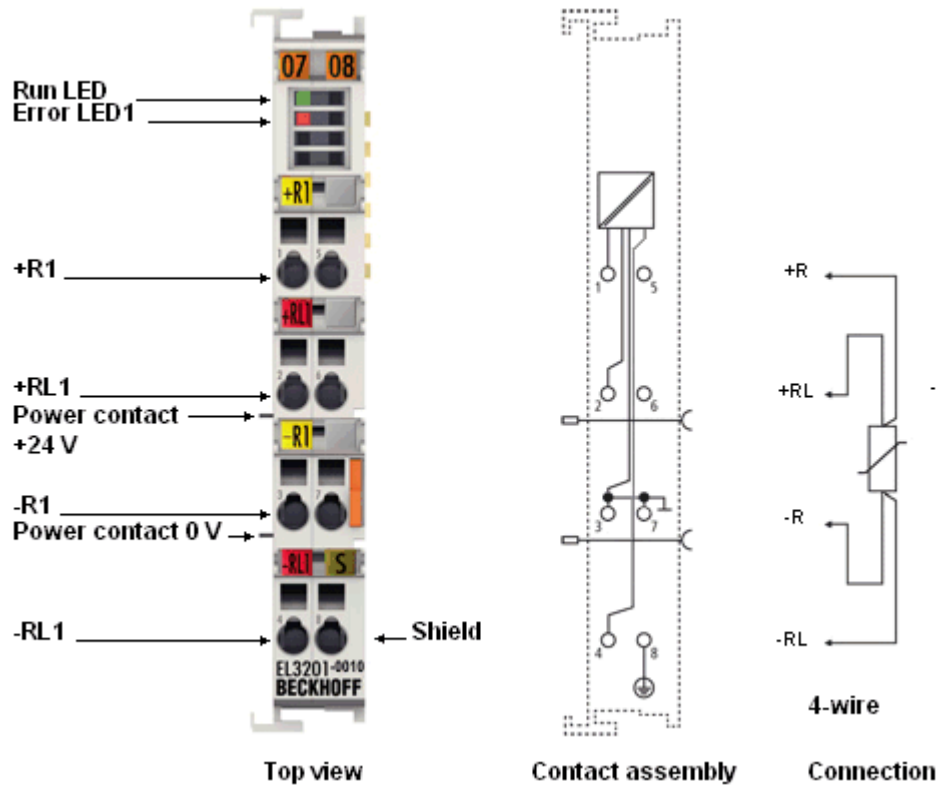


Fig. 39: *EL3201-00x0 pin assignment*

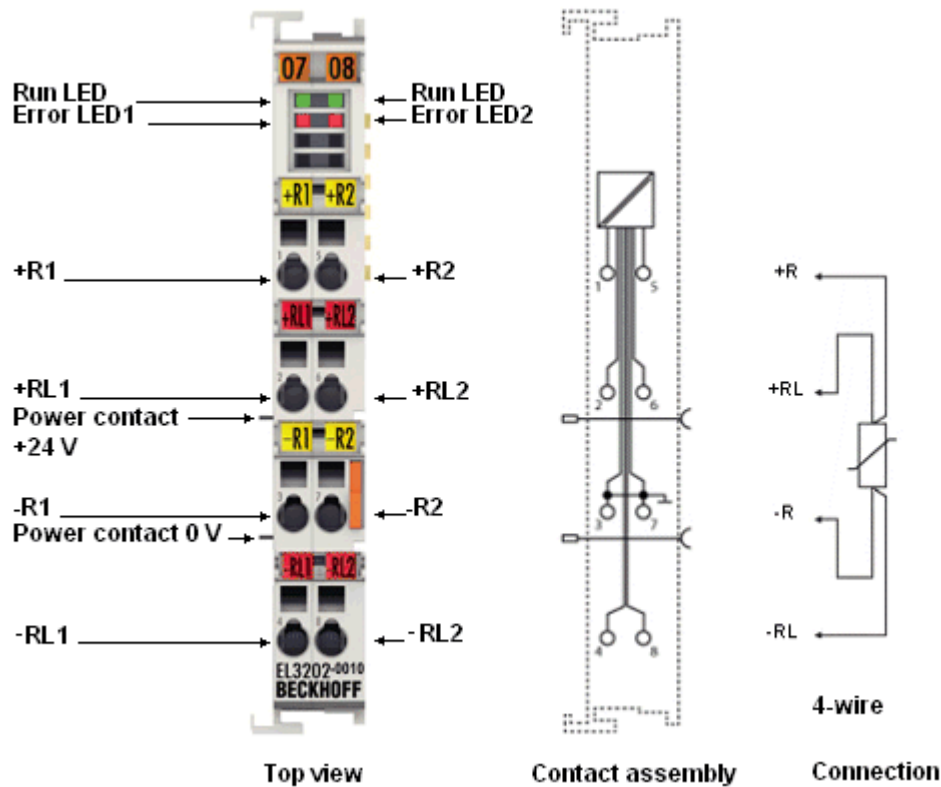


Fig. 40: *EL3202-00x0 pin assignment*

● Four-wire connection

i The EL320x-00x0 high-precision terminals may be operated as intended only using the 4-wire connection technique!

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT state machine: INIT = initialization of the terminal or BOOTSTRAP = function for firmware updates of the terminal
		flashing	State of the EtherCAT state machine: PREOP = function for mailbox communication and different standard-settings set
		Single flash	State of the EtherCAT state machine: SAFEOP = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT state machine: OP = normal operating state; mailbox and process data communication is possible
ERROR1, ERROR2**	red	Short circuit or wire breakage. The resistance is in the invalid range of the characteristic curve	

** EL3202-00x0 only

EL3201-00x0 pin assignment

Terminal point	No.	Comment
+R1	1	Input +R1
+RL1	2	Input +RL1
-R1	3	Input -R1
-RL1	4	Input -RL1
...	5	not used
...	6	not used
...	7	not used
Shield	8	Shield

EL3202-00x0 pin assignment

Terminal point	No.	Comment
+R1	1	Input +R1
+RL1	2	Input +RL1
-R1	3	Input -R1
-RL1	4	Input -RL1
+R2	5	Input +R2
+RL2	6	Input +RL2
-R2	7	Input -R2
-RL2	8	Input -RL2

EL3204-0x00

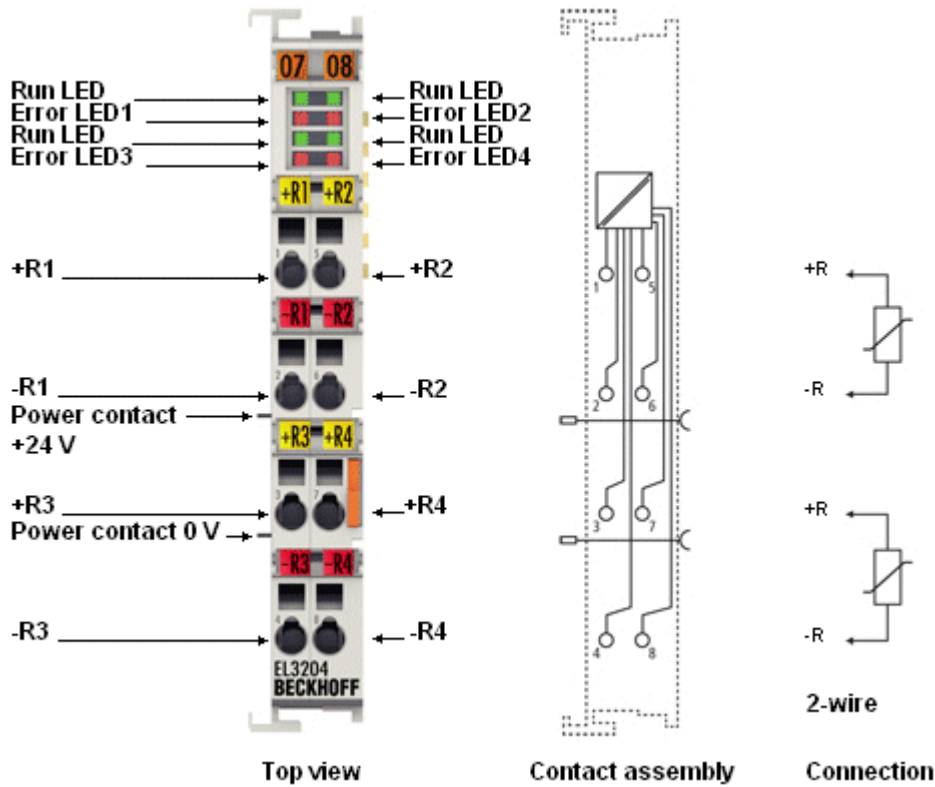


Fig. 41: EL3204-0x00 pin assignment

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT state machine: INIT = initialization of the terminal or BOOTSTRAP = function for firmware updates of the terminal
		flashing	State of the EtherCAT state machine: PREOP = function for mailbox communication and different standard-settings set
		Single flash	State of the EtherCAT state machine: SAFEOP = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT state machine: OP = normal operating state; mailbox and process data communication is possible
ERROR1 - 4	red	Short circuit or wire breakage. The resistance is in the invalid range of the characteristic curve.	

EL3204-0x00 pin assignment

Terminal point	No.	Comment
+R1	1	Input +R1
-R1	2	Input -R1
+R3	3	Input +R3
-R3	4	Input -R3
+R2	5	Input +R2
-R2	6	Input -R2
+R4	7	Input +R4
-R4	8	Input -R4

EL3214-00x0

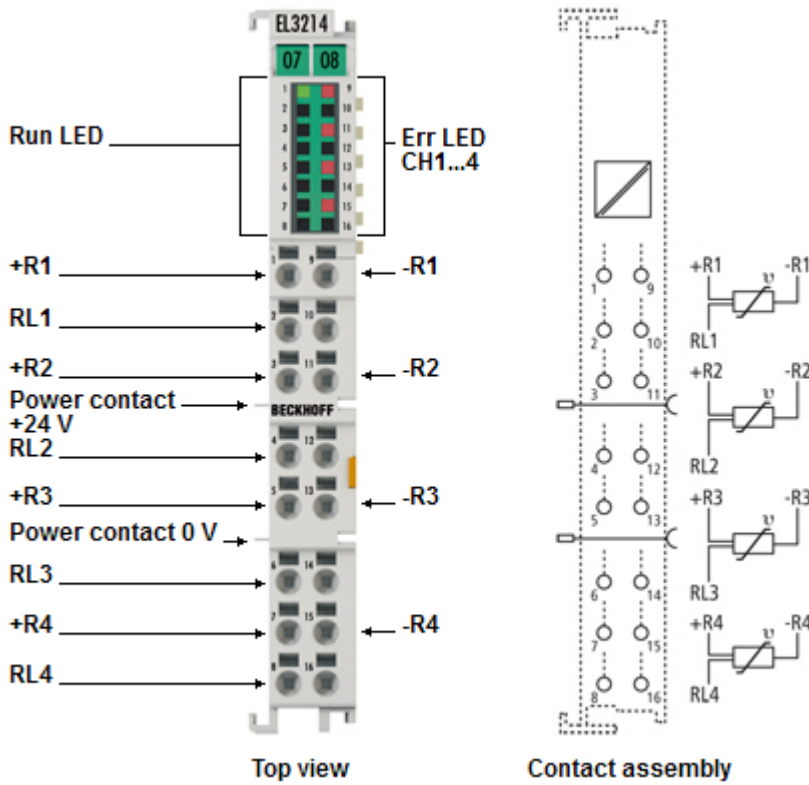


Fig. 42: EL3214 pin assignment

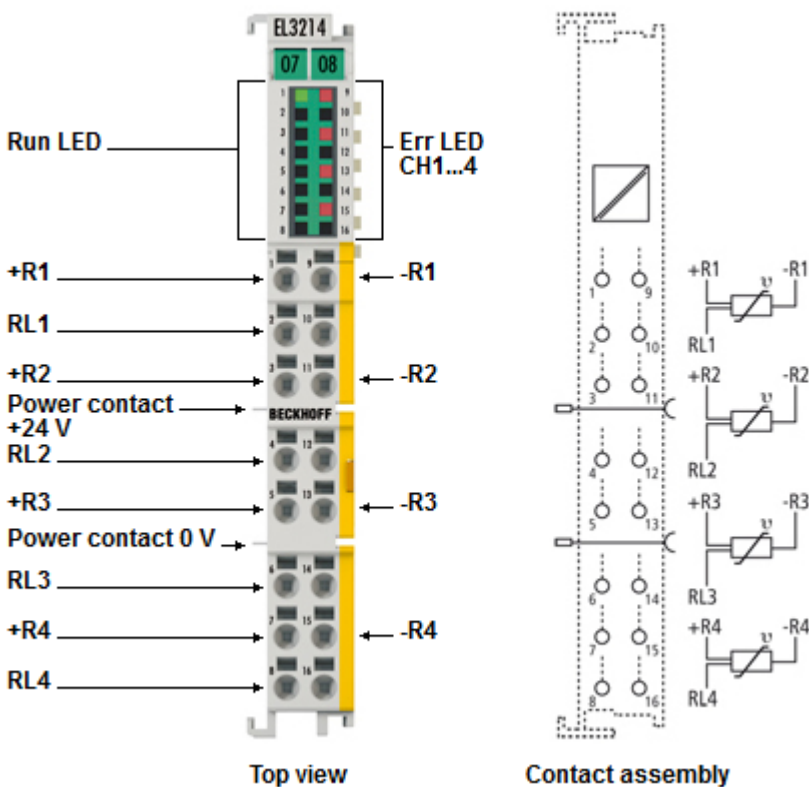


Fig. 43: EL3214-0090 pin assignment

● Two-wire connection EL3214-00x0

i If the EL3214 is operated with a two-wire connection, the inputs +R and +RL must be bridged by the user.

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT state machine: INIT = initialization of the terminal or BOOTSTRAP = function for firmware updates of the terminal
		flashing	State of the EtherCAT state machine: PREOP = function for mailbox communication and different standard-settings set
		Single flash	State of the EtherCAT state machine: SAFEOP = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT state machine: OP = normal operating state; mailbox and process data communication is possible
ERROR1 - 4	red	Short circuit or wire breakage. The resistance is in the invalid range of the characteristic curve.	

EL3214-00x0 pin assignment

Terminal point	No.	Comment
+R1	1	Input +R1
RL1	2	Input RL1
+R2	3	Input +R2
RL2	4	Input RL2
+R3	5	Input +R3
RL3	6	Input RL3
+R4	7	Input +R4
RL4	8	Input RL4
-R1	9	Input -R1
...	10	not used
-R2	11	Input -R2
...	12	not used
-R3	13	Input -R3
...	14	not used
-R4	15	Input -R4
...	16	not used

EL3208

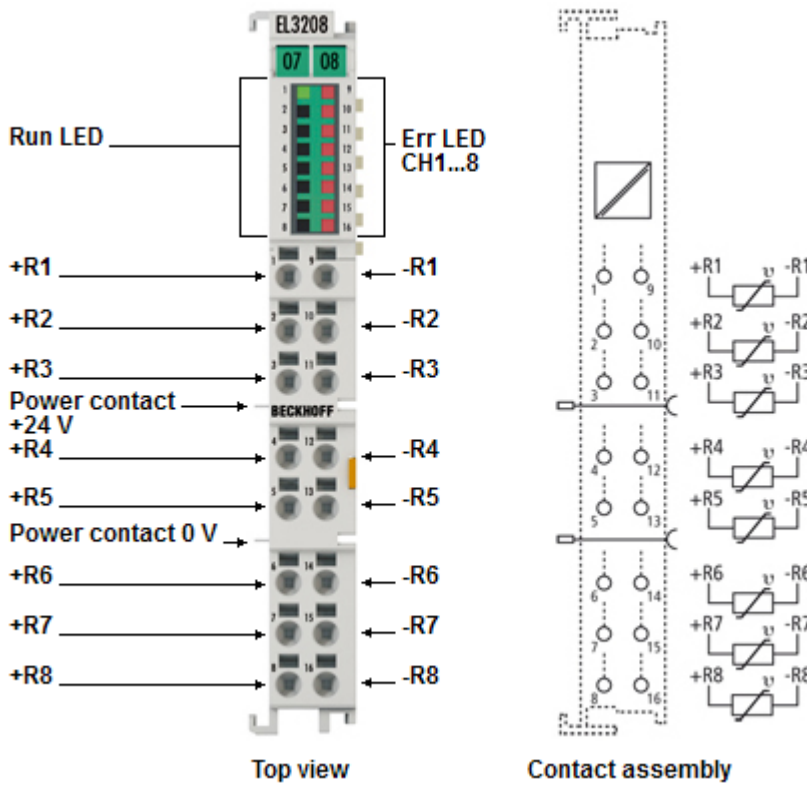


Fig. 44: EL3208 pin assignment

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT state machine: INIT = initialization of the terminal or BOOTSTRAP = function for firmware updates of the terminal
		flashing	State of the EtherCAT state machine: PREOP = function for mailbox communication and different standard-settings set
		Single flash	State of the EtherCAT state machine: SAFEOP = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT state machine: OP = normal operating state; mailbox and process data communication is possible
ERROR1 - 8	red	Short circuit or wire breakage. The resistance is in the invalid range of the characteristic curve.	

EL3208 pin assignment

Terminal point	No.	Comment
+R1	1	Input +R1
+R2	2	Input +R2
+R3	3	Input +R3
+R4	4	Input +R4
+R5	5	Input +R5
+R6	6	Input +R6
+R7	7	Input +R7
+R8	8	Input +R8
-R1	9	Input -R1
-R2	10	Input -R2
-R3	11	Input -R3
-R4	12	Input -R4
-R5	13	Input -R5
-R6	14	Input -R6

Terminal point	No.	Comment
-R7	15	Input -R7
-R8	16	Input -R8

4.16 Connection of analog RTD signal lines

The RTD input terminals of the EL32xx series measure the analog resistance of the sensor. The voltage drop at the sensor (including the line resistances, depending on the connection technology) is equivalent to the sensor resistance and therefore a measure for the sensor temperature, if the characteristic sensor curve is known. The following procedure serves for connecting analog signal cables in order to ensure error-free measurement of the analog signals.

Measures

- Sensor cable to be used
 - Tightly twisted
 - Shielded copper braid
 - Use low-impedance cable, particularly for 2-wire connection
- Keep the sensor and sensor cables free from external potential.
On no account should the GND connections (3/7 for EL3201, EL3202) be connected with other potentials.
- The resistor for the RTD sensor (e.g. 100 or 1000 Ω nominal) should be chosen based on the ratio between sensor resistance and line resistance, taking account of the connection type (2/3/4-wire).

Shielding measures

● Shielding measures

i Due to the complexity in the "EMC" area, there is no generally applicable guideline, but only technical measures in accordance with the state of the art, which can sometimes contradict each other. These must be checked for feasibility and effectiveness, taking into account the plant specifications, and applied by the plant installer following assessment.

The following notes on shielding are to be understood as technical suggestions that have proven themselves from time to time in practical use. It must be checked in each case which measures can be applied, depending on the installation and plant. The effectiveness of each measure must be checked individually. The formal transferability of measures to other types of plant is in general not possible.

Priority is to be given to typical national or general normative specifications.

A shielding approach is described below that in many cases improves the measurement quality. The suggested measures must be checked for feasibility and effectiveness in the actual plant.

- Apply the shield with a low resistance and enveloping the cable by 360°
- at the entry point into the control cabinet, the shield should be earthed conductively
- the shield should be earthed again at the terminal
 - at the terminal connection point, if present
 - if no terminal connection point is available, earth the shield as close to the terminal as possible.
 - to avoid ground loops the shield can be undone after entry into the control cabinet.
A capacitive connection to the terminal shield contact is possible.
 - avoid unshielded cable lengths of > 50 cm!

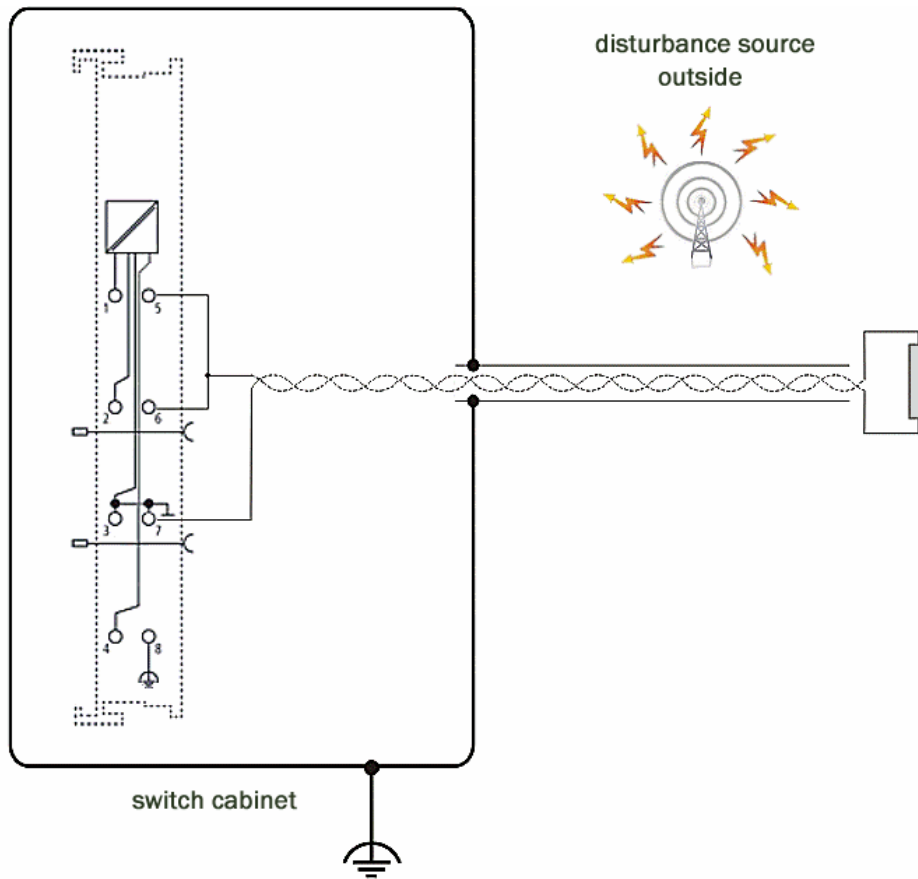


Fig. 45: Example of EL3202-0000 shield connection with shield contact, in the case of potential interference sources inside the control cabinet

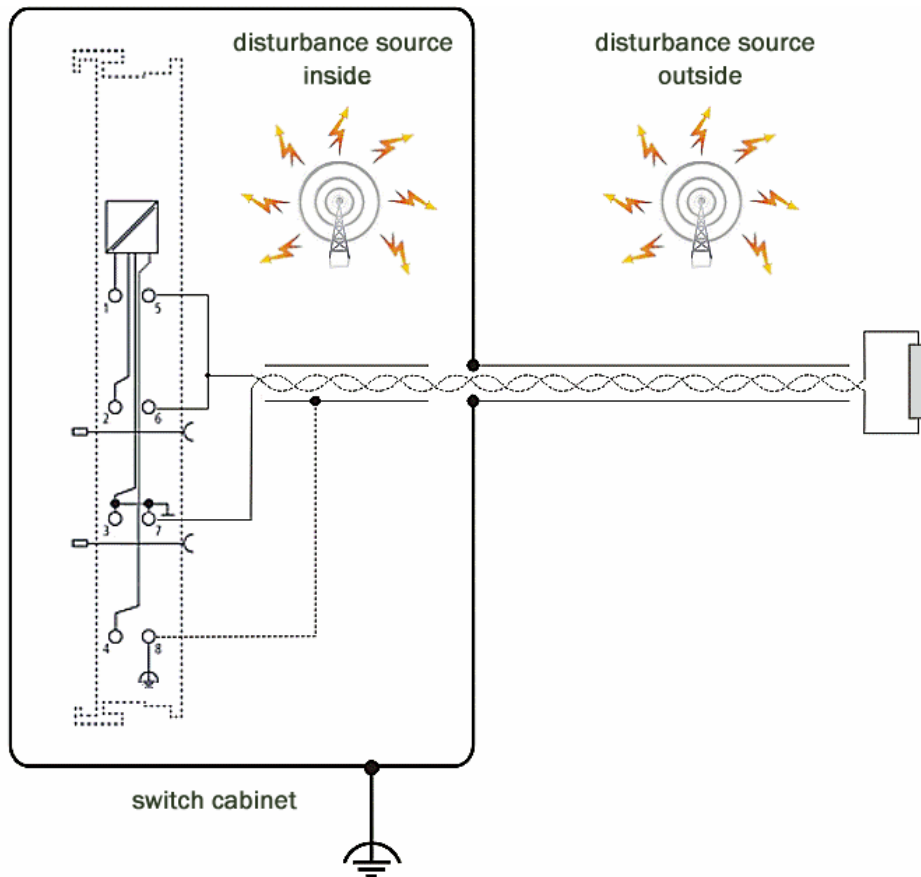


Fig. 46: Example of EL3202-0000 shield connection with shield contact, in the case of potential interference sources inside and outside the control cabinet

5 Commissioning

5.1 TwinCAT Quick Start

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

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

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

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

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

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

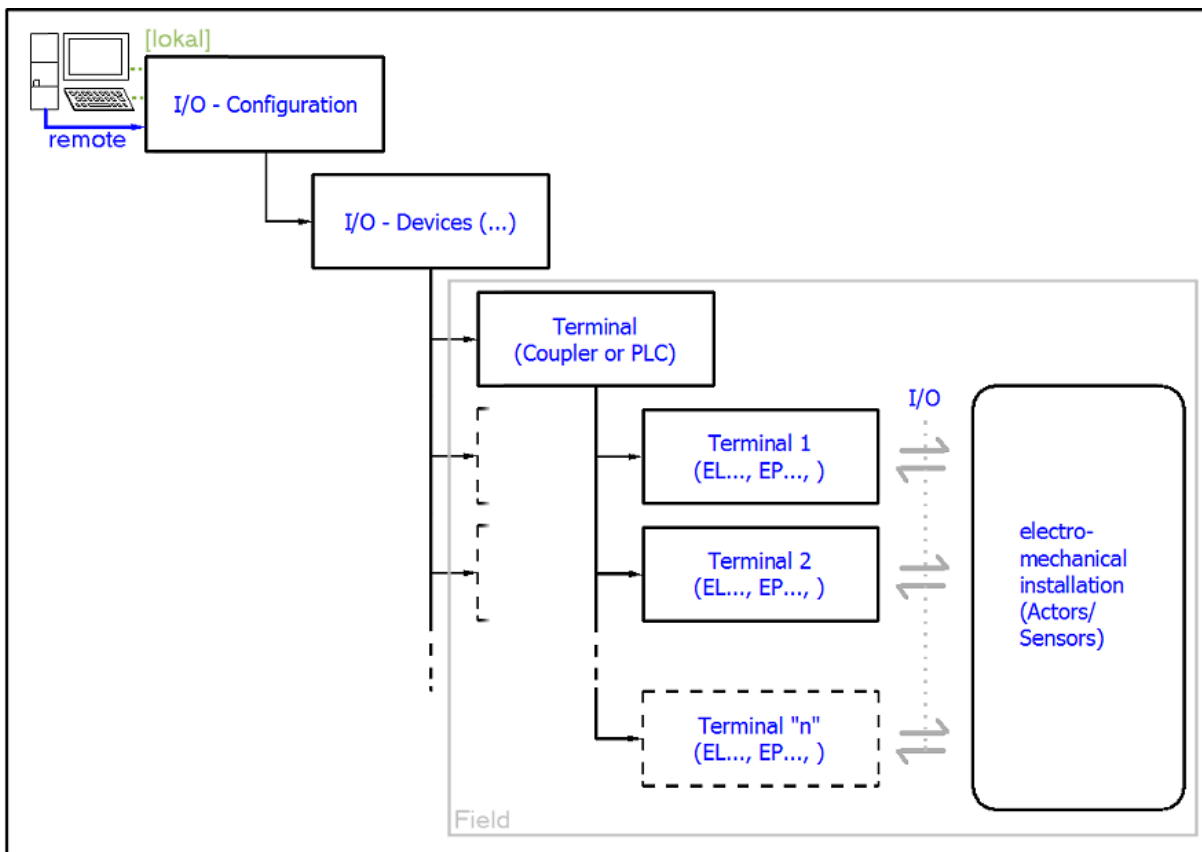


Fig. 47: Relationship between user side (commissioning) and installation

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

Sample configuration (actual configuration)

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

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

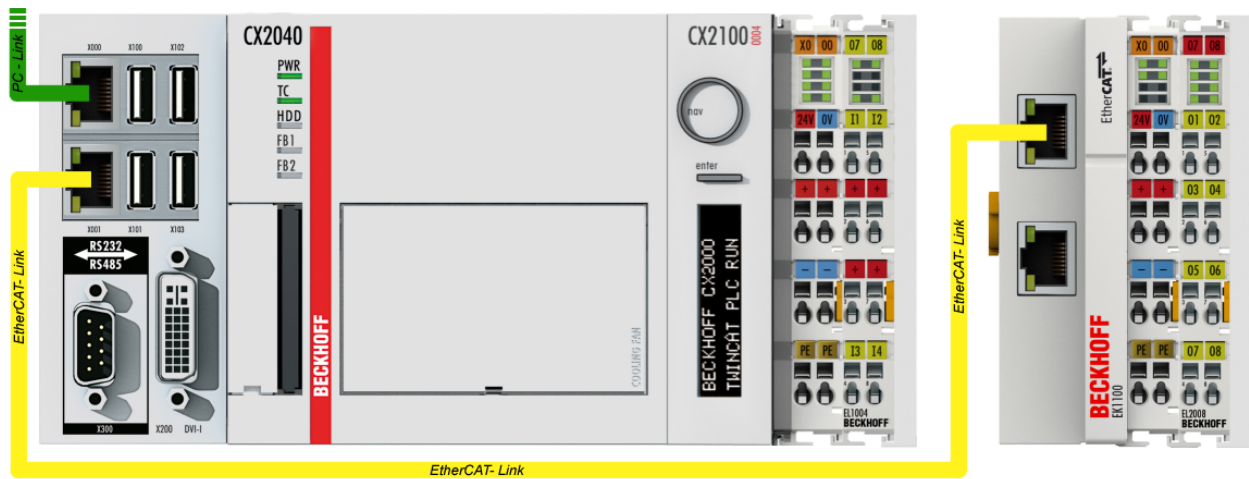


Fig. 48: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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

5.1.1 TwinCAT 2

Startup

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

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

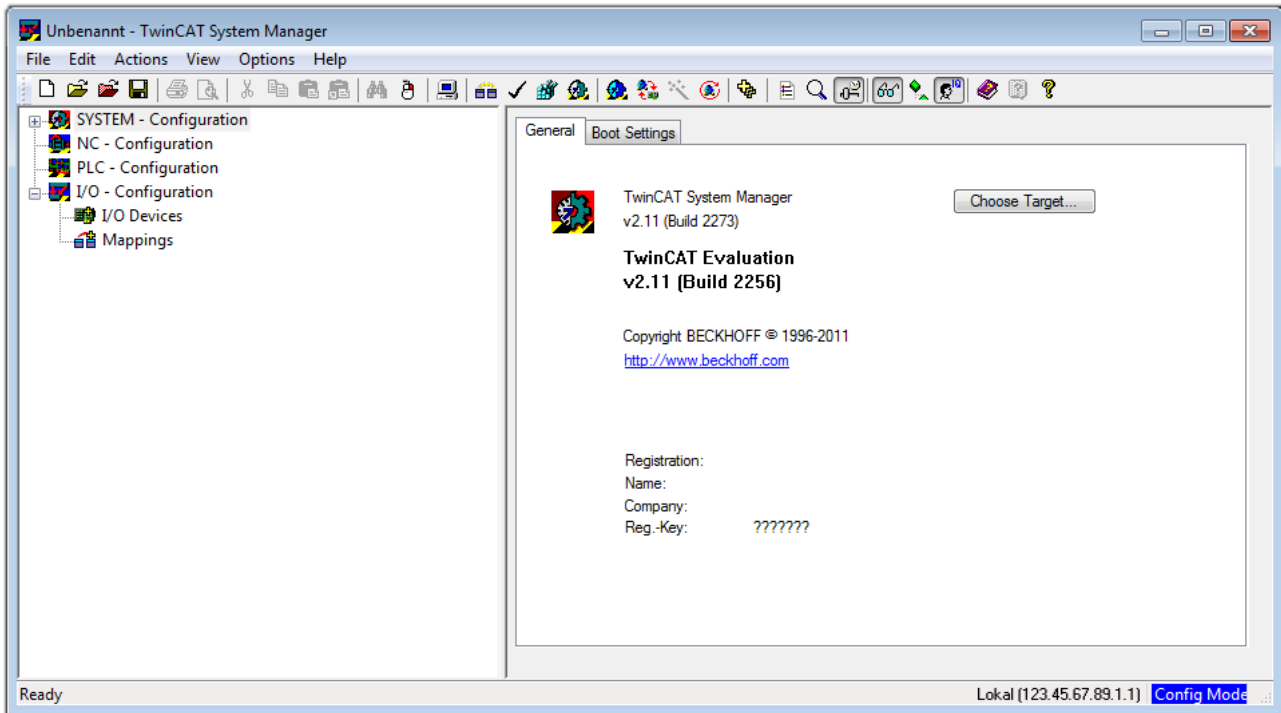



Fig. 49: Initial TwinCAT 2 user interface

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 [▶ 77]".

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:

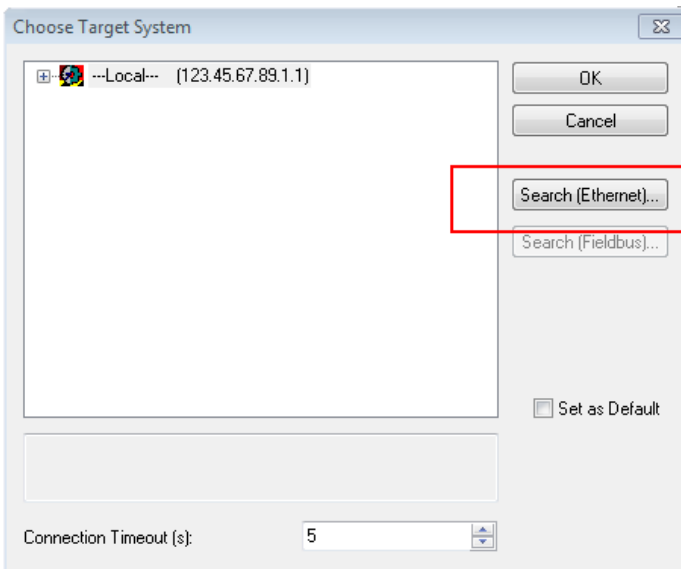


Fig. 50: Selection of the target system

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.

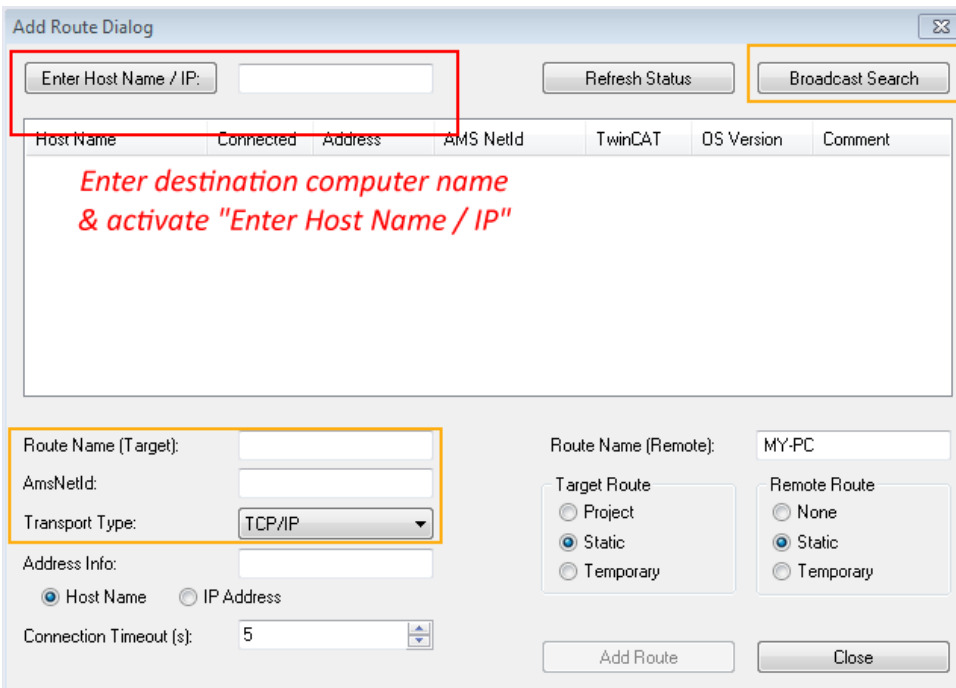
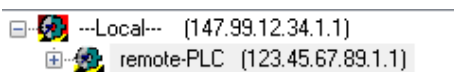


Fig. 51: Specify the PLC for access by the TwinCAT System Manager: selection of the target system



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).

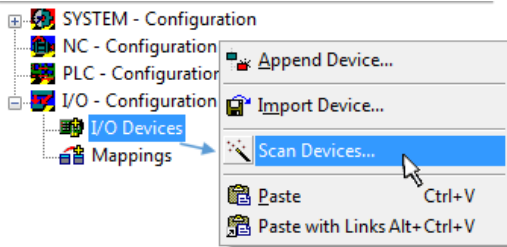


Fig. 52: Select "Scan Devices..."

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

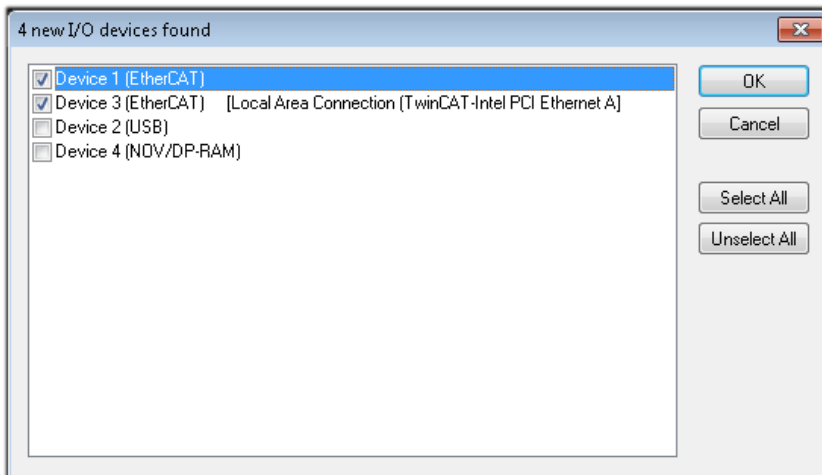


Fig. 53: Automatic detection of I/O devices: selection the devices to be integrated

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 \[▶ 73\]](#) described at the beginning of this section, the result is as follows:

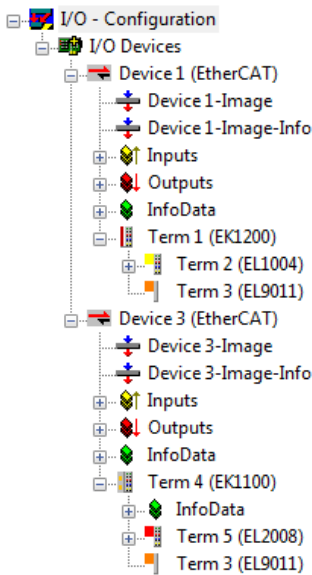


Fig. 54: Mapping of the configuration in the TwinCAT 2 System Manager

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:

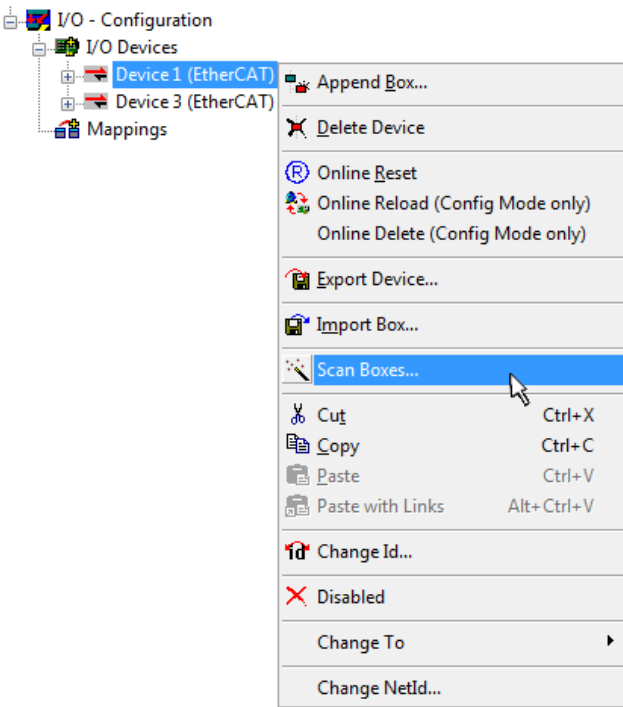


Fig. 55: Reading of individual terminals connected to a device

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:

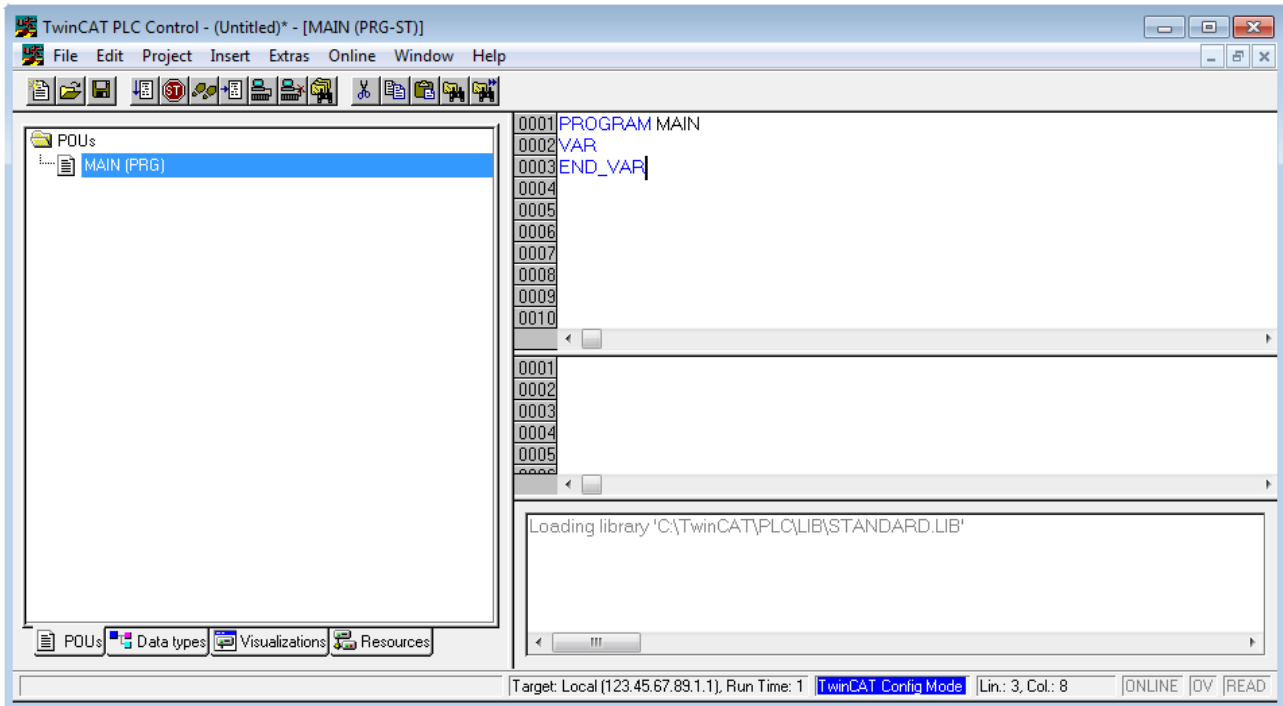


Fig. 56: TwinCAT PLC Control after startup

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

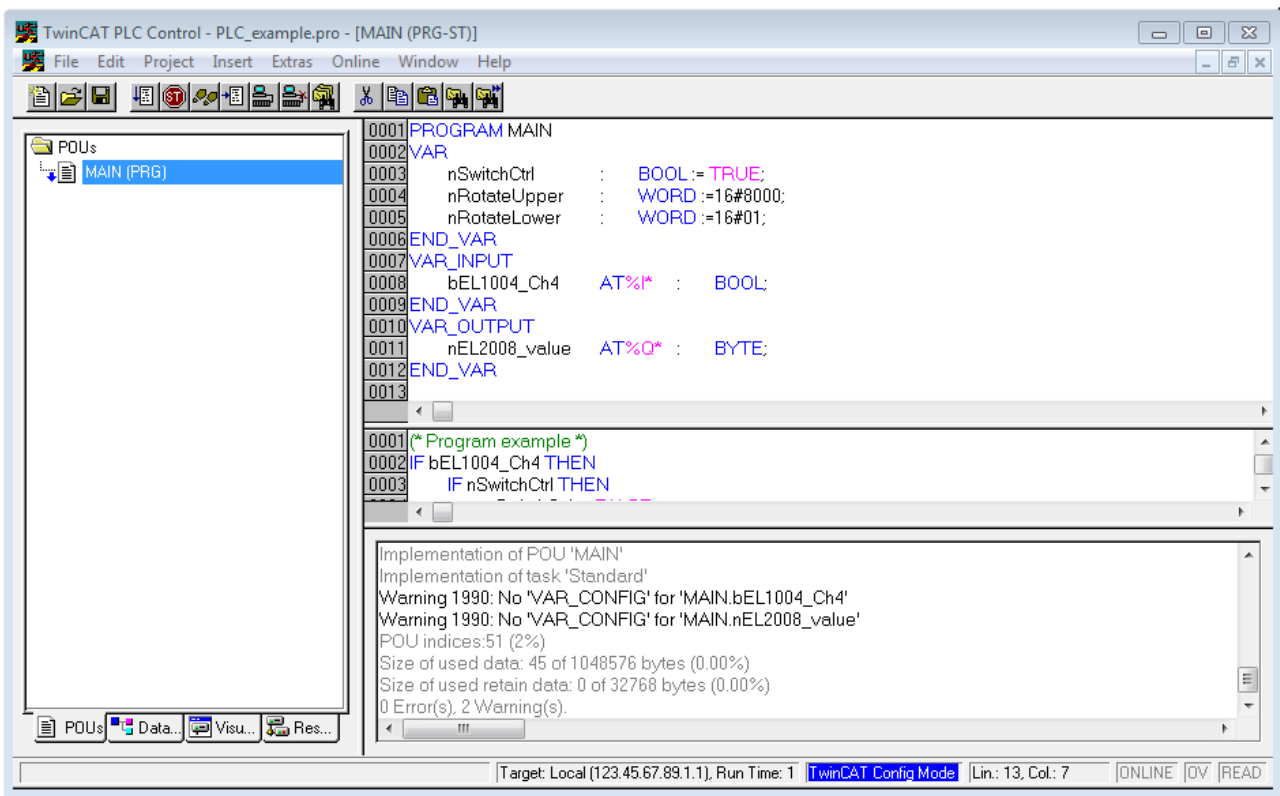


Fig. 57: Sample program with variables after a compile process (without variable integration)

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...":

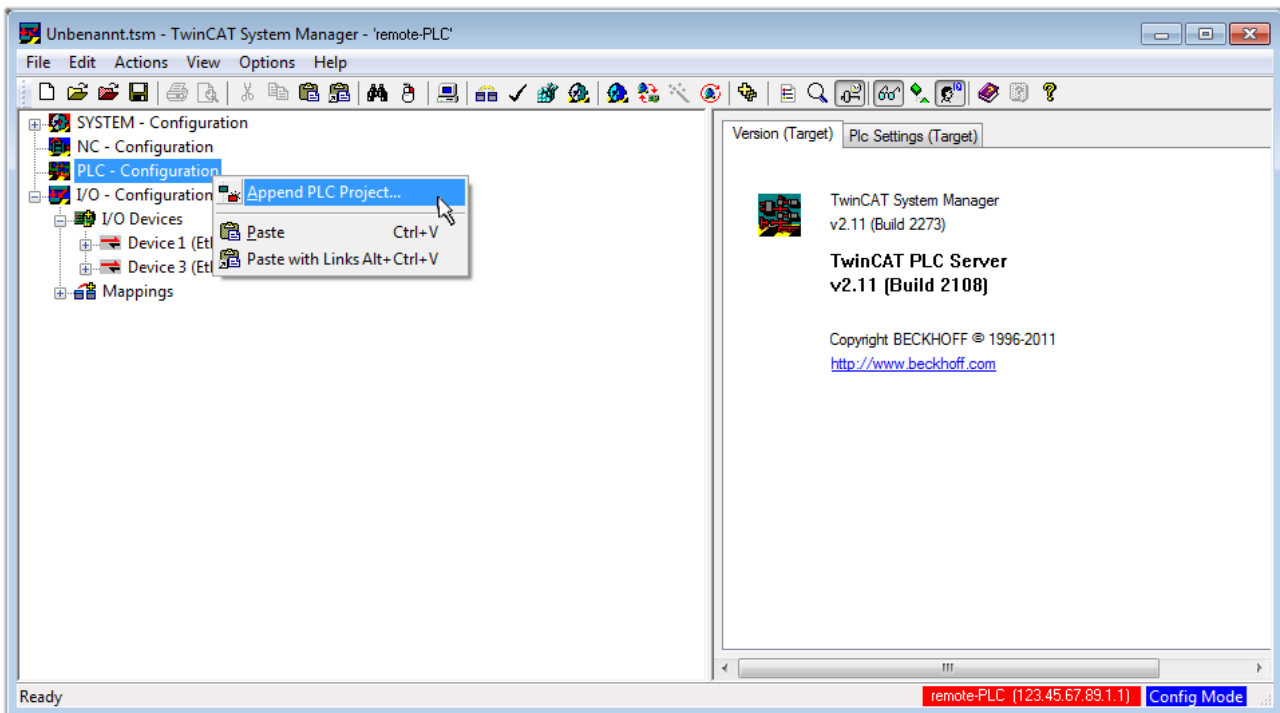


Fig. 58: Appending the TwinCAT PLC Control project

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:

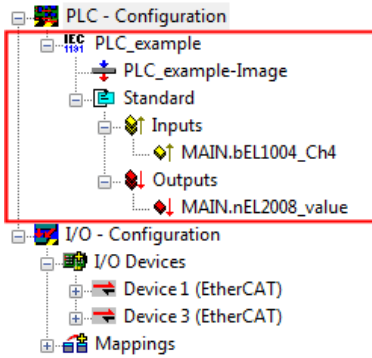


Fig. 59: PLC project integrated in the PLC configuration of the System Manager

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":

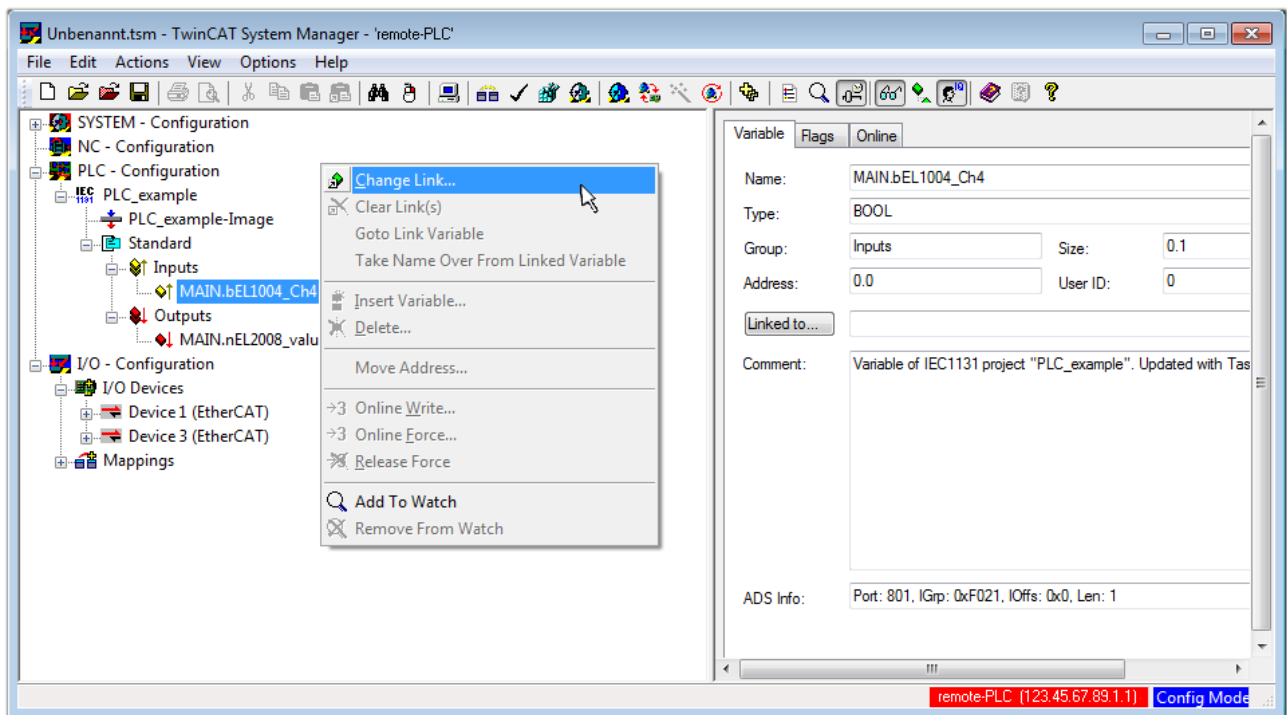


Fig. 60: Creating the links between PLC variables and process objects

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

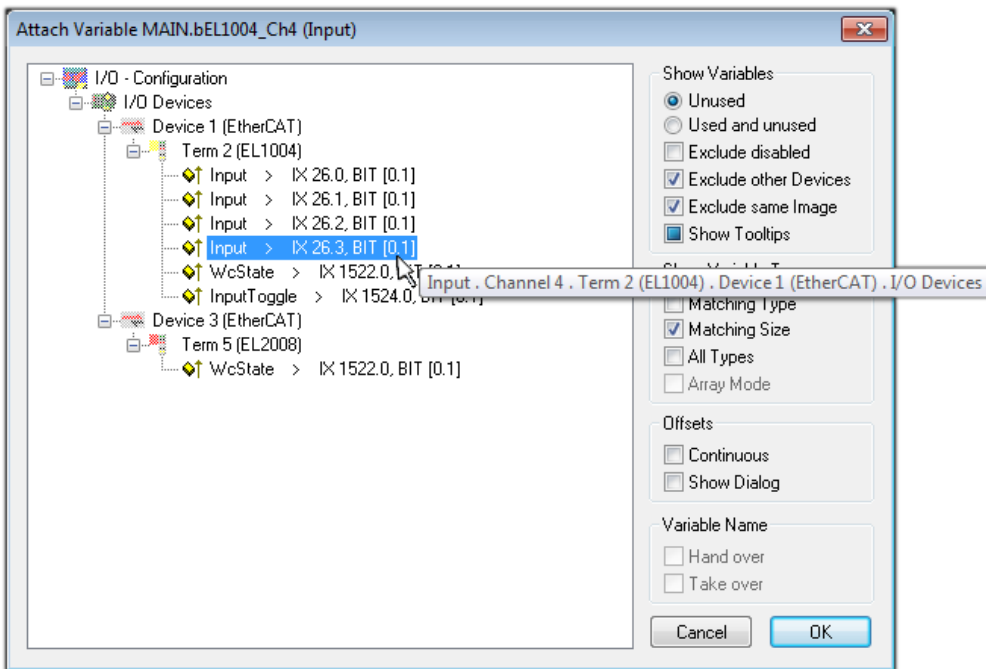


Fig. 61: Selecting PDO of type BOOL

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:

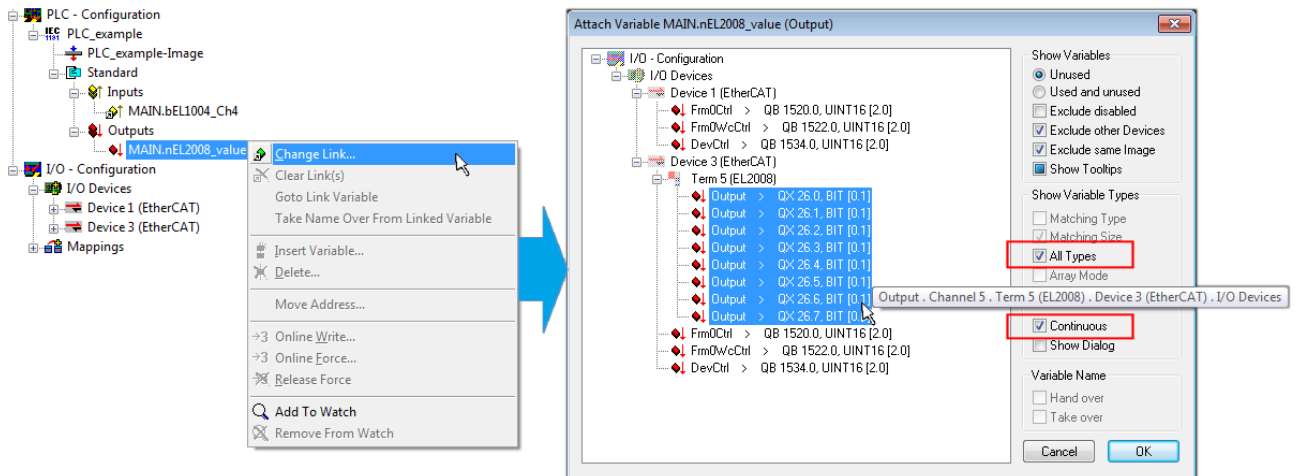



Fig. 62: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

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:

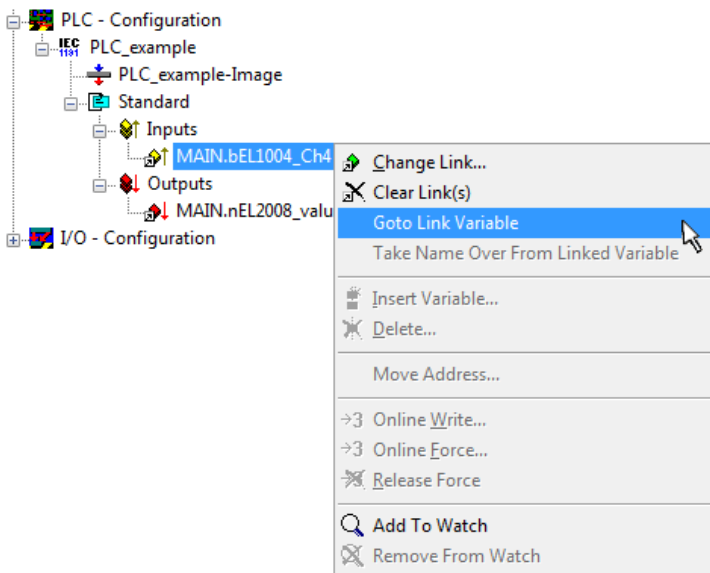

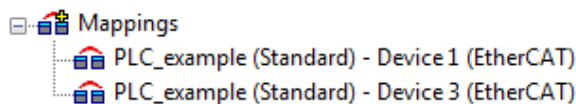


Fig. 63: Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample

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...":

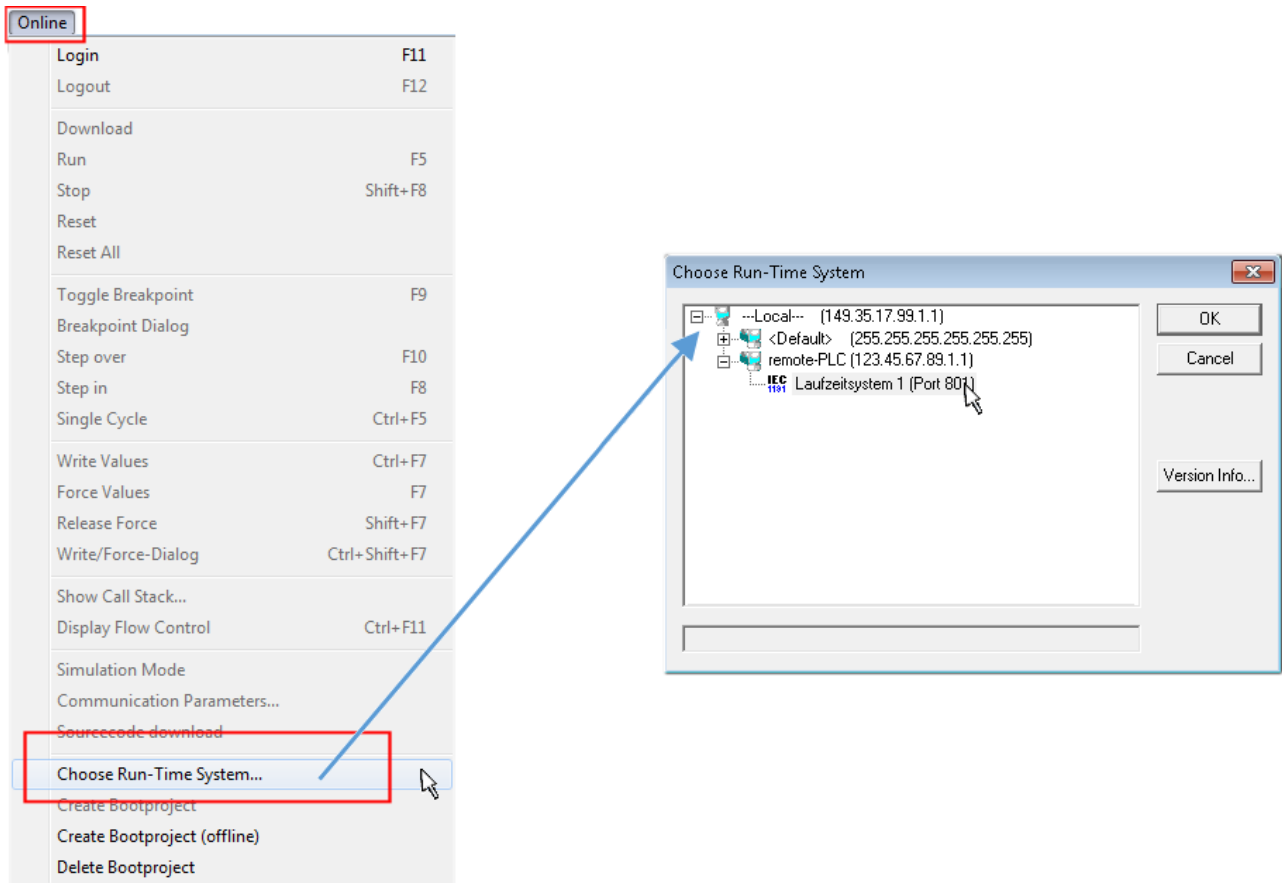



Fig. 64: Choose target system (remote)

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:

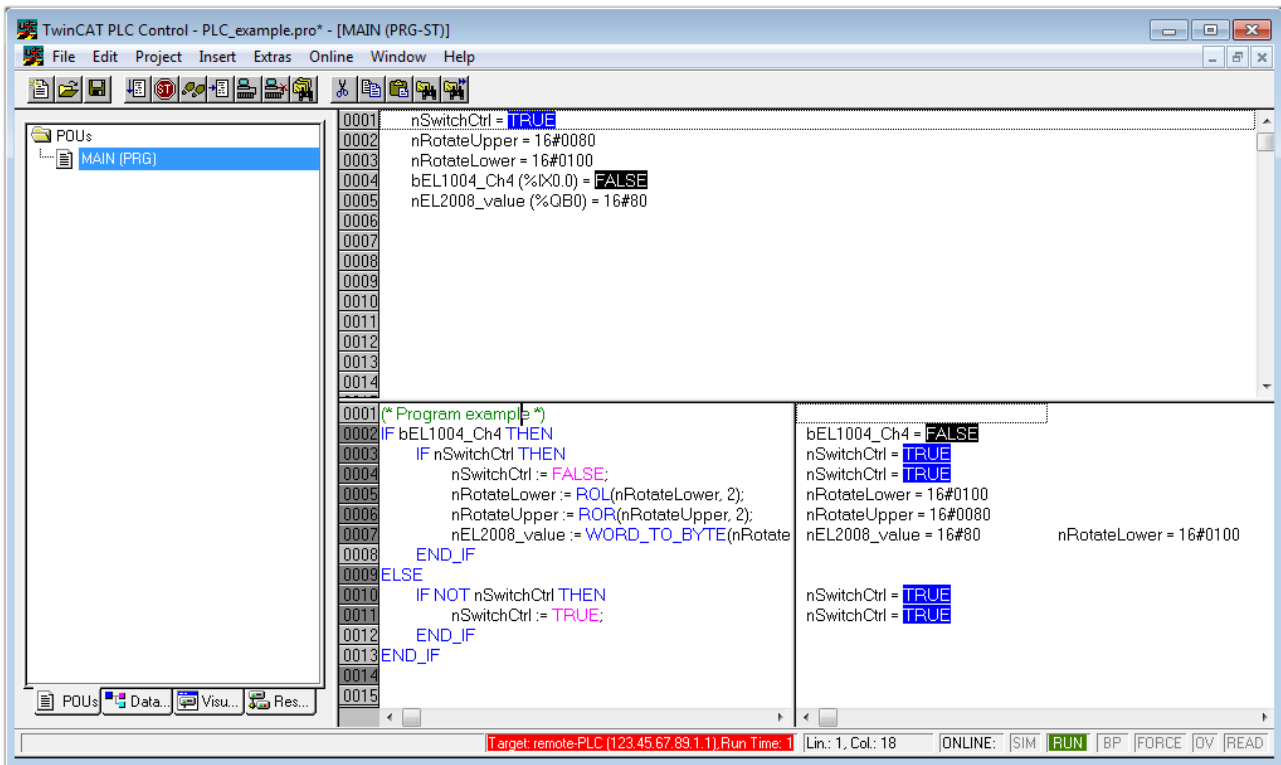


Fig. 65: PLC Control logged in, ready for program startup

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

5.1.2 TwinCAT 3


Startup

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

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



Fig. 66: 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):

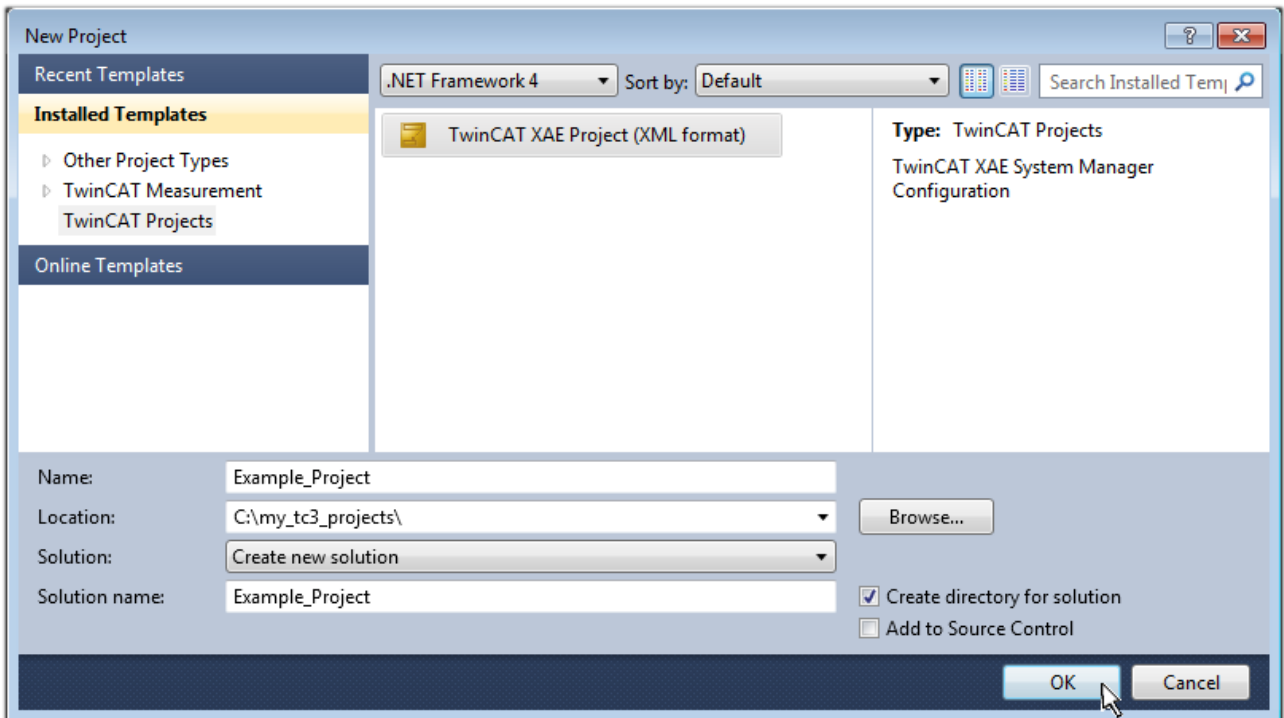


Fig. 67: Create new TwinCAT project

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

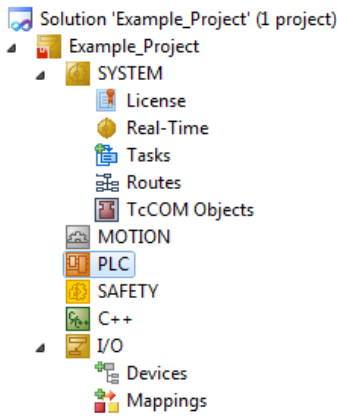
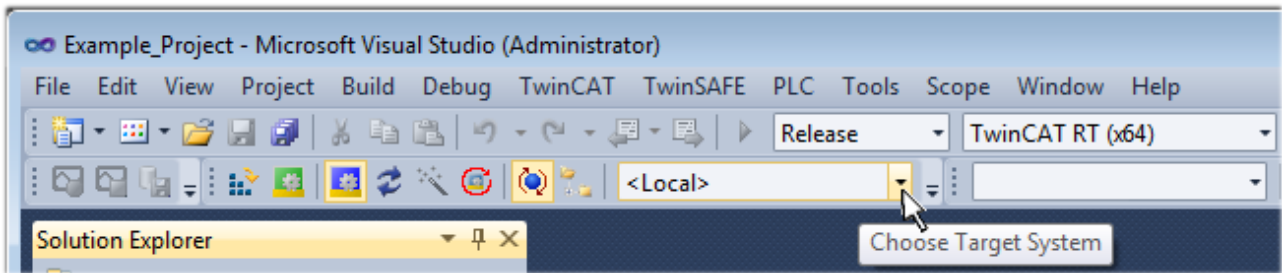


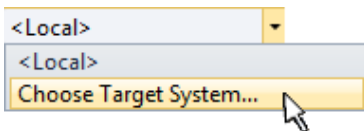
Fig. 68: New TwinCAT3 project in the project folder explorer

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 [▶ 88]".

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:

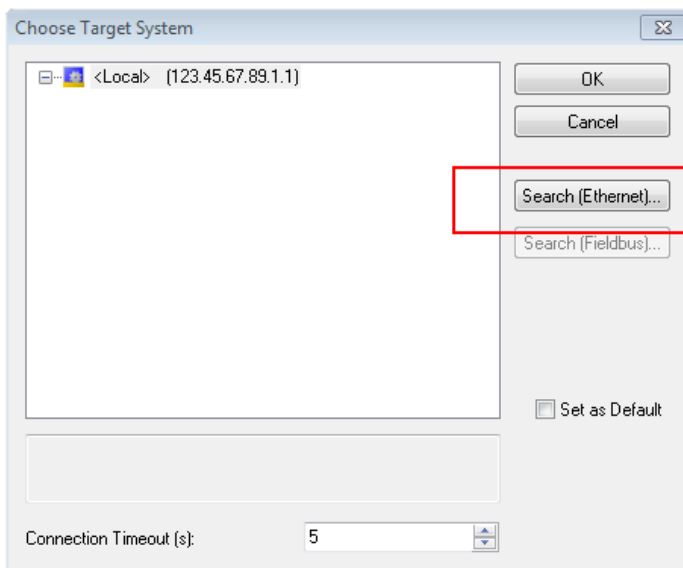


Fig. 69: Selection dialog: Choose the target system

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.

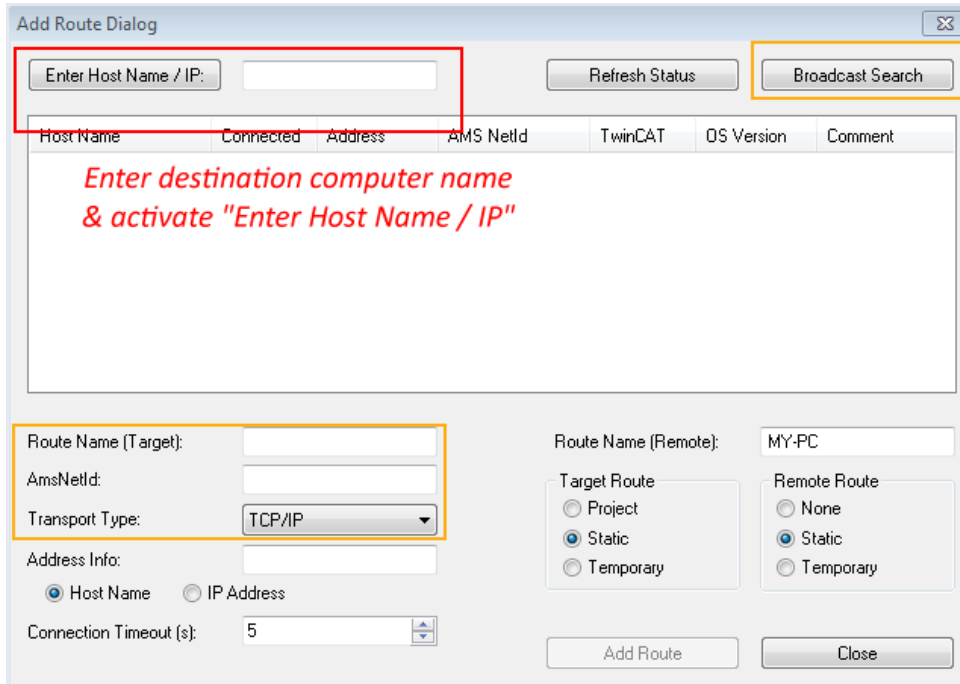
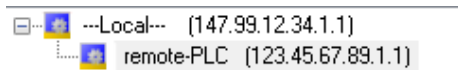


Fig. 70: Specify the PLC for access by the TwinCAT System Manager: selection of the target system


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)".

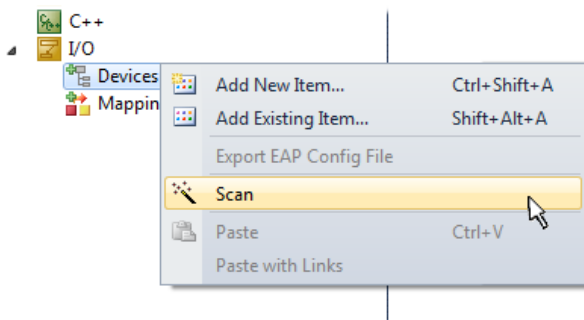


Fig. 71: Select "Scan"

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

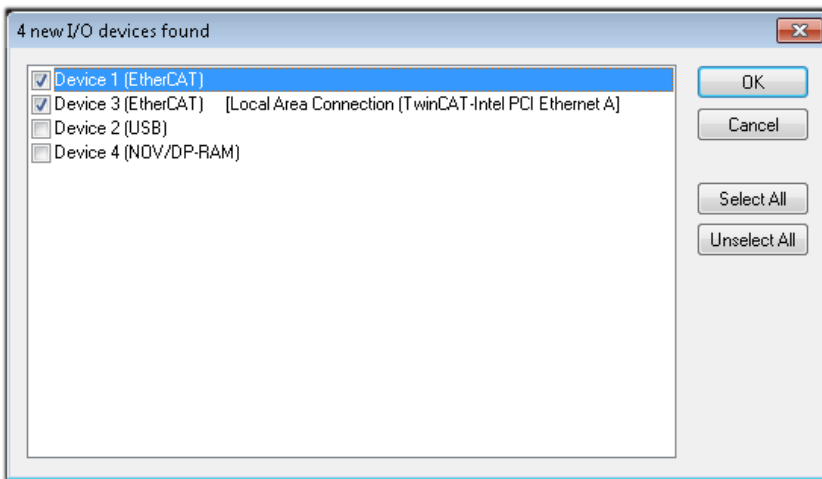


Fig. 72: Automatic detection of I/O devices: selection the devices to be integrated

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 \[▶ 73\]](#) described at the beginning of this section, the result is as follows:

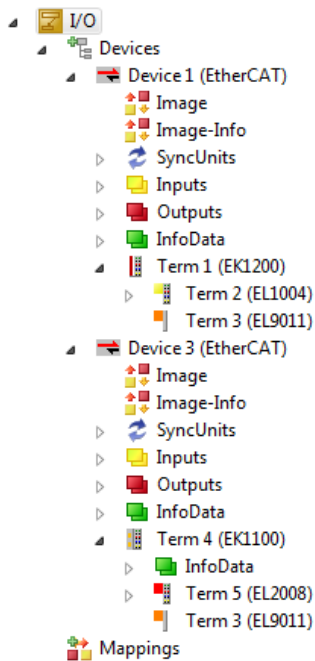


Fig. 73: Mapping of the configuration in VS shell of the TwinCAT3 environment

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:

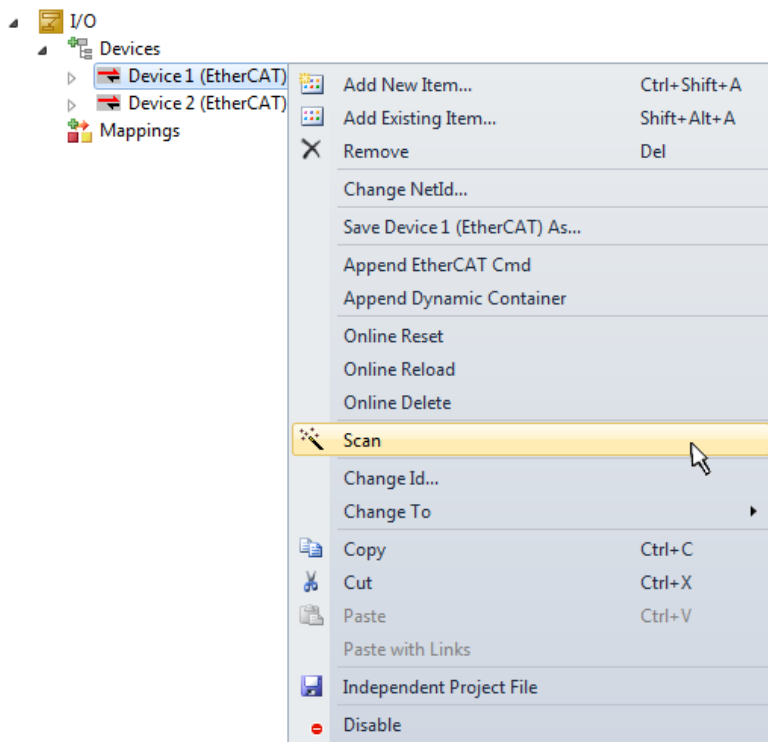


Fig. 74: Reading of individual terminals connected to a device

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....":

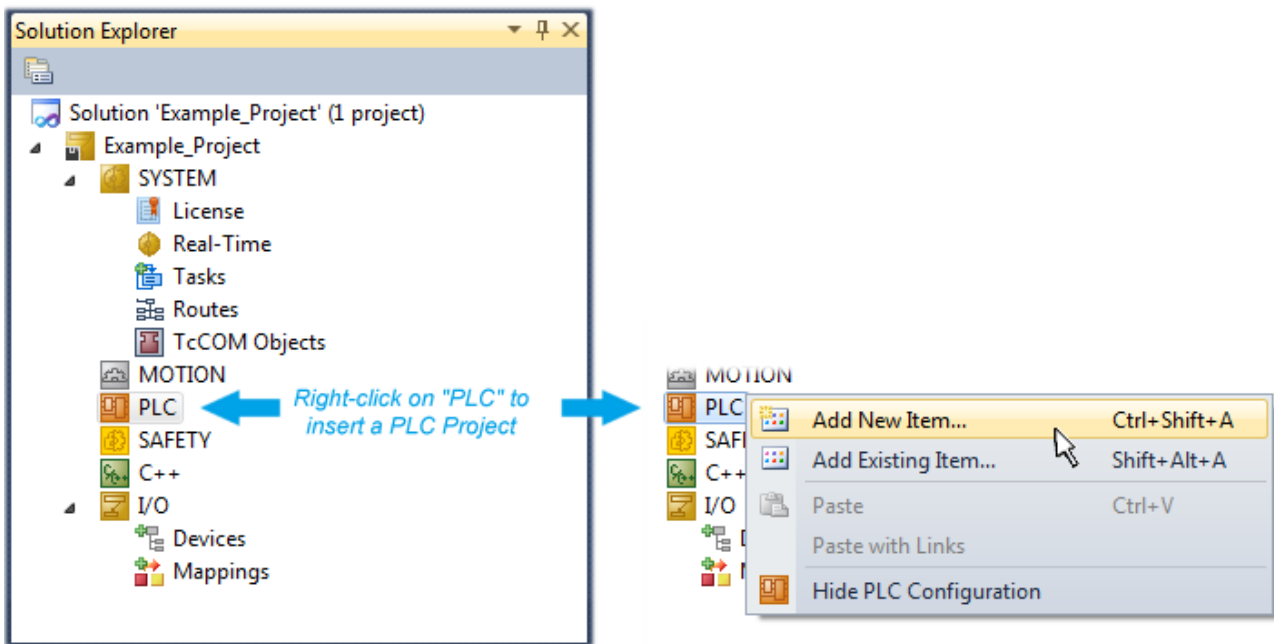


Fig. 75: Adding the programming environment in "PLC"

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

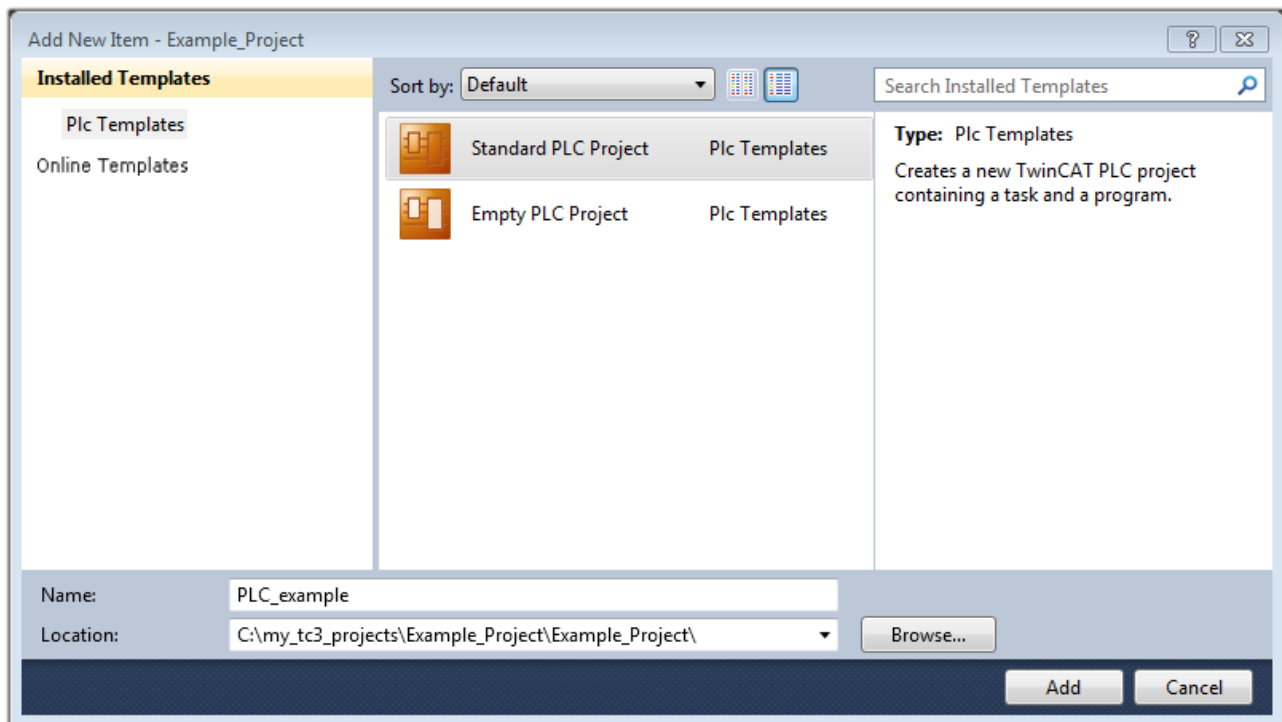


Fig. 76: Specifying the name and directory for the PLC programming environment

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:

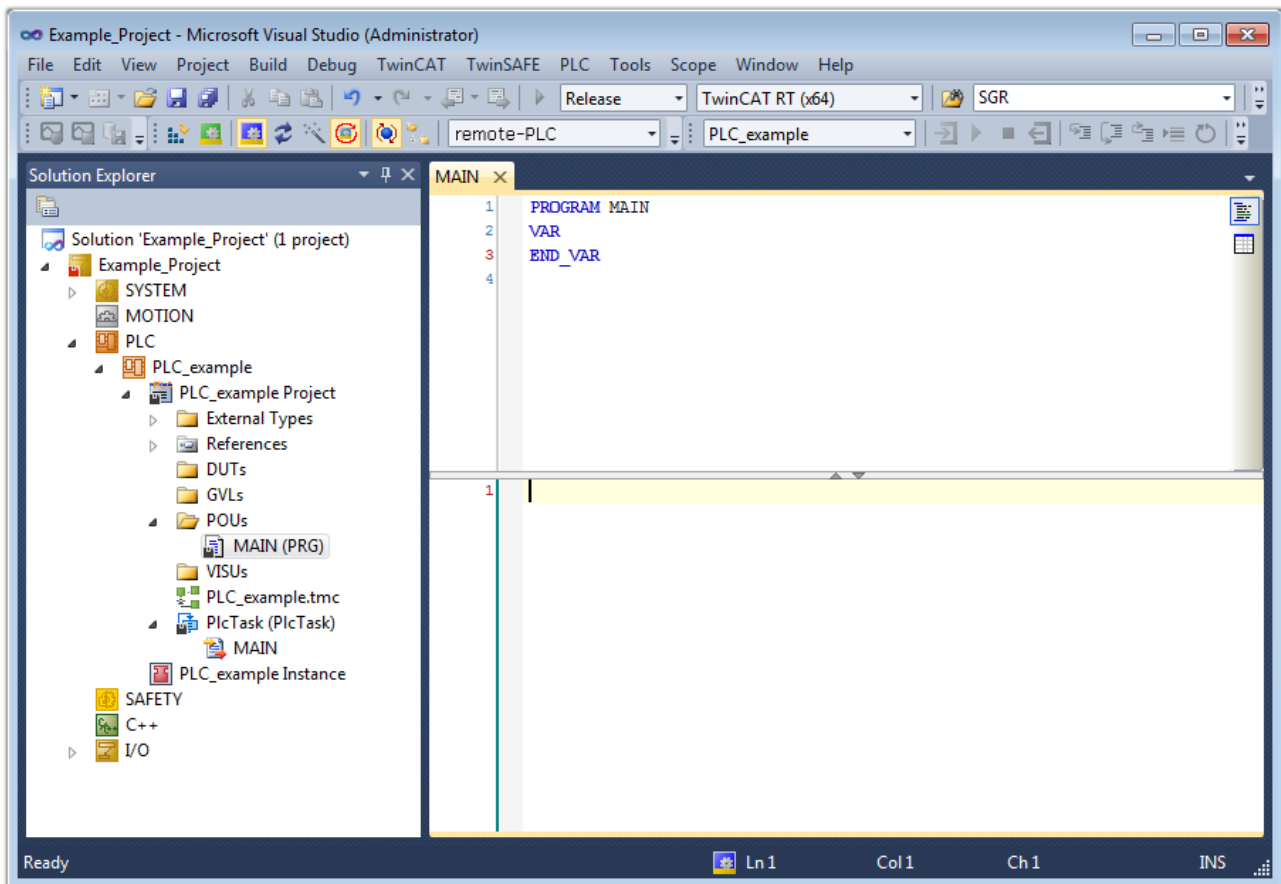


Fig. 77: Initial "Main" program of the standard PLC project

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

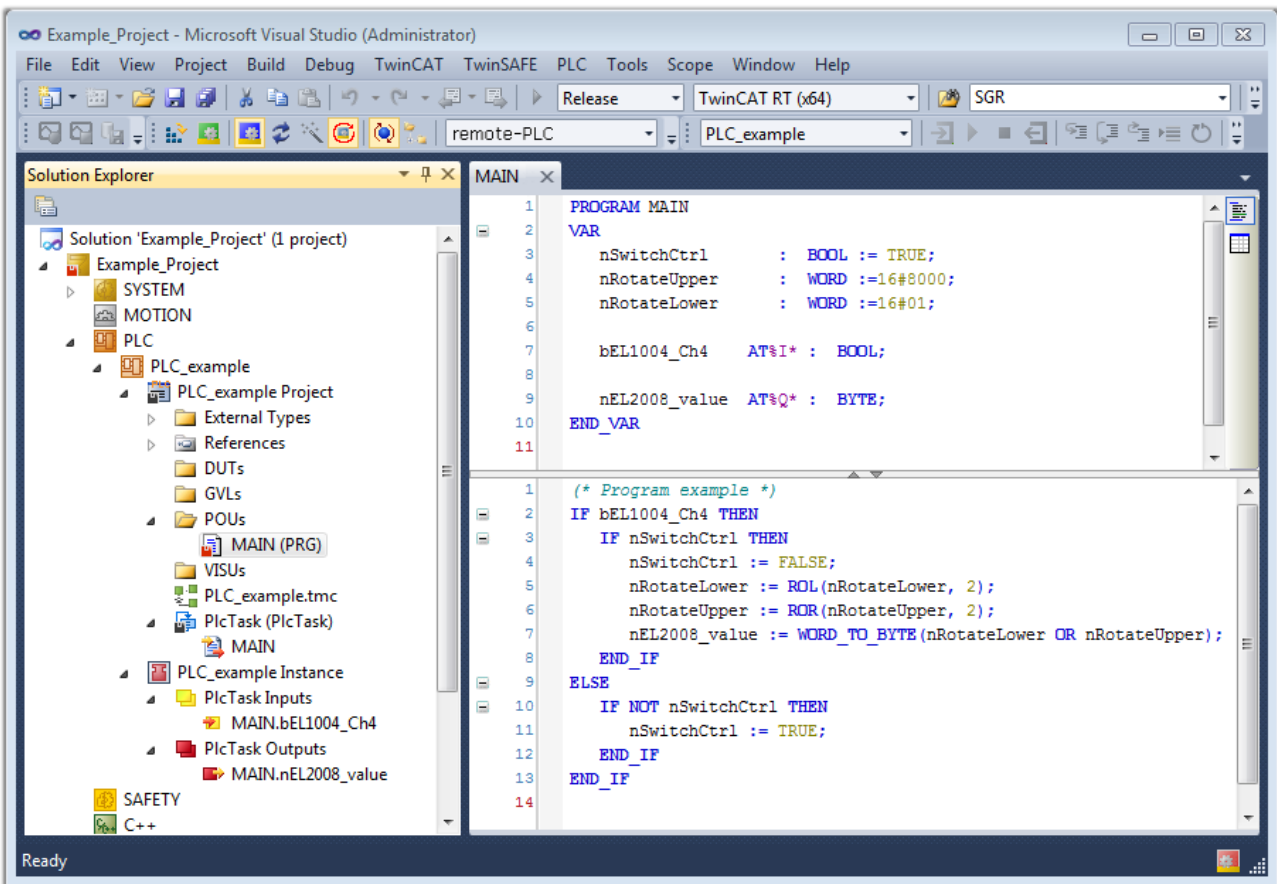


Fig. 78: Sample program with variables after a compile process (without variable integration)

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

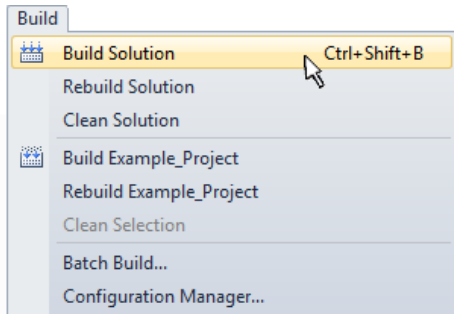
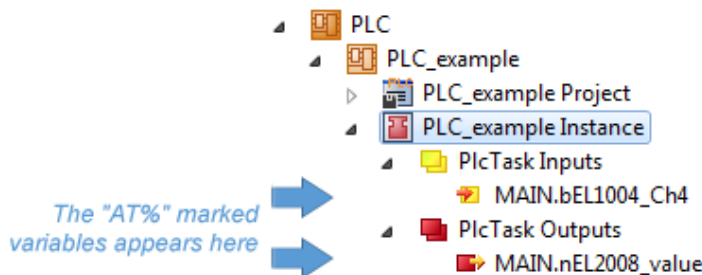


Fig. 79: Start program compilation

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:

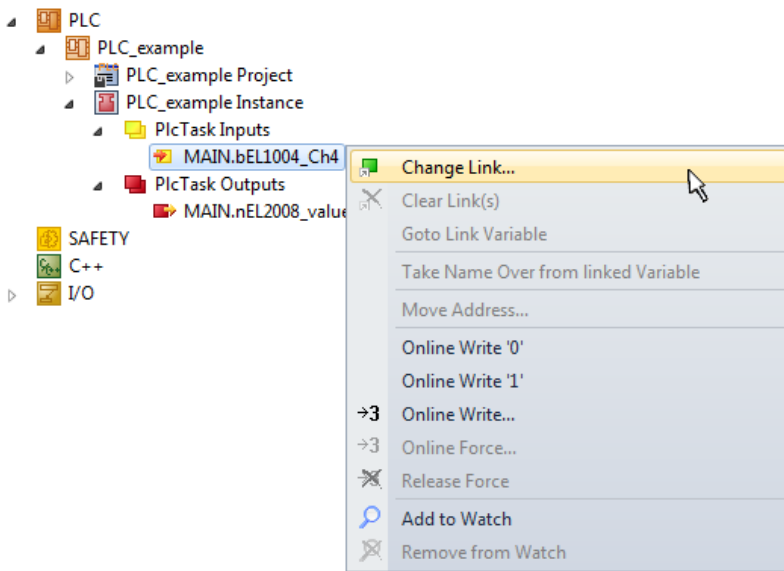


Fig. 80: Creating the links between PLC variables and process objects

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

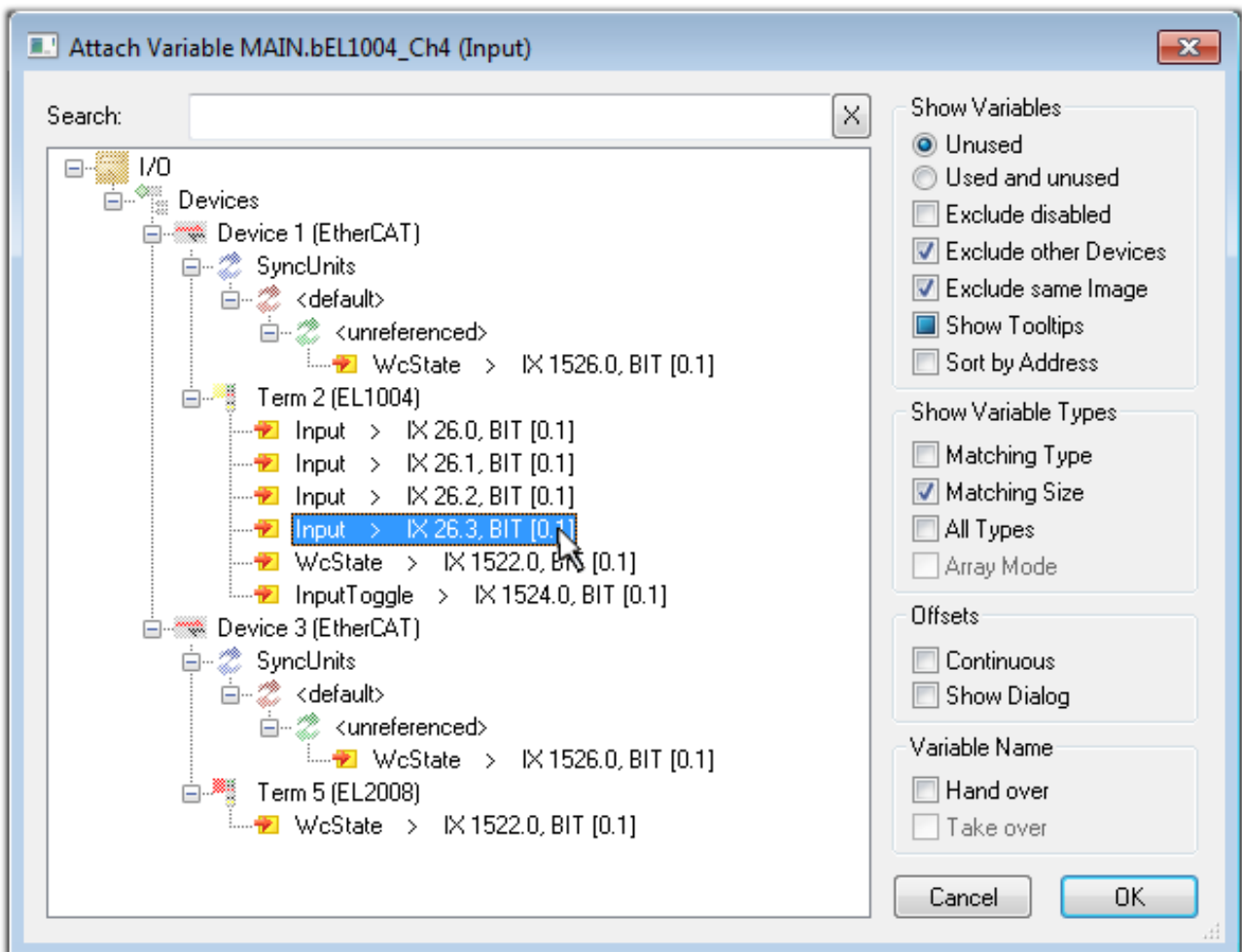


Fig. 81: Selecting PDO of type BOOL

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:

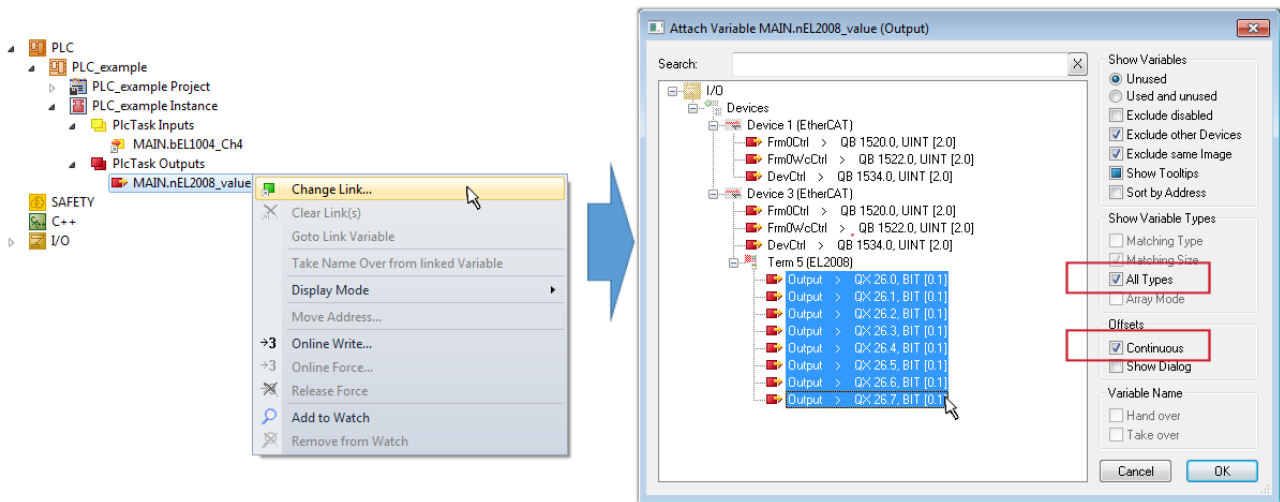



Fig. 82: Selecting several PDOs simultaneously: activate "Continuous" and "All types"

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:

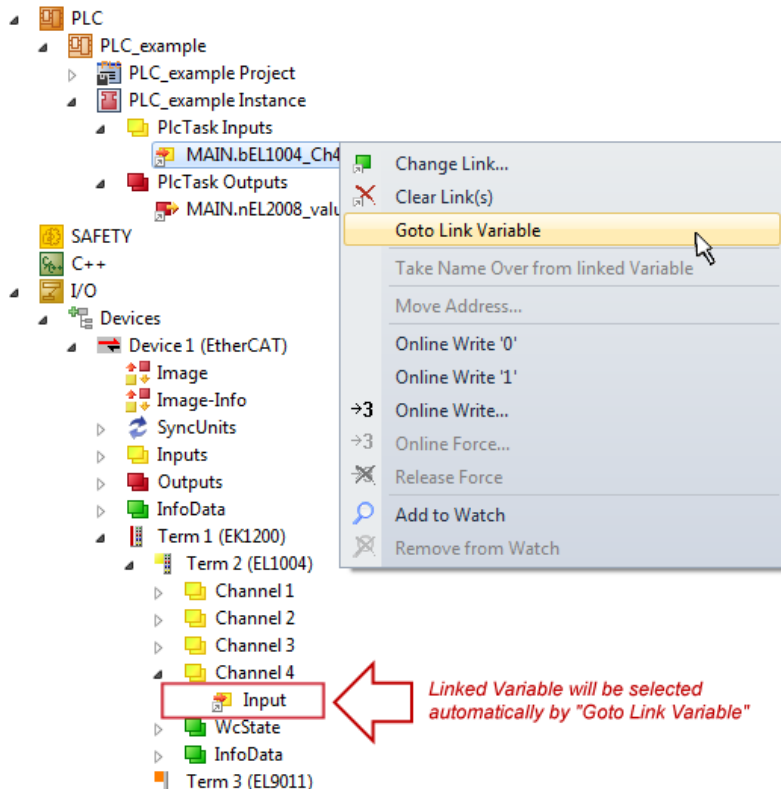


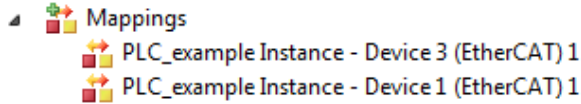
Fig. 83: Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample

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:

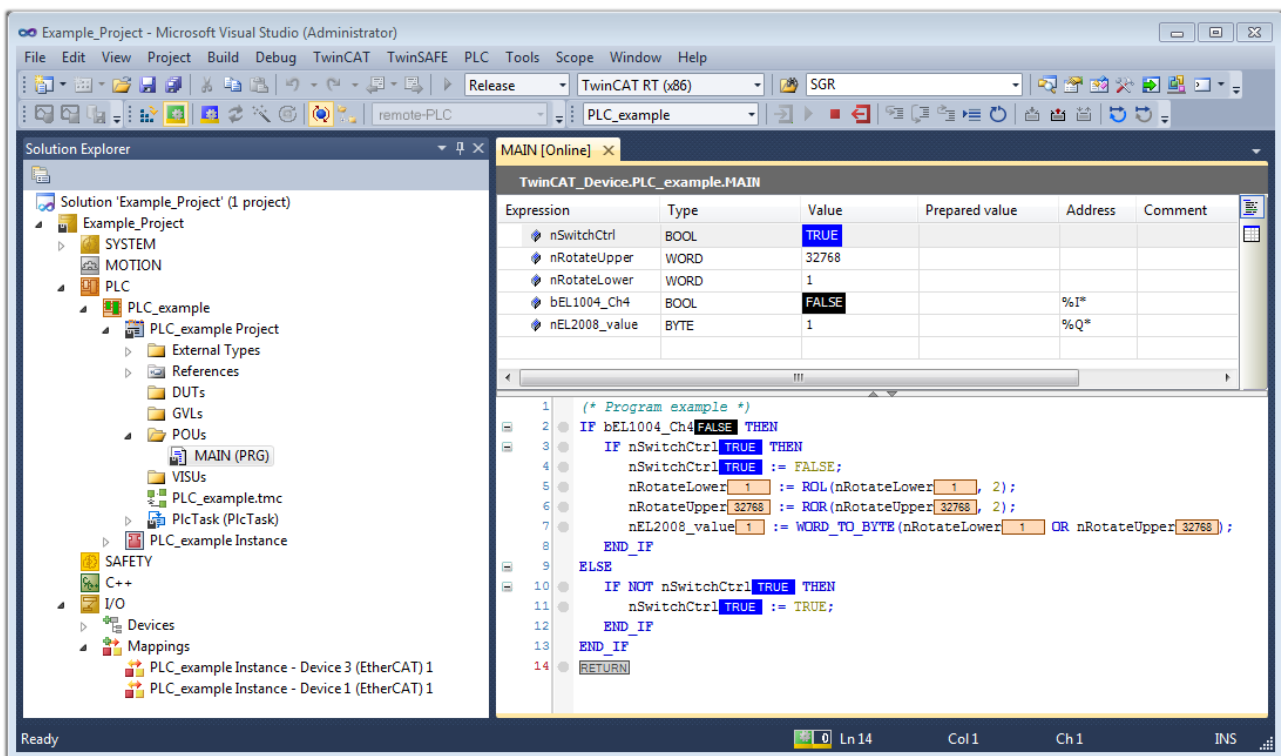


Fig. 84: TwinCAT development environment (VS shell): logged-in, after program startup

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

5.2 TwinCAT Development Environment

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

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

Details:

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

Additional features:

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

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

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

5.2.1 Installation of the TwinCAT real-time driver

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

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

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

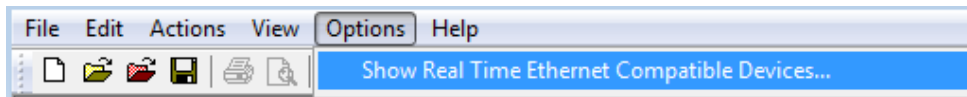


Fig. 85: System Manager "Options" (TwinCAT 2)

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

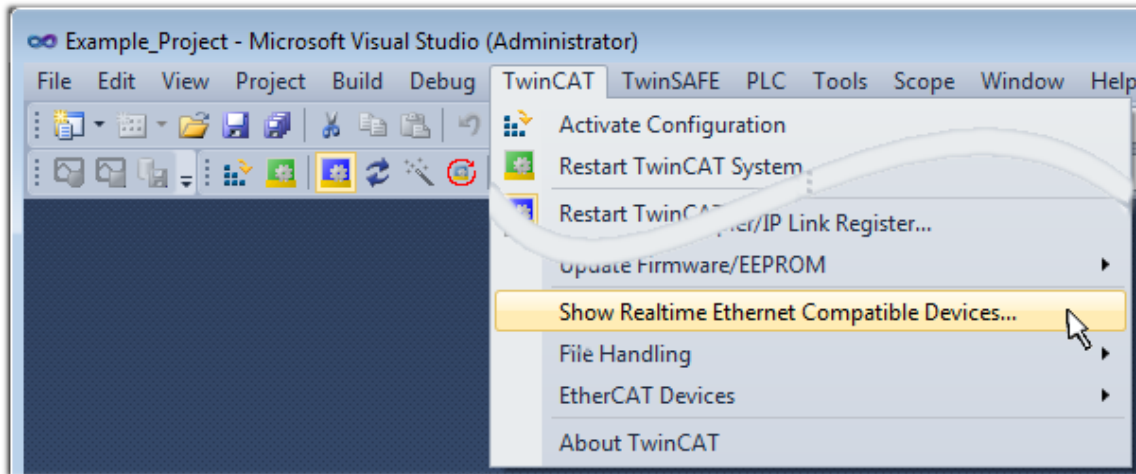


Fig. 86: Call up under VS Shell (TwinCAT 3)

The following dialog appears:

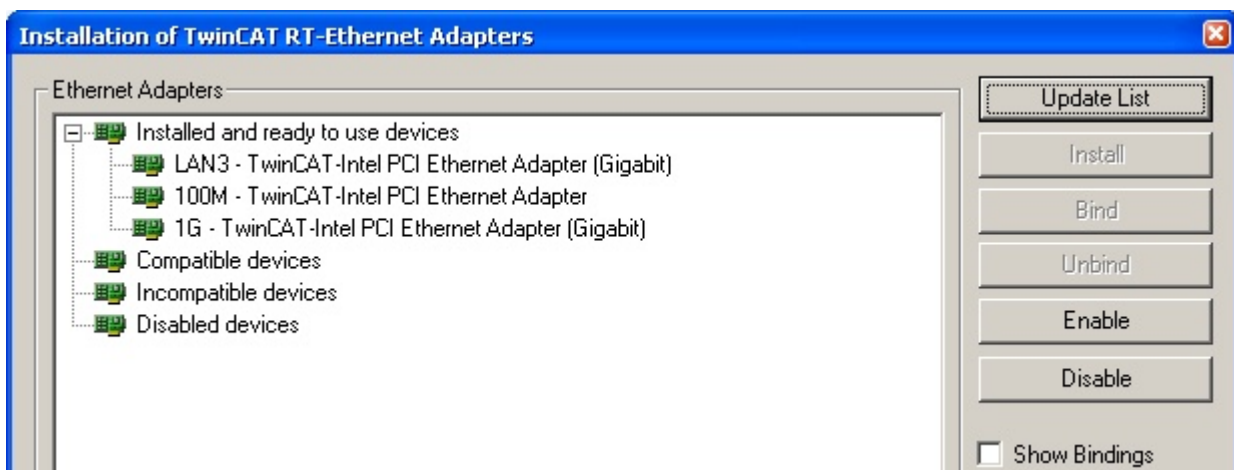


Fig. 87: Overview of network interfaces

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

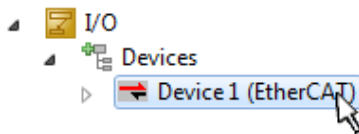
A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter [Offline configuration creation, section "Creating the EtherCAT device" \[▶ 108\]](#) in order to view the compatible ethernet ports via its EtherCAT properties (tab „Adapter“, button „Compatible Devices...“):



Fig. 88: 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)

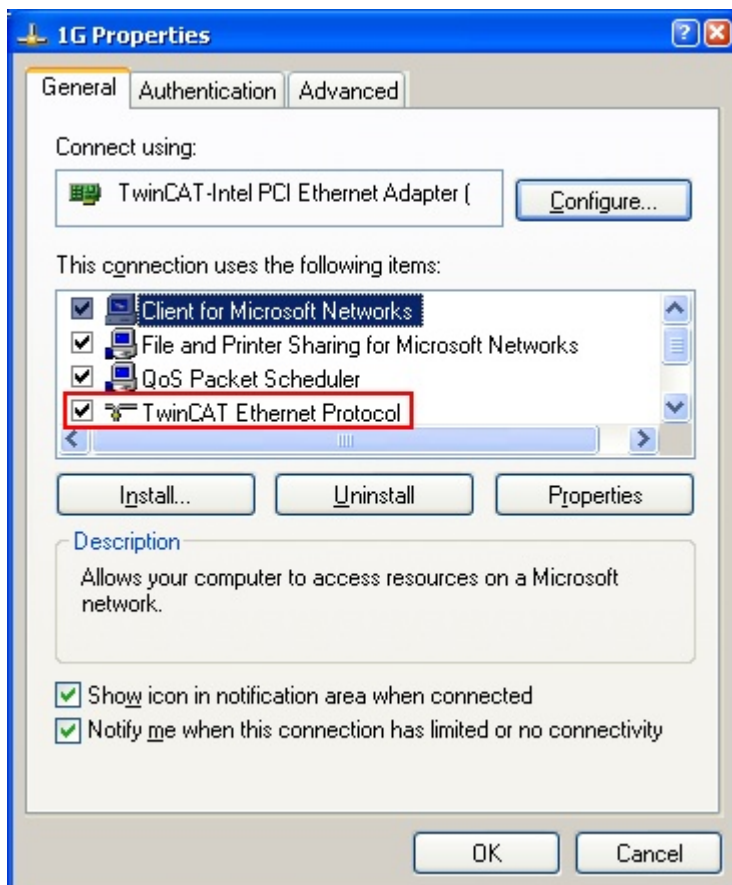


Fig. 89: Windows properties of the network interface

A correct setting of the driver could be:

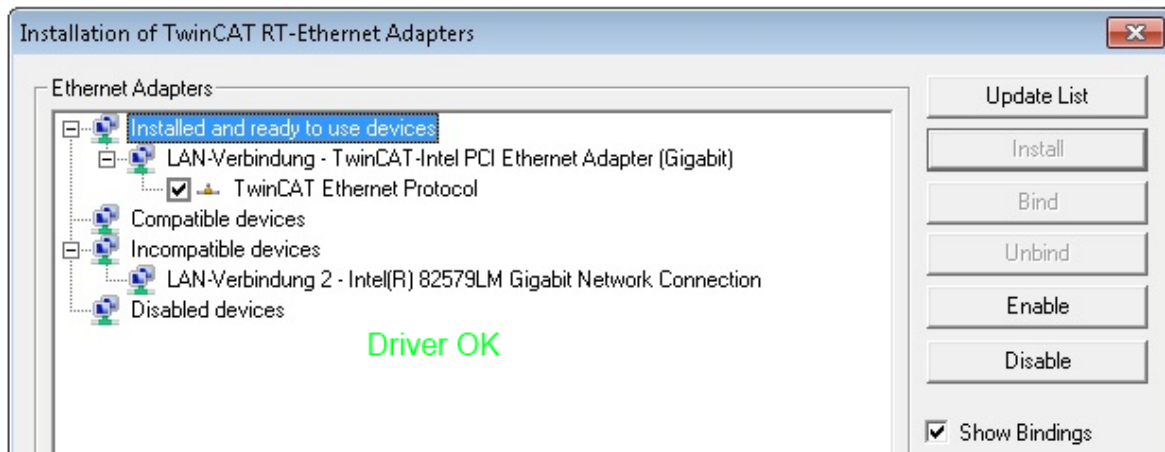


Fig. 90: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

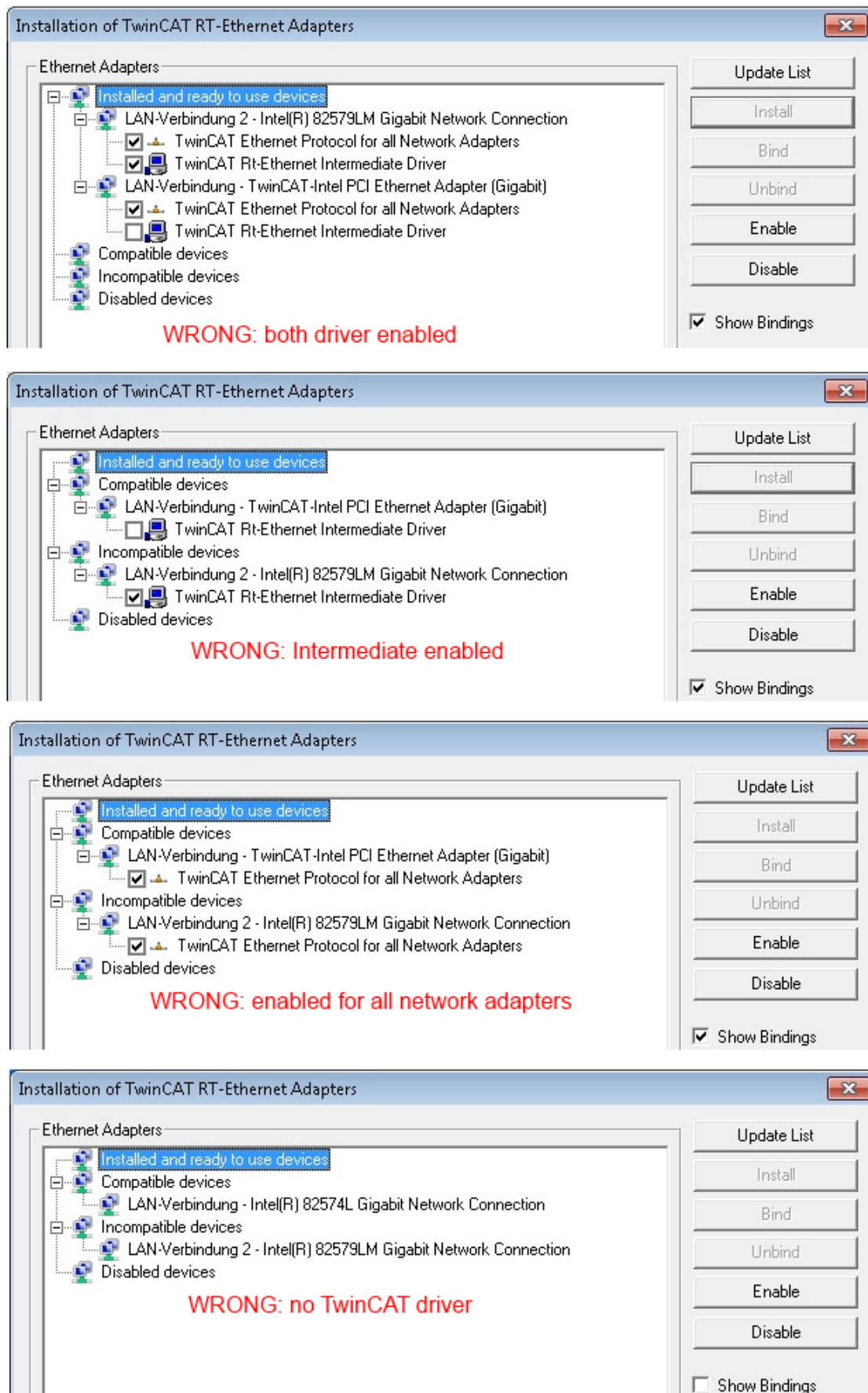


Fig. 91: Incorrect driver settings for the Ethernet port

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.

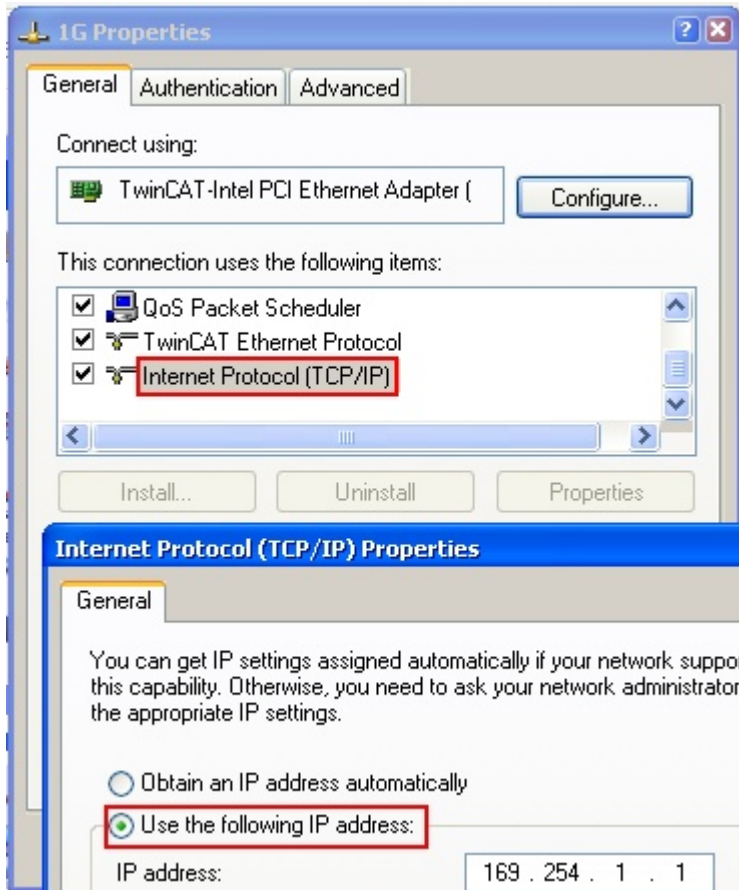


Fig. 92: TCP/IP setting for the Ethernet port

5.2.2 Notes regarding ESI device description

Installation of the latest ESI device description

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

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

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

Default settings:

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

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

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

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

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

The [TwinCAT ESI Updater](#) [► 107] 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”

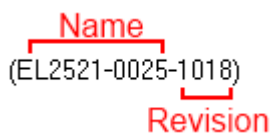


Fig. 93: Identifier structure

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](#) [► 10].

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.

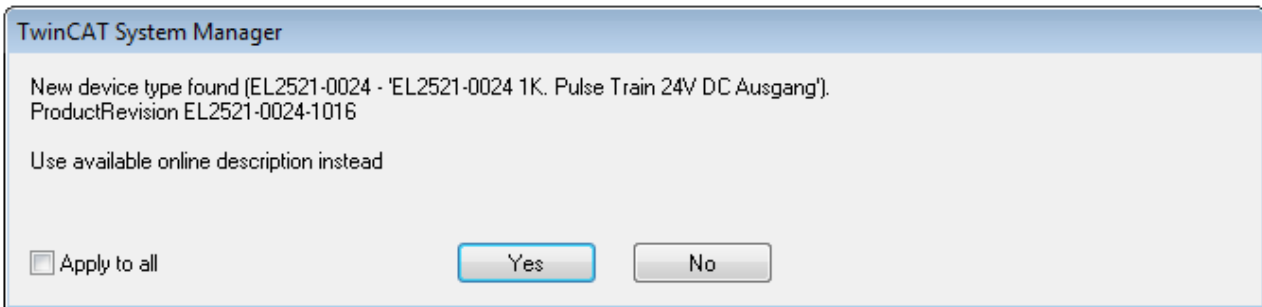


Fig. 94: *OnlineDescription information window (TwinCAT 2)*

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

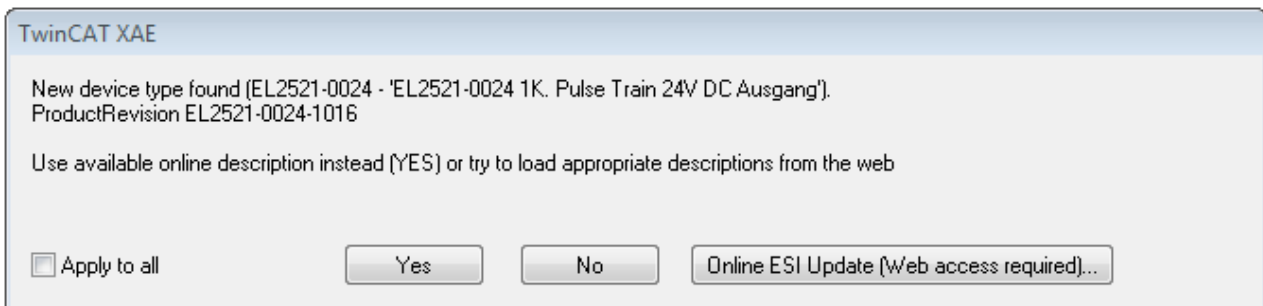


Fig. 95: *Information window OnlineDescription (TwinCAT 3)*

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](#)' [[▶ 108](#)].

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. 96: 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").

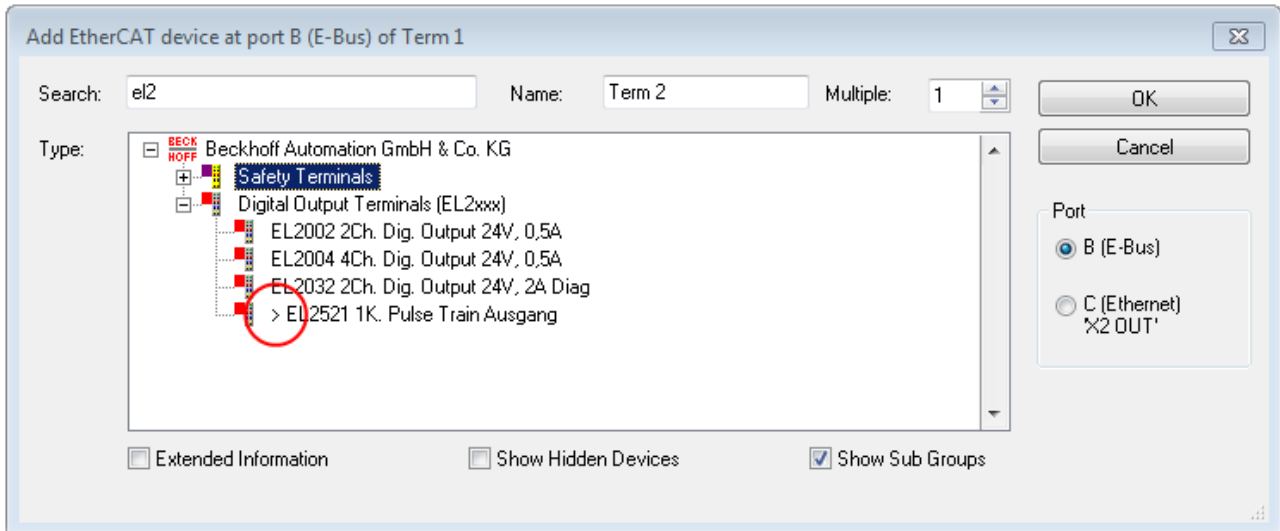


Fig. 97: Indication of an online recorded ESI of EL2521 as an example

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.

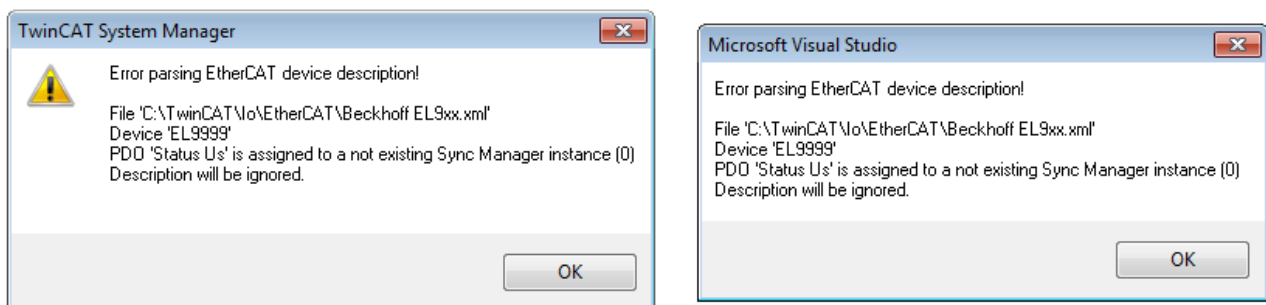


Fig. 98: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

Reasons may include:

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

5.2.3 TwinCAT ESI Updater

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

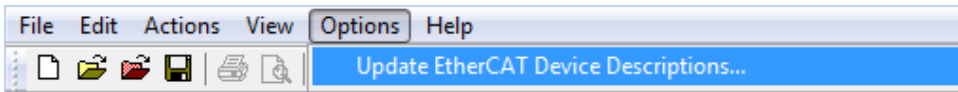


Fig. 99: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:

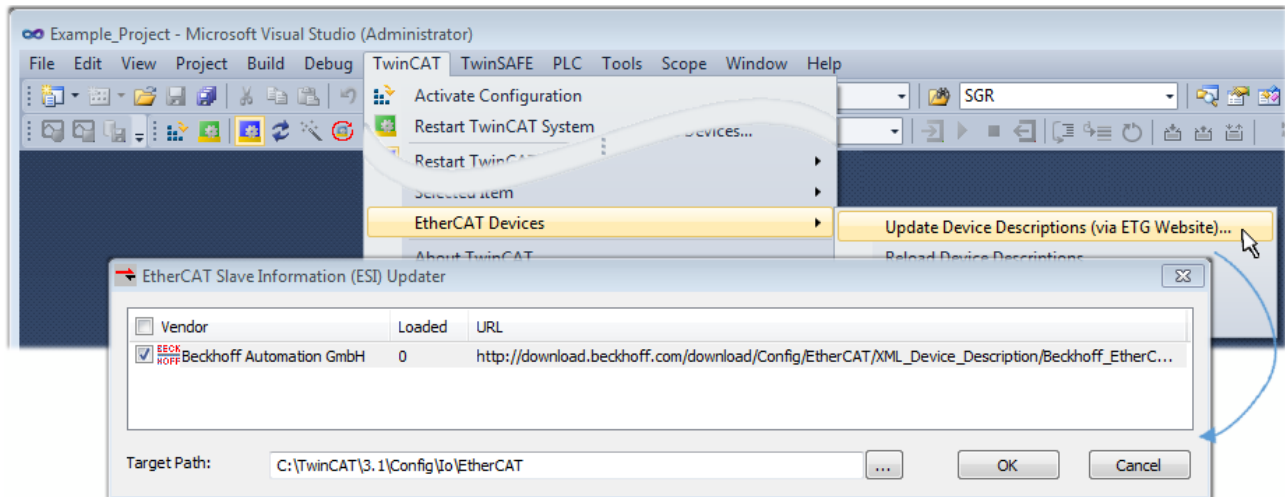


Fig. 100: Using the ESI Updater (TwinCAT 3)

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

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

5.2.4 Distinction between Online and Offline

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

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

In any case, during each startup the EtherCAT master checks whether the slaves it finds match the configuration. This test can be parameterised in the extended slave settings. Refer to [note “Installation of the latest ESI-XML device description” \[► 103\]](#).

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 [▶ 113] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 114]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 117]

The scan with existing configuration [▶ 118] can also be carried out for comparison.

5.2.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

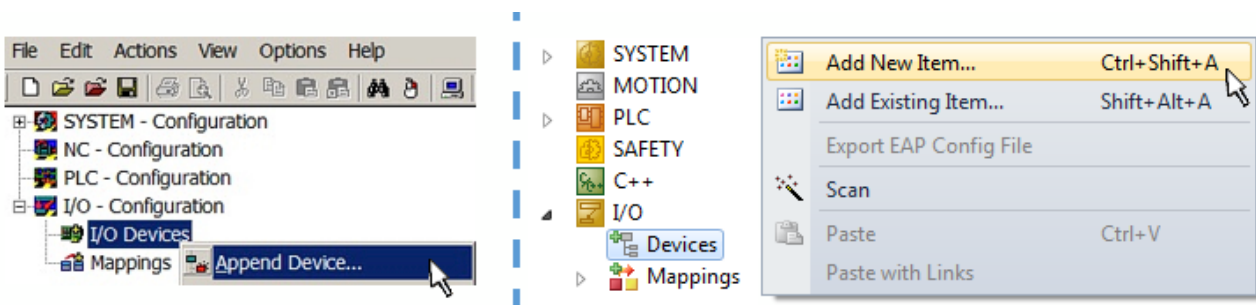


Fig. 101: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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".

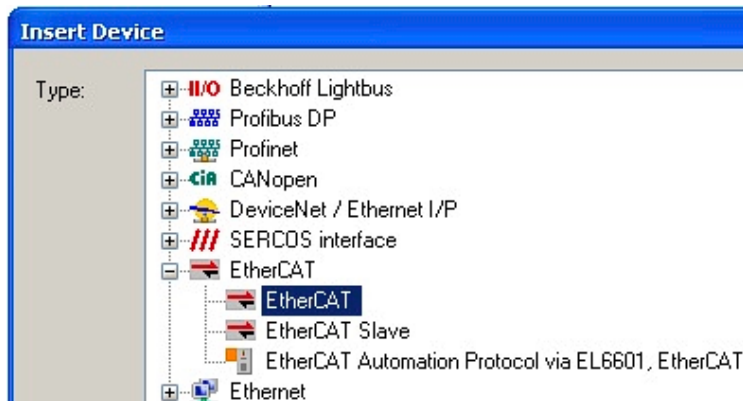


Fig. 102: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

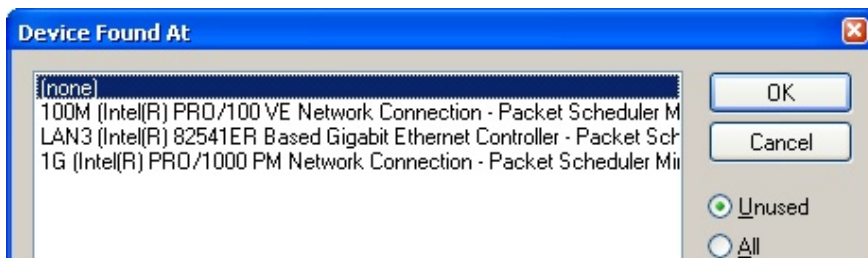


Fig. 103: Selecting the Ethernet port

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)”.

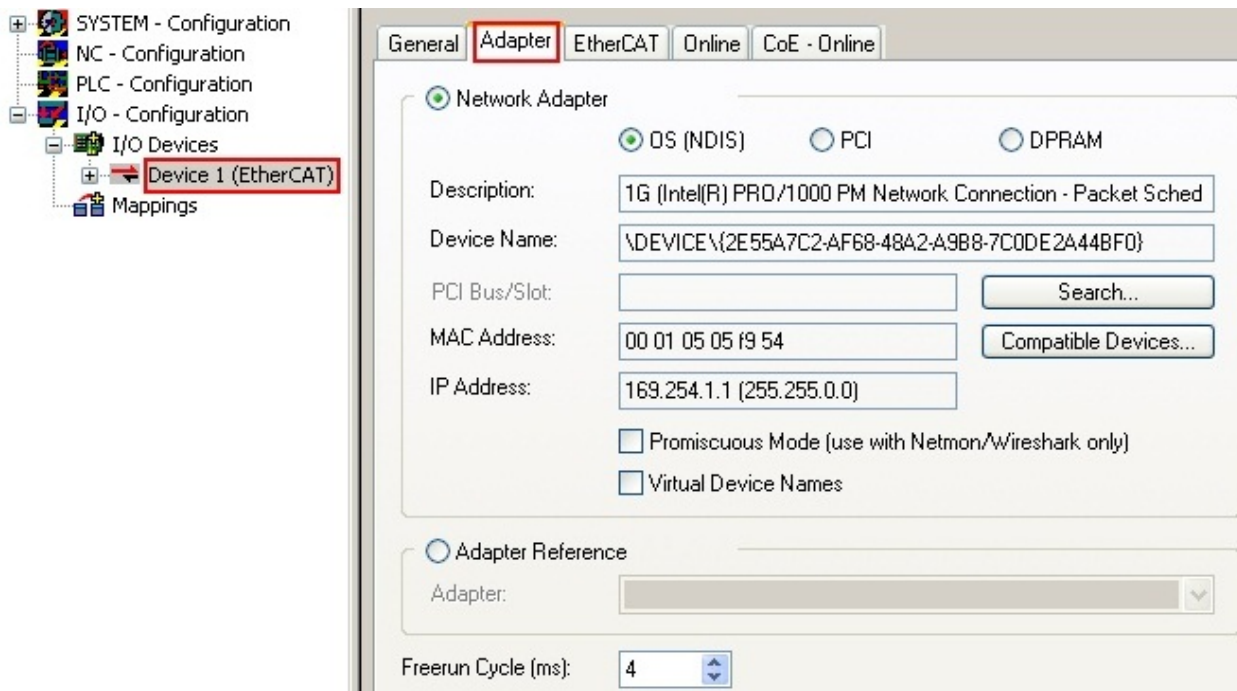
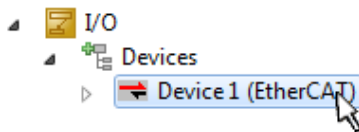


Fig. 104: EtherCAT device properties (TwinCAT 2)

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 \[▶ 97\]](#).

Defining EtherCAT slaves

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

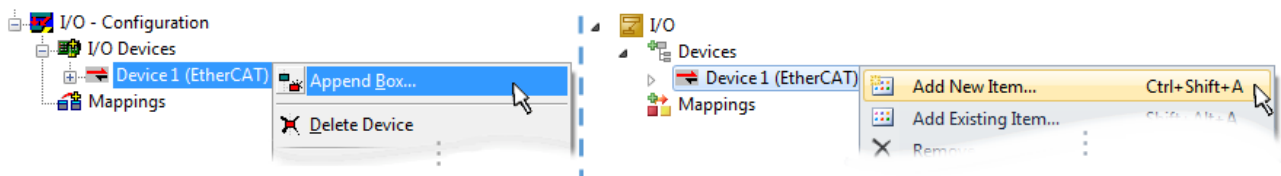


Fig. 105: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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).

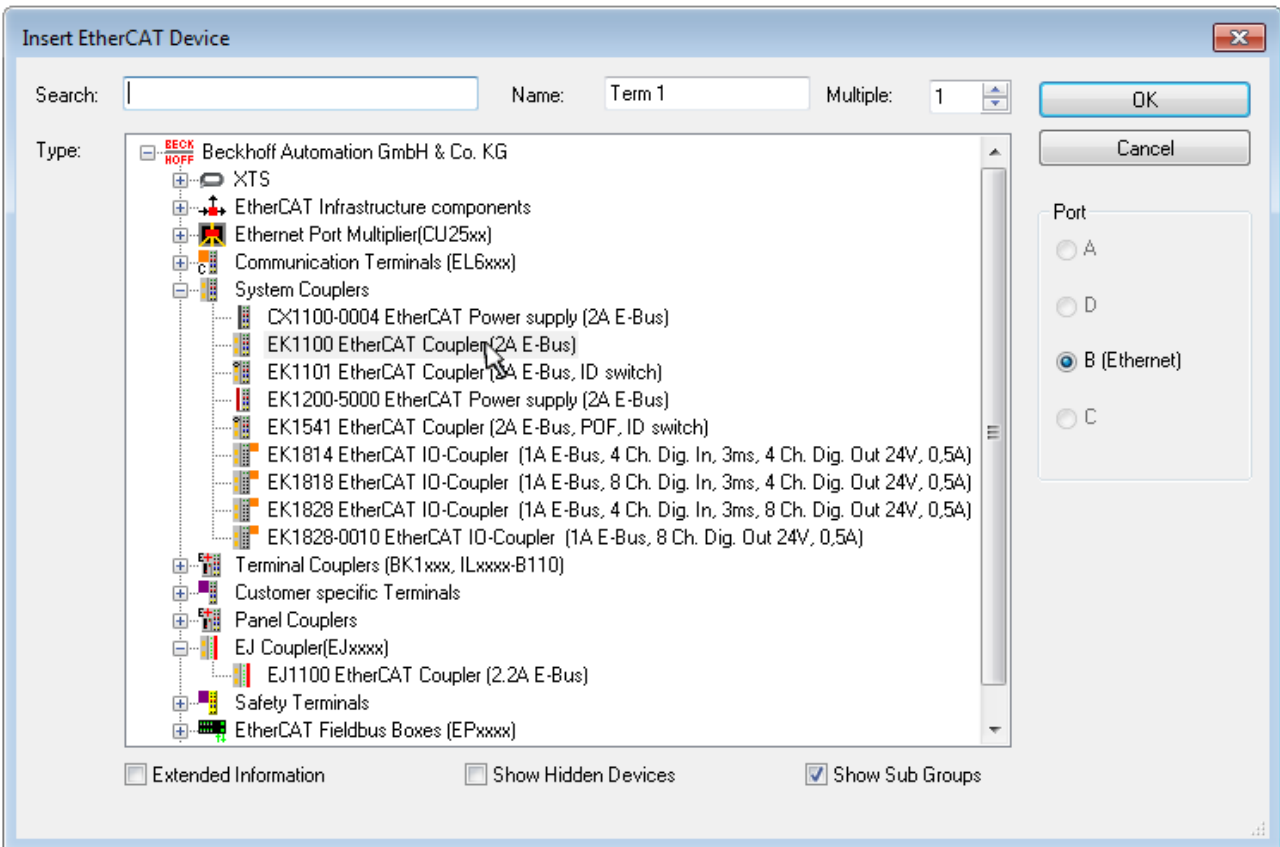


Fig. 106: Selection dialog for new EtherCAT device

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”.

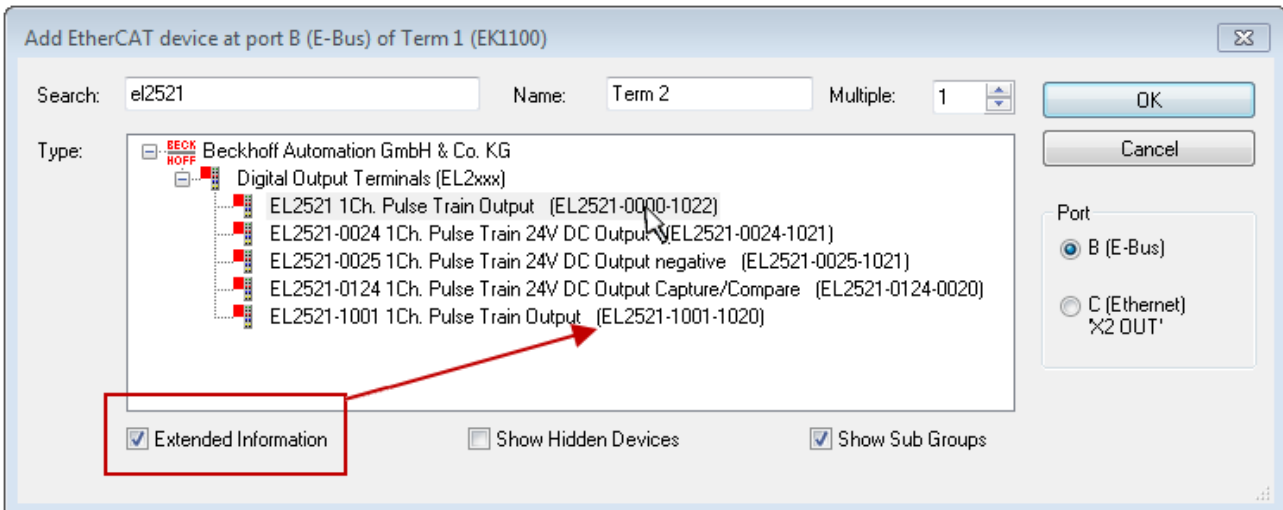


Fig. 107: Display of device revision

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”.

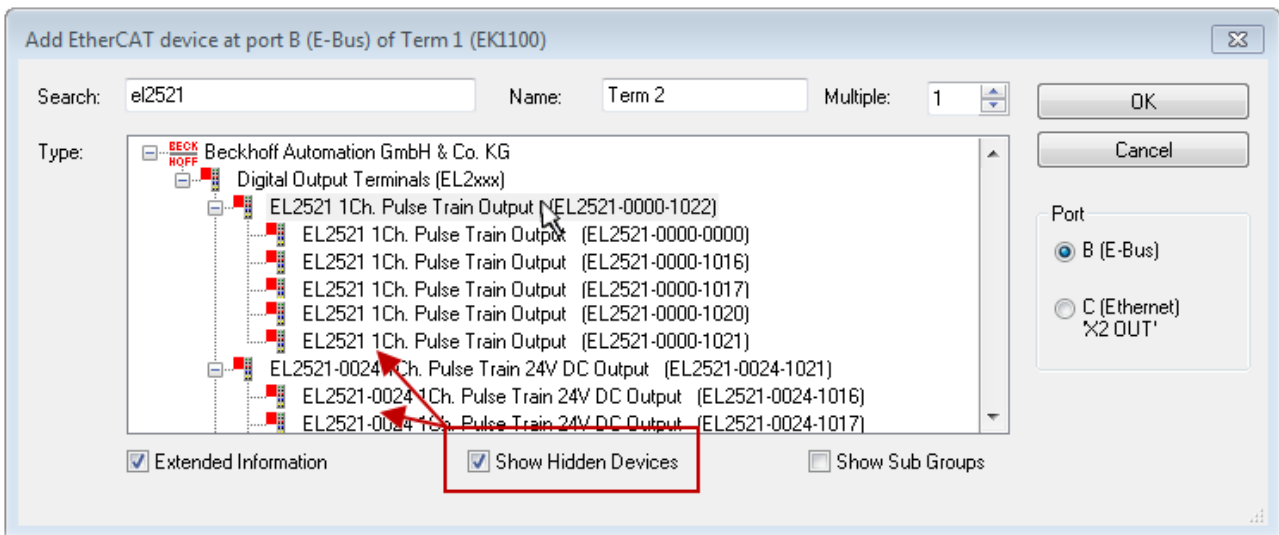


Fig. 108: Display of previous revisions

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.

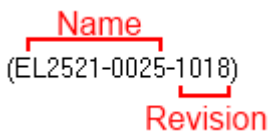


Fig. 109: Name/revision of the terminal

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, ...

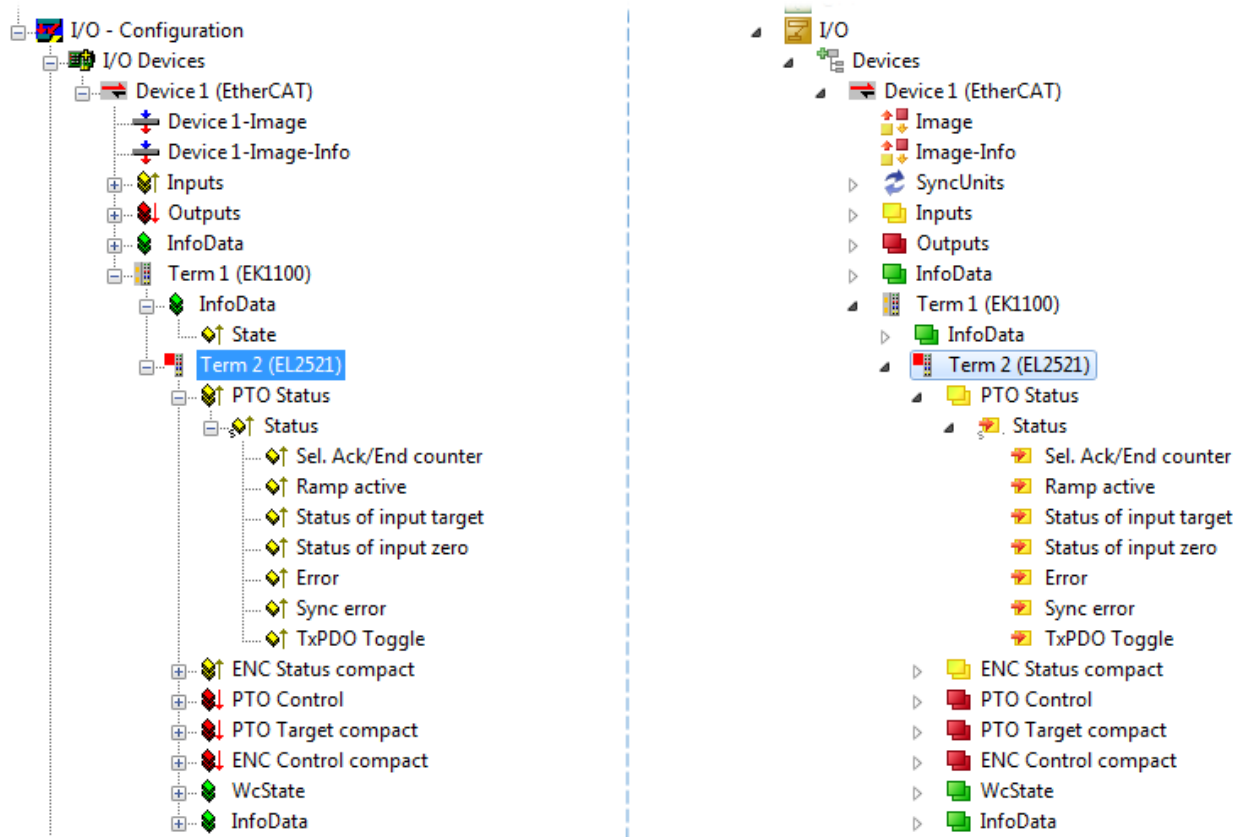




Fig. 110: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)



5.2.6 ONLINE configuration creation

Detecting/scanning of the EtherCAT device

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



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

TwinCAT can be set into this mode:

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

i Online scanning in Config mode

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

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

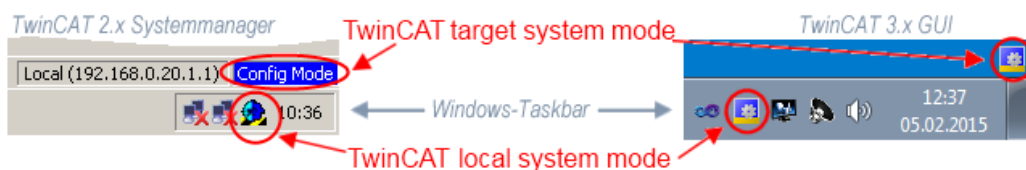


Fig. 111: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

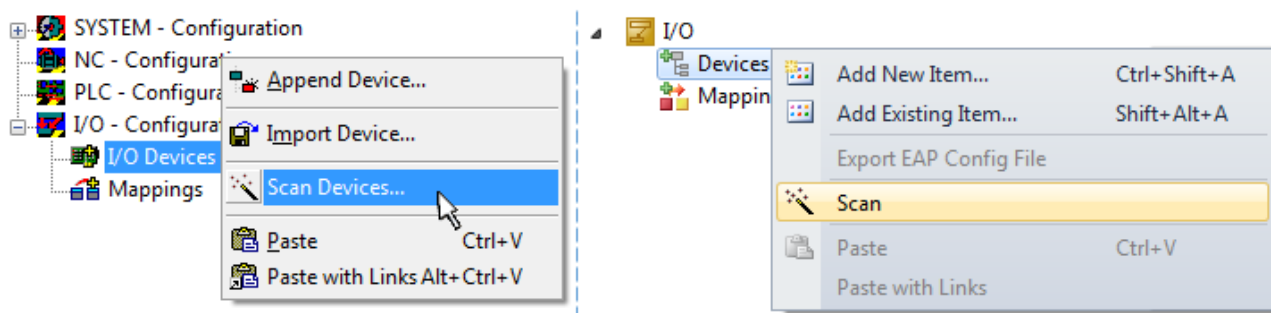


Fig. 112: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.

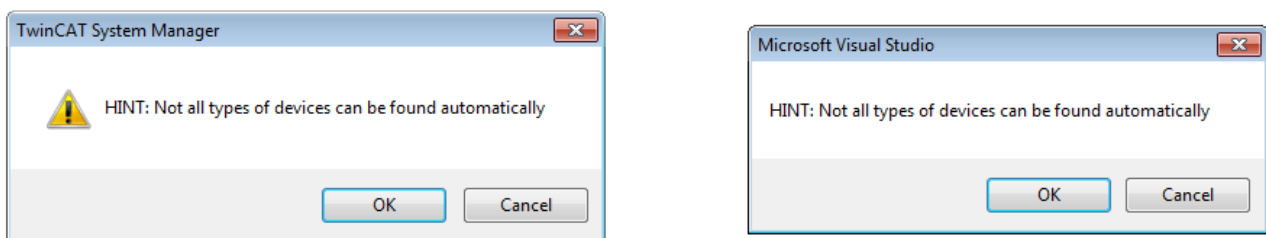


Fig. 113: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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” .

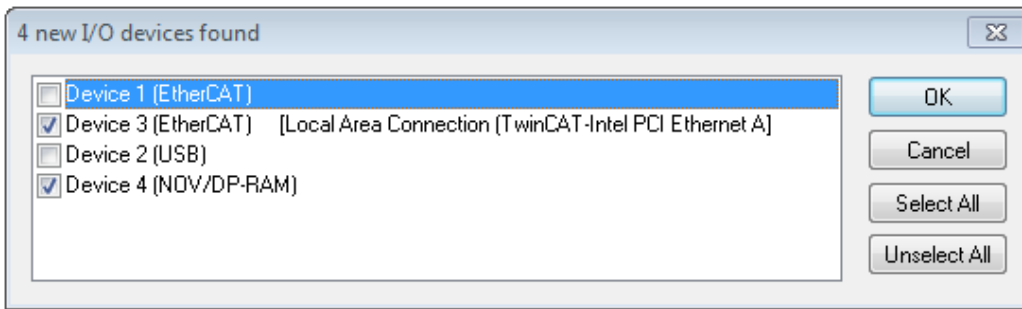


Fig. 114: *Detected Ethernet devices*

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](#) [▶ 97].

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. 115: *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](#) [▶ 118] 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:

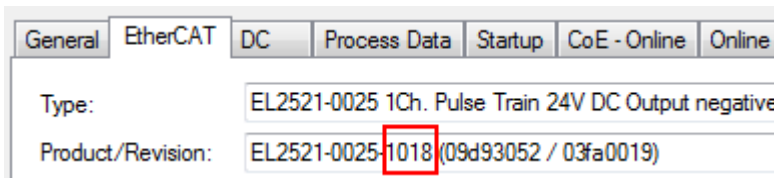


Fig. 116: Installing EthetCAT terminal with revision -1018

Likewise, during the prototype test phase, the functions and properties of this terminal are tested by the programmers/commissioning engineers and used if necessary, i.e. addressed from the PLC 'B.pro' or the NC. (the same applies correspondingly to the TwinCAT 3 solution files).

The prototype development is now completed and series production of machine B starts, for which Beckhoff continues to supply the EL2521-0025-0018. If the commissioning engineers of the series machine production department always carry out a scan, a B configuration with the identical contents results again for each machine. Likewise, A might create spare parts stores worldwide for the coming series-produced machines with EL2521-0025-1018 terminals.

After some time Beckhoff extends the EL2521-0025 by a new feature C. Therefore the FW is changed, outwardly recognizable by a higher FW version and a **new revision -1019**. Nevertheless the new device naturally supports functions and interfaces of the predecessor version(s); an adaptation of 'B.tsm' or even 'B.pro' is therefore unnecessary. The series-produced machines can continue to be built with 'B.tsm' and 'B.pro'; it makes sense to perform a comparative scan [▶ 118] 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:

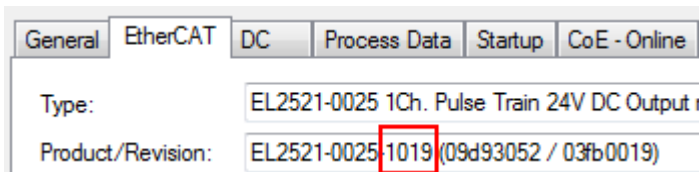


Fig. 117: Detection of EtherCAT terminal with revision -1019

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

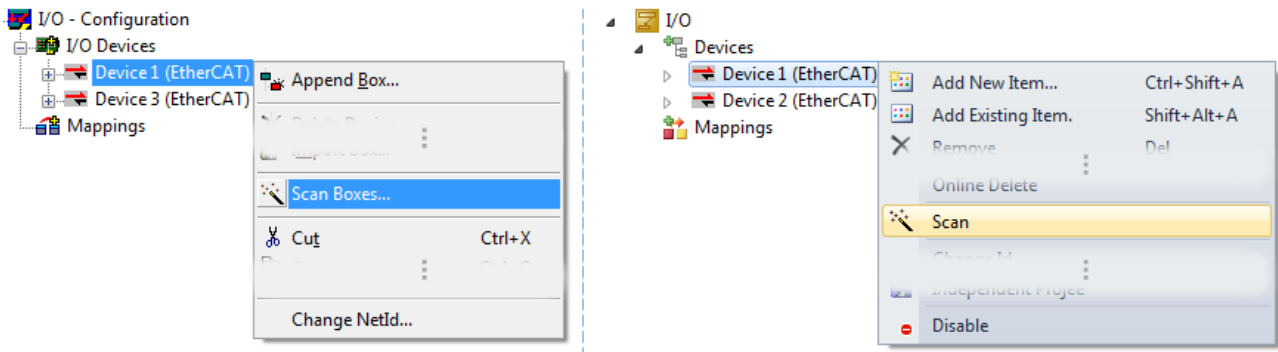


Fig. 119: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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. 120: Scan progress exemplary by TwinCAT 2

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



Fig. 121: 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. 122: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar



Fig. 123: 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”.

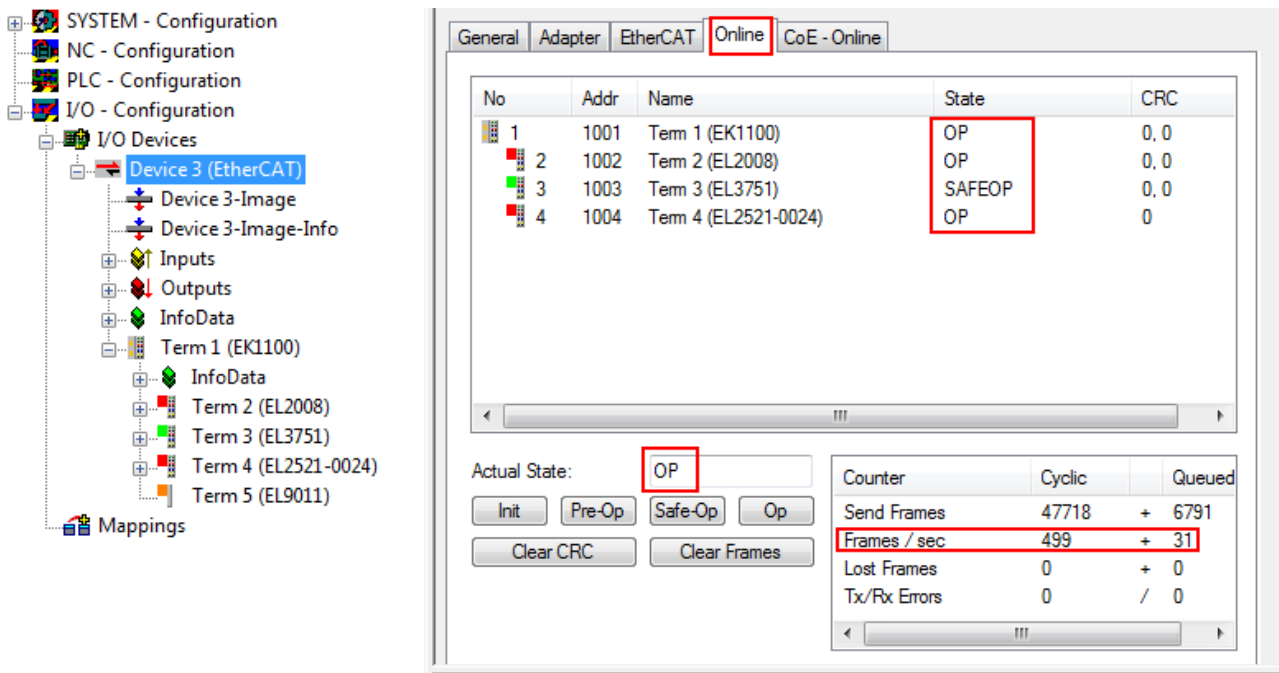


Fig. 124: Online display example

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 \[► 108\]](#).

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.

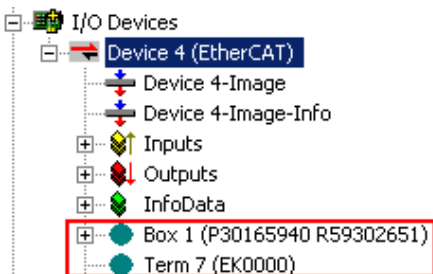


Fig. 125: Faulty identification

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. 126: 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.

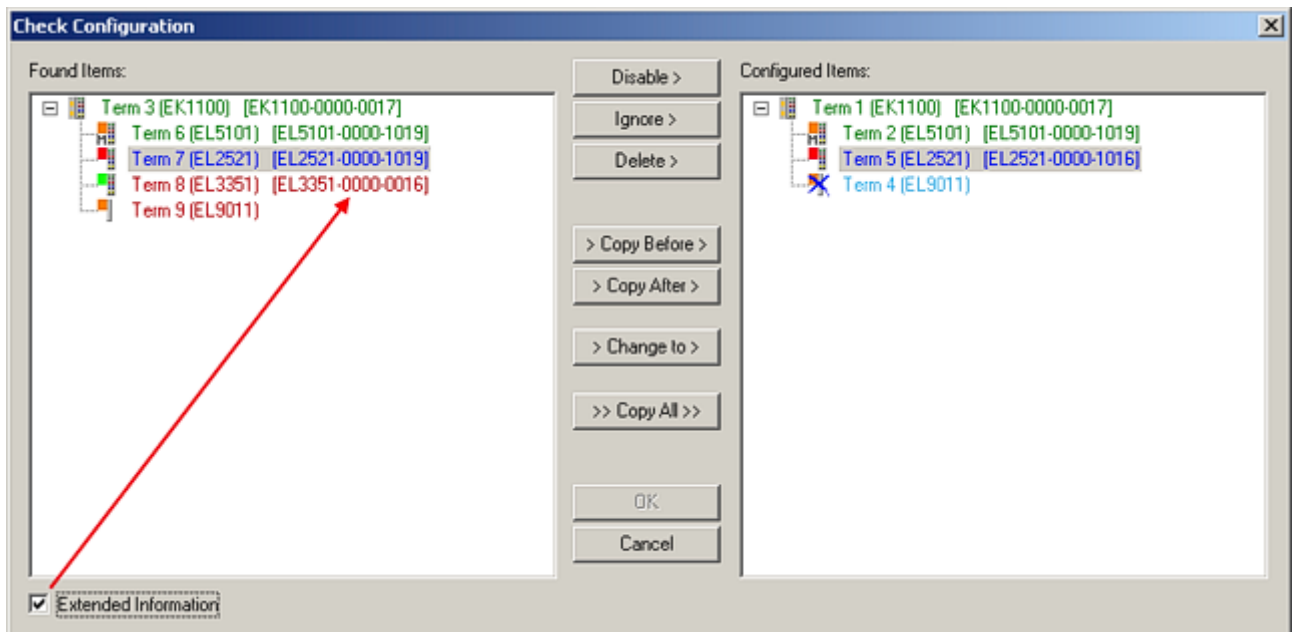


Fig. 127: Correction dialog

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)

Colour	Explanation
red	<ul style="list-style-type: none"> This EtherCAT slave is not present on the other side. It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

i Device selection based on revision, compatibility

The ESI description also defines the process image, the communication type between master and slave/device and the device functions, if applicable. The physical device (firmware, if available) has to support the communication queries/settings of the master. This is backward compatible, i.e. newer devices (higher revision) should be supported if the EtherCAT master addresses them as an older revision. The following compatibility rule of thumb is to be assumed for Beckhoff EtherCAT Terminals/ Boxes/ EJ-modules:

device revision in the system >= 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.

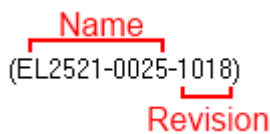


Fig. 128: Name/revision of the terminal

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, ...

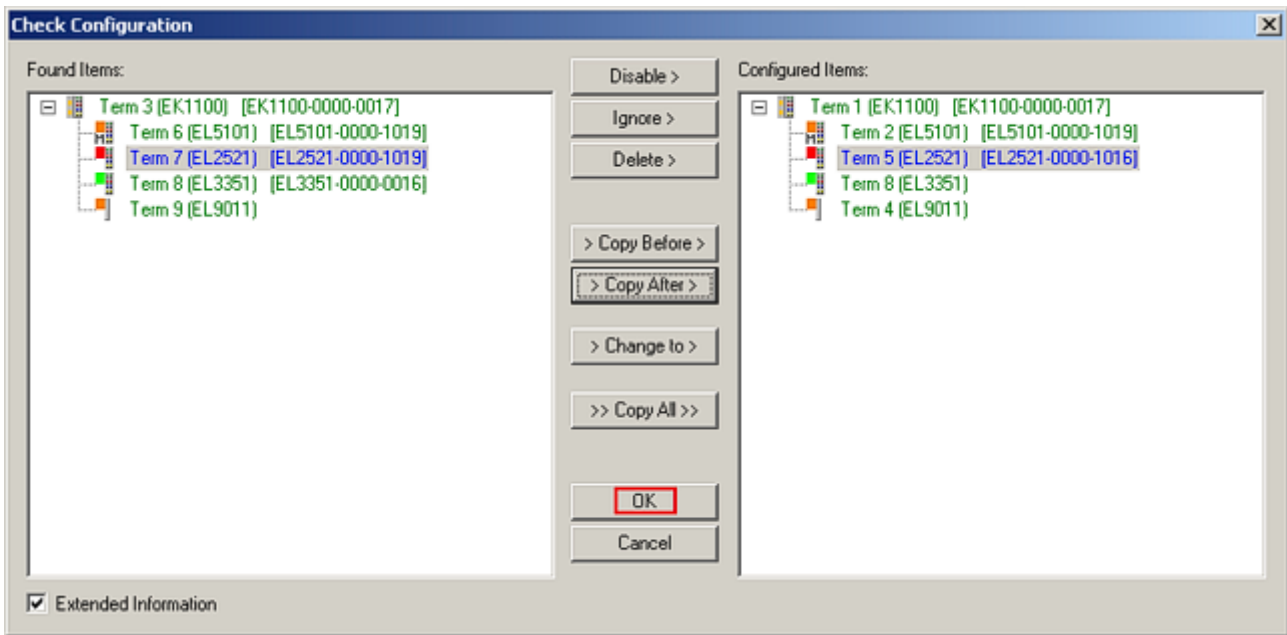


Fig. 129: Correction dialog with modifications

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.

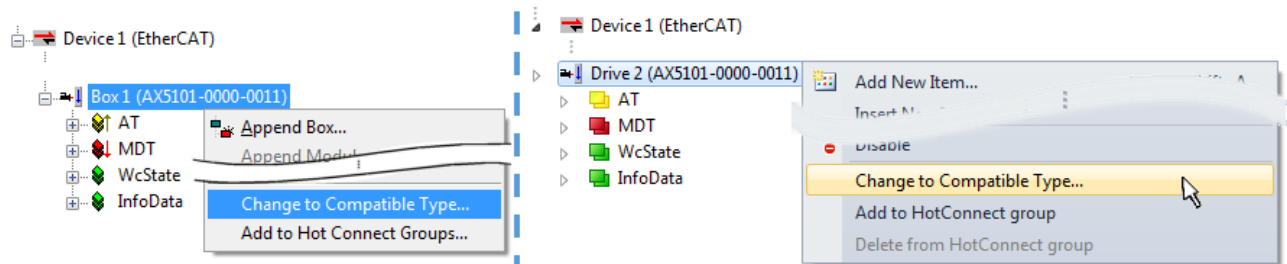


Fig. 130: Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3)

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*

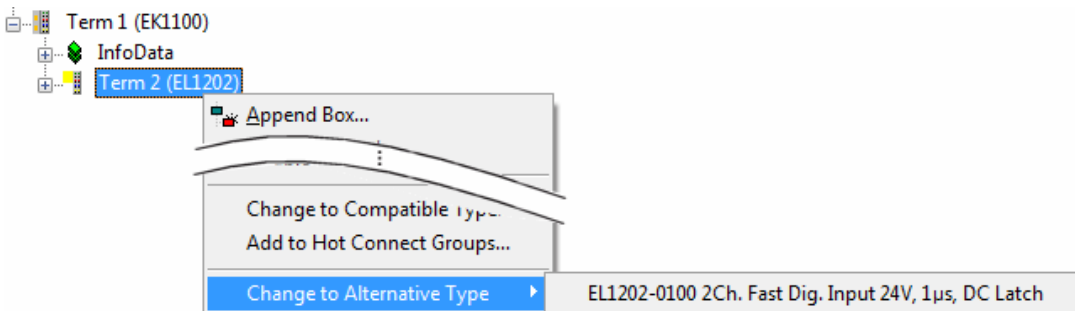


Fig. 131: TwinCAT 2 Dialog Change to Alternative Type

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

5.2.7 EtherCAT subscriber configuration

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

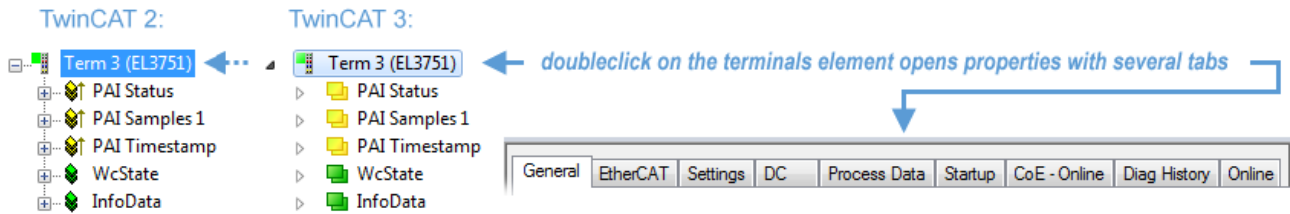


Fig. 132: Branch element as terminal EL3751

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

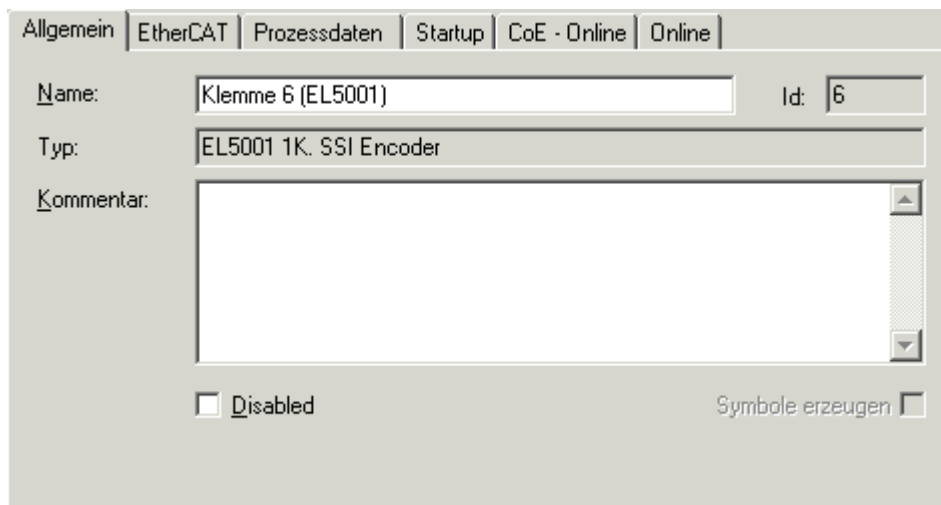


Fig. 133: "General" tab

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. 134: „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.

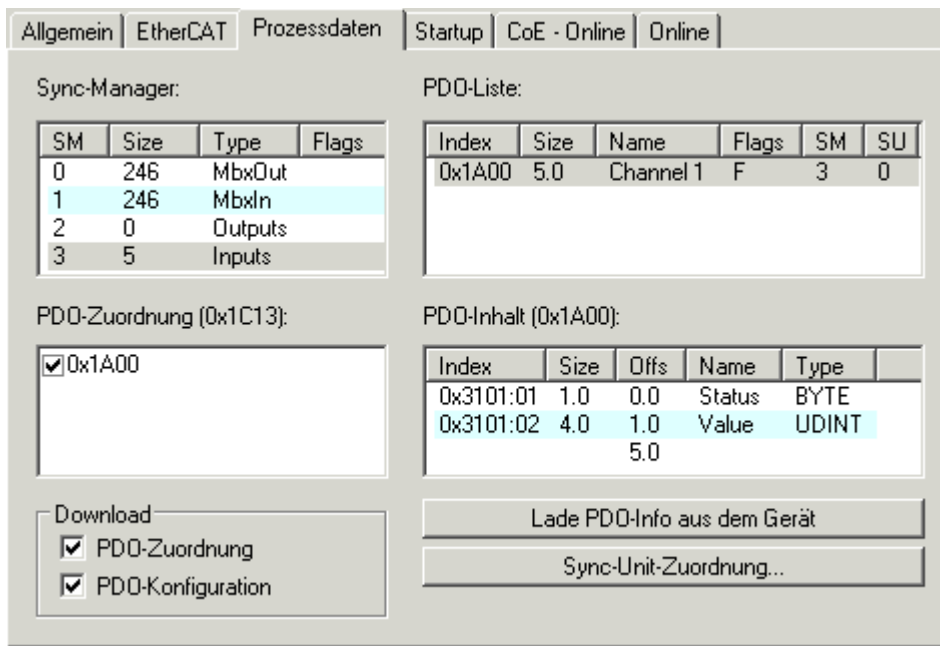


Fig. 135: "Process Data" tab

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").

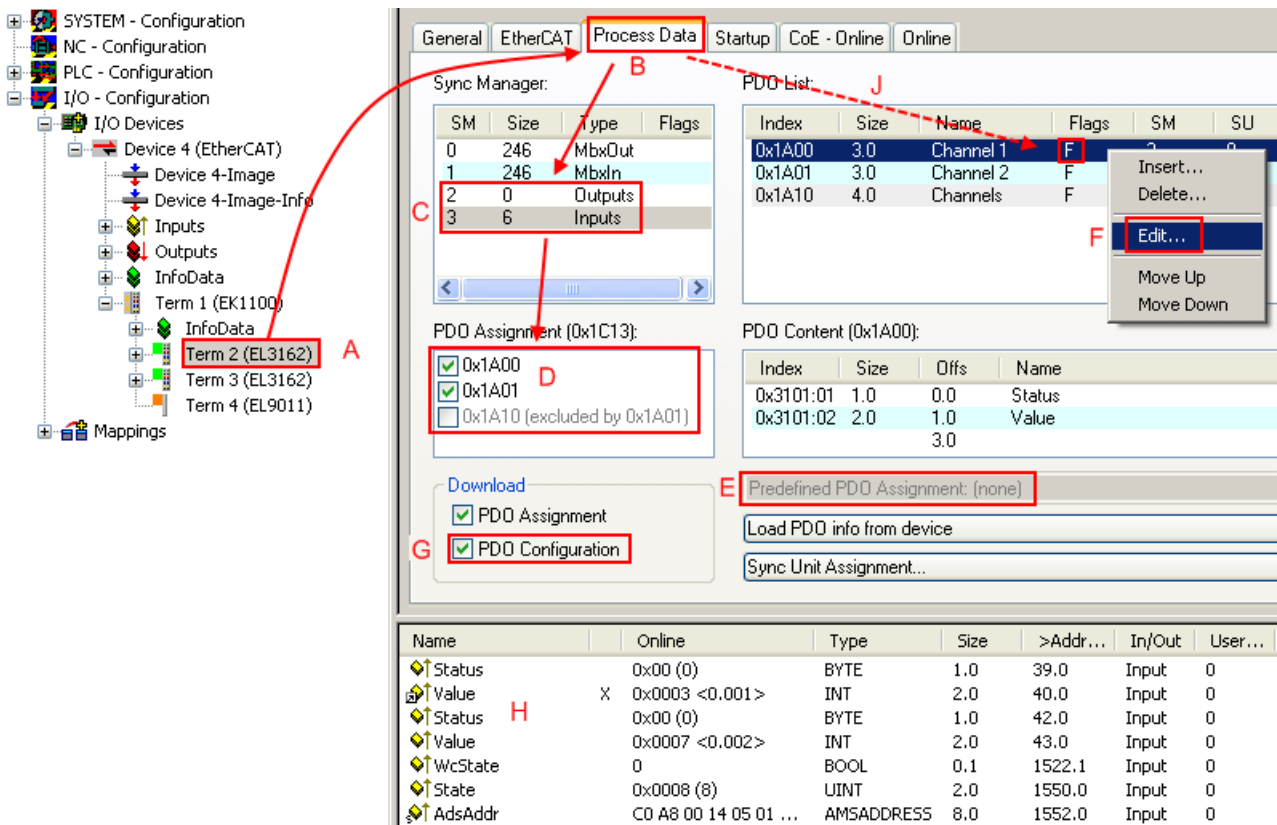


Fig. 136: Configuring the process data

Manual modification of the process data

According to the ESI description, a PDO can be identified as “fixed” with the flag “F” in the PDO overview (Fig. “Configuring the process data”, J). The configuration of such PDOs cannot be changed, even if TwinCAT offers the associated dialog (“Edit”). In particular, CoE content cannot be displayed as cyclic process data. This generally also applies in cases where a device supports download of the PDO configuration, “G”. In case of incorrect configuration the EtherCAT slave usually refuses to start and change to OP state. The System Manager displays an “invalid SM cfg” 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 [► 129] 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.

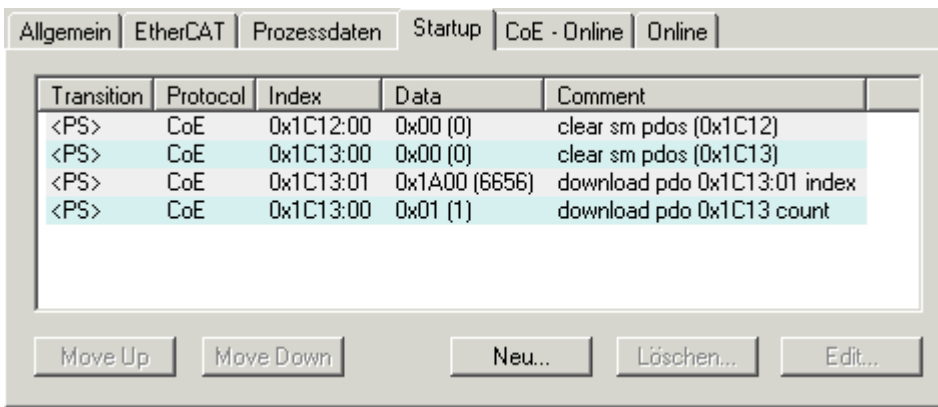


Fig. 137: „Startup“ tab

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.

Index	Name	Flags	Wert
1000	Device type	RO	0x00000000 (0)
1008	Device name	RO	EL5001-0000
1009	Hardware version	RO	V00.01
100A	Software version	RO	V00.07
1011:0	Restore default parame...	RW	> 1 <
1011:01	Restore all	RW	0
1018:0	Identity object	RO	> 4 <
1018:01	Vendor id	RO	0x00000002 (2)
1018:02	Product code	RO	0x13893052 (327757906)
1018:03	Revision number	RO	0x00000000 (0)
1018:04	Serial number	RO	0x00000001 (1)
1A00:0	TxPDO 001 mapping	RO	> 2 <
1A00:01	Subindex 001	RO	0x3101:01, 8
1A00:02	Subindex 002	RO	0x3101:02, 32
1C00:0	SM type	RO	> 4 <
1C00:01	Subindex 001	RO	0x01 (1)
1C00:02	Subindex 002	RO	0x02 (2)
1C00:03	Subindex 003	RO	0x03 (3)
1C00:04	Subindex 004	RO	0x04 (4)
1C13:0	SM 3 PDO assign (inputs)	RW	> 1 <
1C13:01	Subindex 001	RW	0x1A00 (6656)
3101:0	Inputs	RO P	> 2 <
3101:01	Status	RO P	0x41 (65)
3101:02	Value	RO P	0x00000000 (0)
4061:0	Feature bits	RW	> 4 <
4061:01	disable frame error	RW	FALSE
4061:02	enable power failure Bit	RW	FALSE
4061:03	enable inhibit time	RW	FALSE
4061:04	enable test mode	RW	FALSE
4066	SSI-coding	RW	Gray code (1)
4067	SSI-baudrate	RW	500 kBaud (3)
4068	SSI-frame type	RW	Multiturn 25 bit (0)
4069	SSI-frame size	RW	0x0019 (25)
406A	Data length	RW	0x0018 (24)
406B	Min. inhibit time[μs]	RW	0x0000 (0)

Fig. 138: "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 <i>Update list</i> 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 <i>Advanced</i> button opens the <i>Advanced Settings</i> dialog. Here you can specify which objects are displayed in the list.

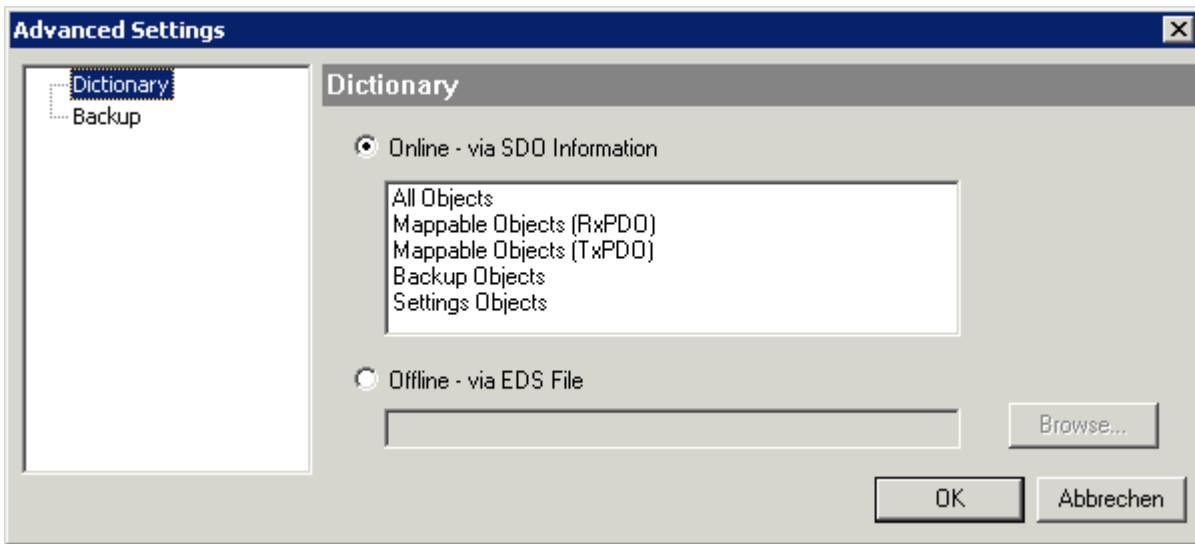


Fig. 139: Dialog “Advanced settings”

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

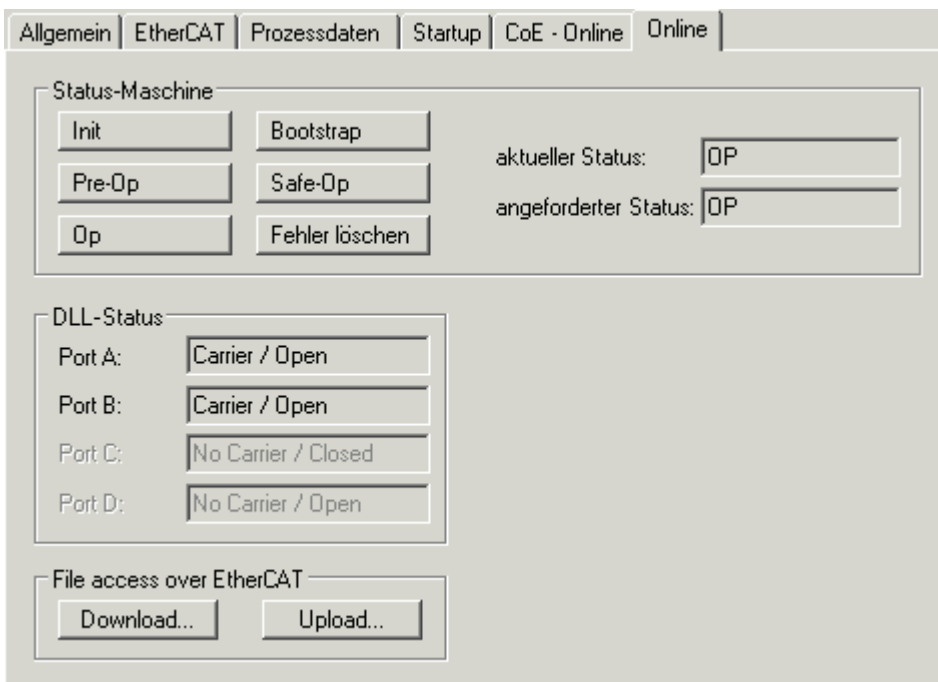


Fig. 140: „Online“ tab

State Machine

Init	This button attempts to set the EtherCAT device to the <i>Init</i> state.
Pre-Op	This button attempts to set the EtherCAT device to the <i>pre-operational</i> state.
Op	This button attempts to set the EtherCAT device to the <i>operational</i> state.
Bootstrap	This button attempts to set the EtherCAT device to the <i>Bootstrap</i> state.
Safe-Op	This button attempts to set the EtherCAT device to the <i>safe-operational</i> 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 <i>Clear Error</i> 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)

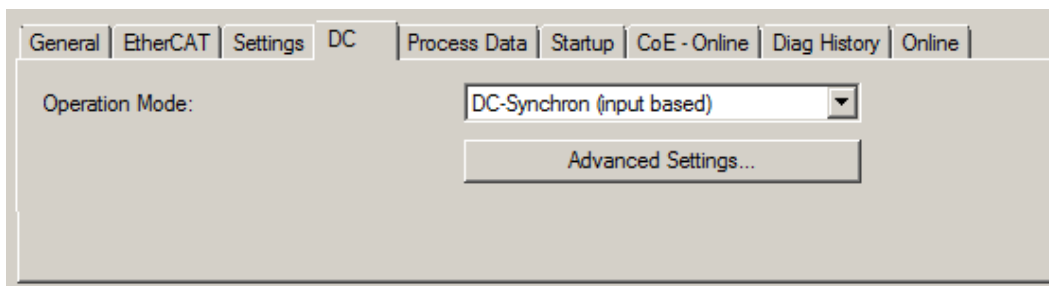


Fig. 141: "DC" tab (Distributed Clocks)

Operation Mode	Options (optional): <ul style="list-style-type: none"> • FreeRun • SM-Synchron • DC-Synchron (Input based) • DC-Synchron
Advanced Settings...	Advanced settings for readjustment of the real time determinant TwinCAT-clock

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

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

5.2.7.1 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](#) |▶ 127|),
 - 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 [▶ 124] tab.

PDO Configuration

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

5.3 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.

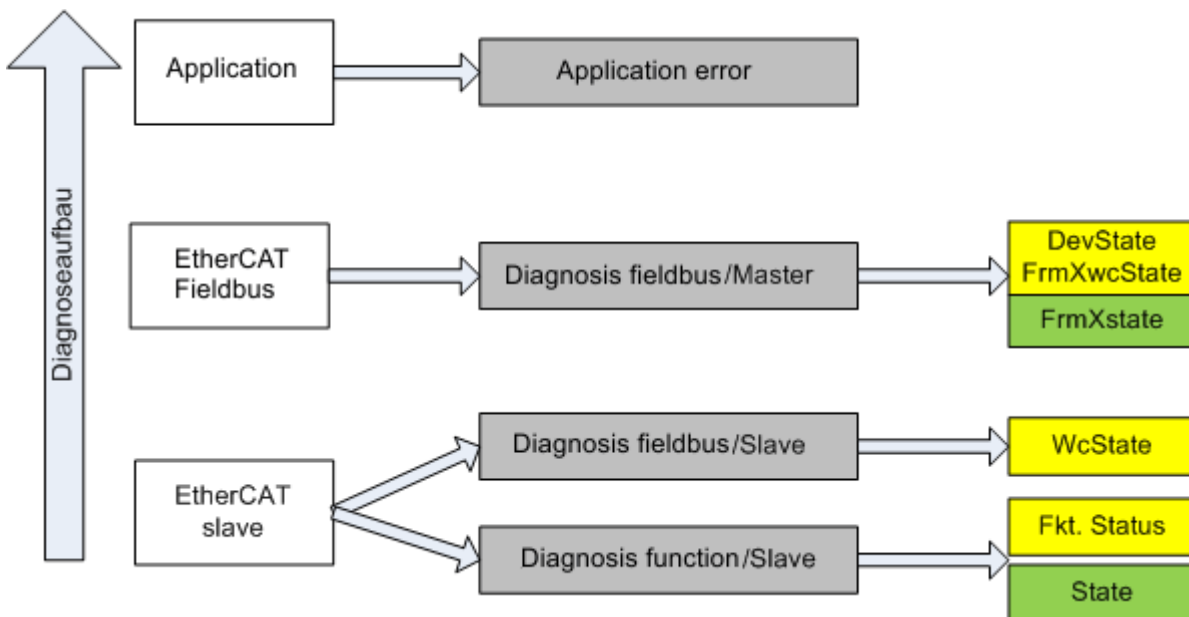


Fig. 142: Selection of the diagnostic information of an EtherCAT Slave

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.

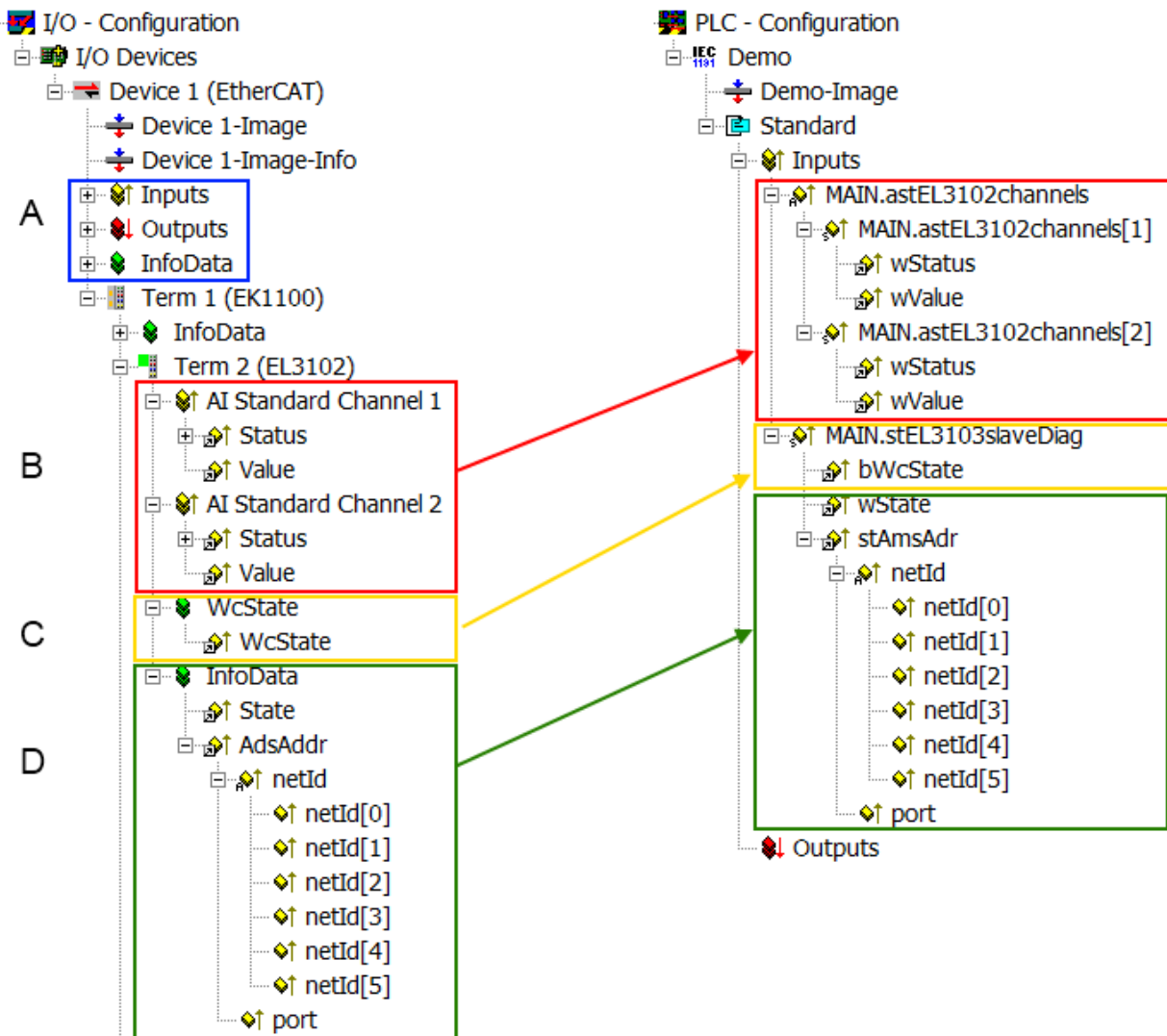


Fig. 143: Basic EtherCAT Slave Diagnosis in the PLC

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".

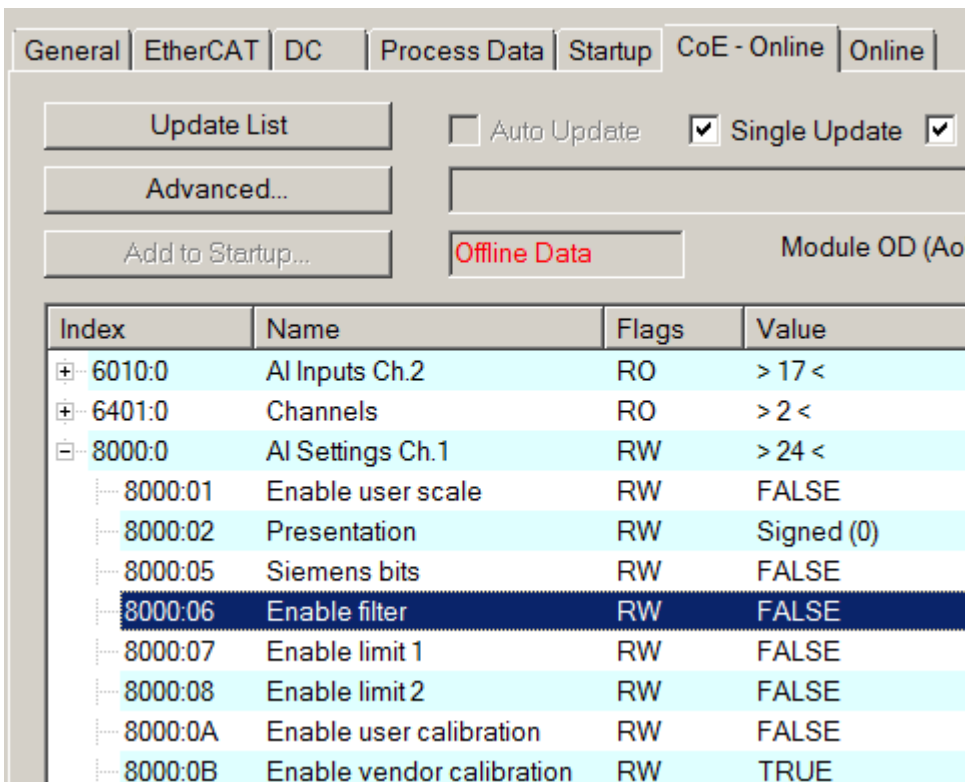


Fig. 144: EL3102, CoE directory

● 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.

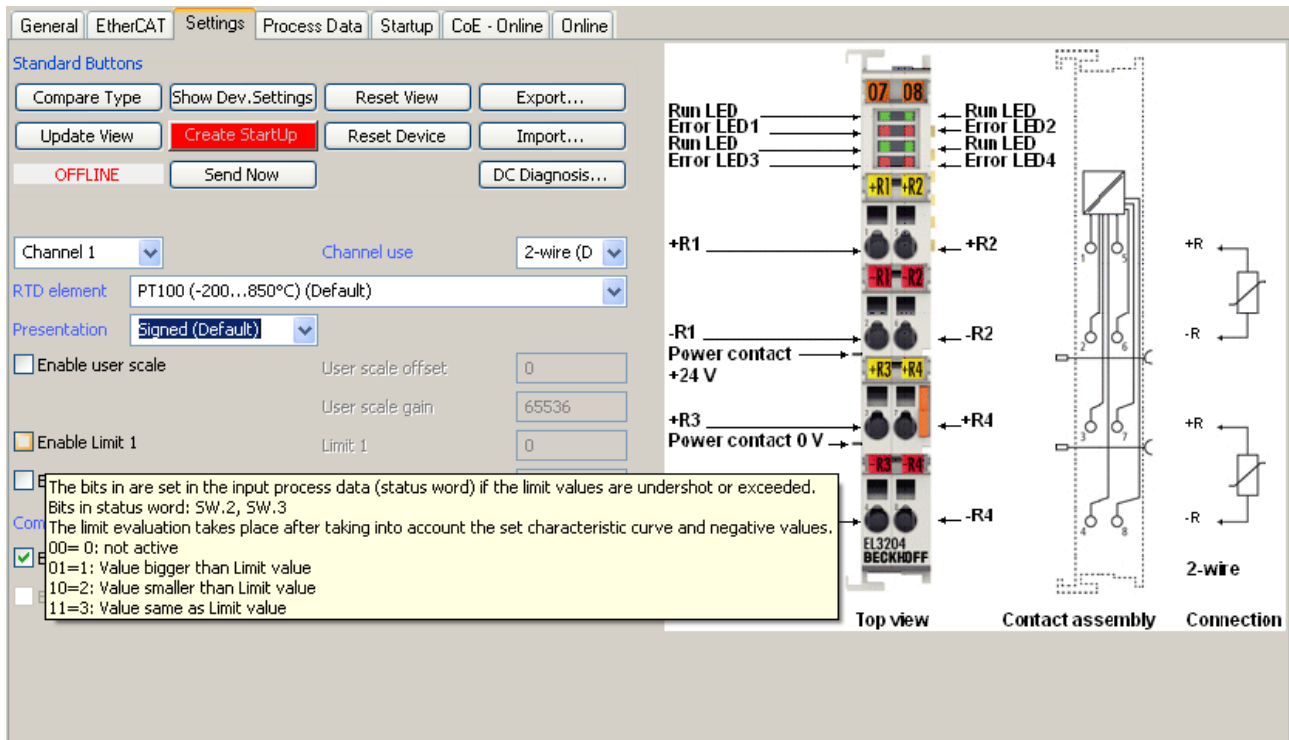


Fig. 145: Example of commissioning aid for a EL3204

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 \[► 36\]](#)" 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.

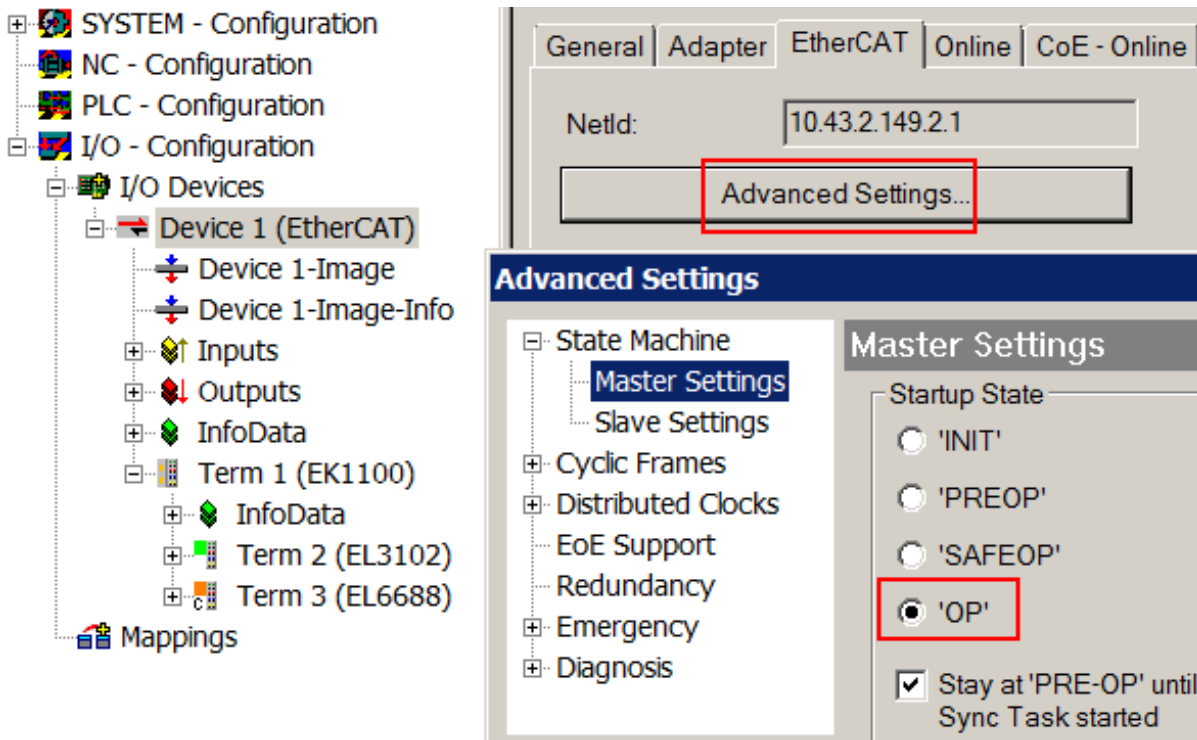


Fig. 146: Default behaviour of the System Manager

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

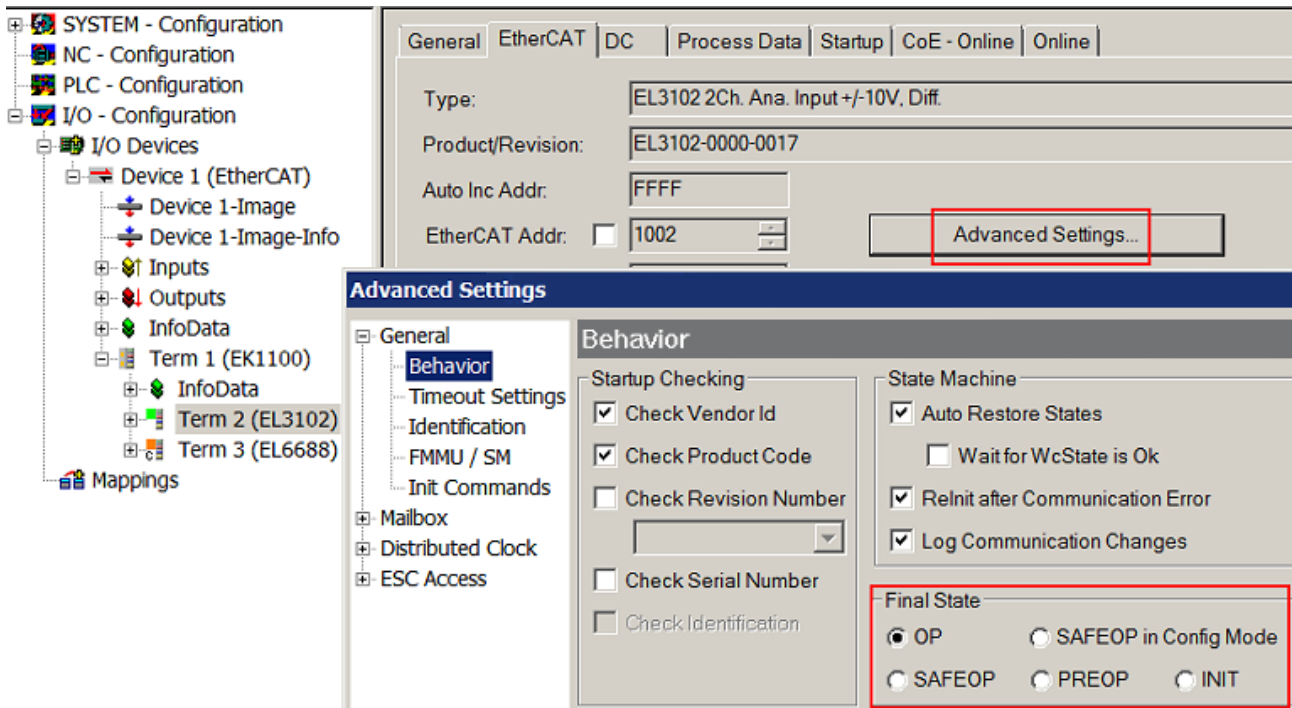


Fig. 147: Default target state in the Slave

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.

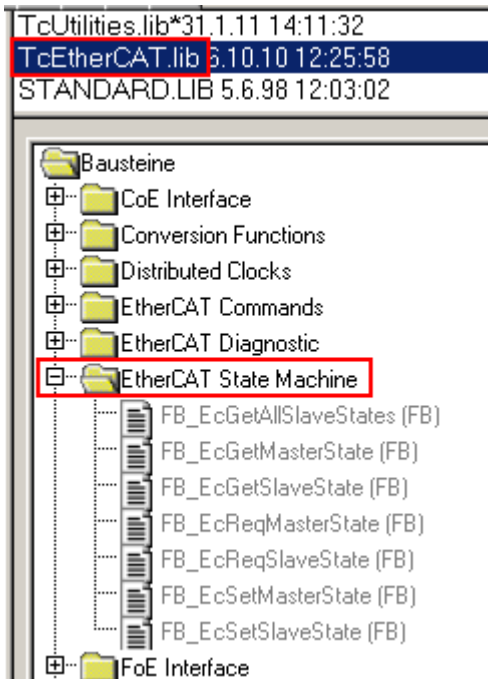


Fig. 148: PLC function blocks

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. 149: 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:

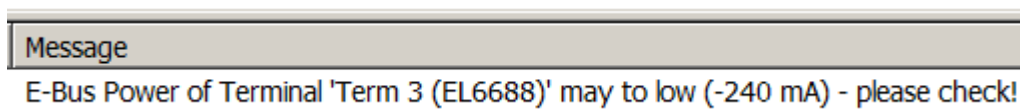


Fig. 150: Warning message for exceeding E-Bus current

NOTE

Caution! Malfunction possible!

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

5.4 Settings and application notes for EL32xx-0010/0020

Default setting

The EL32xx-0000 can be used for direct temperature or resistance measurements. Version -0010/-0020 offers enhanced measurement precision. The corresponding CoE settings are shown in the following table.

The relationship between temperature and resistance of a Pt100/Pt1000 sensor is shown below:

Temperature	typical resistance, approx.
850°C	Pt1000: 3.9 kΩ Pt100: 390 Ω
320°C	Pt1000: 2.2 kΩ Pt100: 220 Ω
-200°C	Pt1000: 180 Ω Pt100: 18 Ω

Characteristic sensor curves are available from the sensor manufacturers.

You can obtain further information in the chapters [Object description and parameterization \[► 158\]](#) and [Process data and operating modes \[► 149\]](#).

Notes regarding EL320x-0010/-0020 series (high-precision)

Thanks to its special hardware and software properties the EL320x-0010/0020 is able to measure the resistance value and thus the temperature with high accuracy. Please note:

i Increased accuracy EL320x-0010/0020

In the case of the high-precision terminals the settings in the object `0x80n0 [► 158]` / RTD-Settings must not be changed! The high-precision resistance/temperature measurement can only be performed in the range 10 to 220 Ohm, corresponding to a Pt100 measuring range of -200 to 320 °C.

In order to be able to display the high accuracy, the 0.01 °C/digit display is preset. The increased accuracy of the EL320x-00x0 is therefore valid only for the following settings:

- Resistance measurement or Pt100 characteristic curve (0.00385 Ω/Ω/°C, IEC60751 characteristic curve Pt385) – other characteristic curves are not guaranteed!
- Ambient temperature of 40 °C
- 4-wire connection technique
- Measuring range -200 to 320 °C
- 50 Hz filter

Temperatures or resistances outside of the range specified above cannot be measured with high precision. Outside these benchmark figures the high-precision terminal behaves like a terminal with standard accuracy.

To ensure that the measurements are carried out with the required increased accuracy in the long term, it is recommended that the measurement accuracy be checked at regular intervals, e.g. annually. Ageing effects or environmental influences can then be compensated by recalibration (user calibration).

	EL32xx-0000	EL320x-0010/-0020
Default/factory setting	<ul style="list-style-type: none"> • 2-wire connection • Pt100 (CoE 0x80n0:19) • Presentation signed (CoE 0x80n0:02) • Limits disabled • 50 Hz filter enabled • All channels enabled <p>for EL3214:</p> <ul style="list-style-type: none"> • 3-wire connection • Pt100 (CoE 0x80n0:19) • Presentation signed (CoE 0x80n0:02) • 50 Hz filter enabled • All channels enabled 	<ul style="list-style-type: none"> • 4-wire connection (CoE 0x80n0:1A) The connection type is not detected automatically. • Pt100 (CoE 0x80n0:19) Measurements take place in a reduced measuring range of -200 to 320 °C or an equivalent resistance range. <ul style="list-style-type: none"> ◦ The type label "Pt100 (-200..850°C)" indicates the usual measuring range of a Pt100 element • Presentation HighResolution (CoE 0x80n0:02) The temperature is displayed with 1/100 °C per digit (signed). • Limits disabled • 50 Hz filter enabled • All channels enabled
Area of application	The terminal is calibrated in the measuring range "1/16 Ω" (10 Ω to 4 kΩ) and can be used in this resistance range.	The high-precision version is calibrated in the measuring range "1/64 Ω" (10 Ω to 1 kΩ) and over a measuring range of 18 to 220 Ω and should only be used in this range or for Pt100 measurements. In other measuring ranges it offers the standard precision of $\pm 0.5\text{ °C}$.

Notice regarding Resistance Measurement mode

In Resistance Measurement mode the measured value is always displayed unsigned, irrespective of the Presentation setting (object 0x80n0:02 [▶ 159]), as 0..xFFFF with the respective value.

1/16 Ω -> ~62 mΩ/Digit

1/64 Ω -> ~15 mΩ/Digit

Changing the slave parameters in the CoE directory

During commissioning it may be necessary to change the CoE parameters.

- If the terminal is accessed online, the CoE directory is read from the slave and can be changed, see fig. "Accessing the CoE parameters in the TwinCAT System Manager".
- If the terminal is accessed offline (the terminal is then not addressable) the data from the XML description is used. The parameters cannot be changed.

The CoE parameters (CAN over EtherCAT) can be changed in the "CoE-Online" tab of the EtherCAT slave.

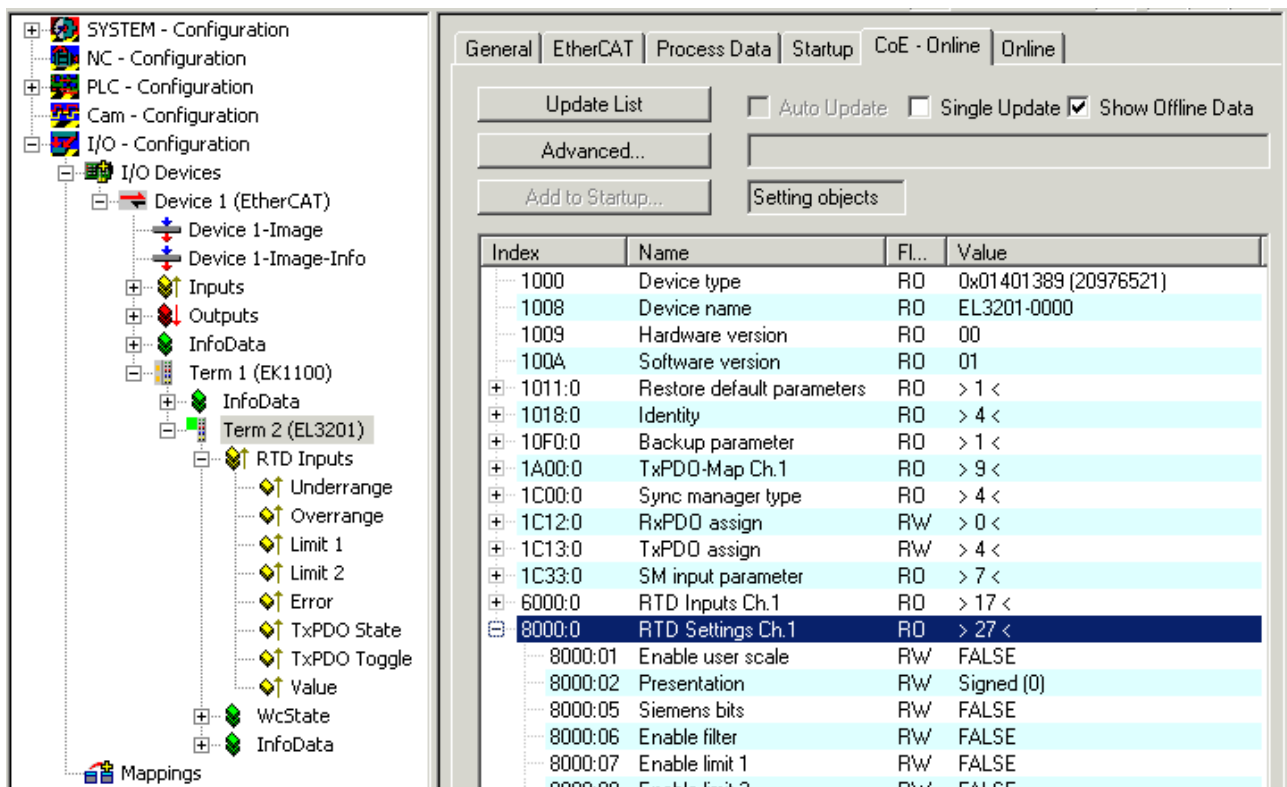


Fig. 151: Accessing the CoE parameters in TwinCAT System Manager , EL3201

NOTE

Changes in the CoE directory, in the case of exchange

So that, in the case of exchanging an EtherCAT slave, the CoE parameters are set correctly in the new slave, changes on the customer side must be entered in the start-up list.

Changes in the CoE directory, program access

The CoE parameters can also be read and written during operation (insofar as it makes sense) by the PLC using corresponding control commands. For details, see the corresponding articles in the Beckhoff Information System, for example.

For detailed information on settings and operating modes, please read the chapter "Process data and operating modes" [► 149].

Further application notes

"Filter enable" setting

The filter function is always active, even if the object 8000:06 "Filter enable" is set to FALSE.

"Connection technology" setting

Besides 2, 3 and 4-wire connection**, "not connected" can also be selected for each channel. The channel concerned is completely disabled and the conversion time of the EL32xx is correspondingly shortened.

**) 4-wire not valid for EL3214

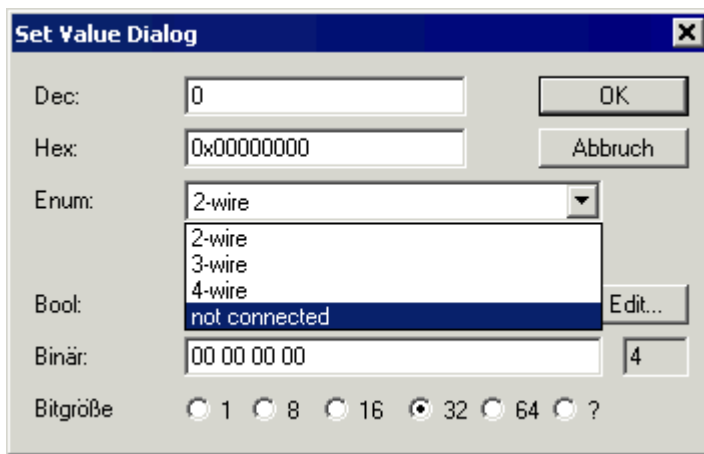


Fig. 152: “not connected” setting

By means of a double click on the corresponding object `0x80n0:1A [► 159]`, the respective channels can be switched off in the Set Value Dialog by selecting the “Enum” value “not connected”; see fig. “not connected” setting”.

5.5 Adjustments and application notes for EL3204-0200

The EL3204-0200 differs from the EL3204-0000 (standard type) in terms of an

- extended measuring range 0 ... 240 k Ω
- additional linearizations/characteristic sensor curves
 - programmed characteristic sensor curves match the measuring range
 - freely programmable according to sampling points table
 - parameterizable formulas according to IEC 60751, Steinhart-Hart equation and B-parameter equation

NOTE

Changes in the CoE directory, in the case of exchange

So that, in the case of exchanging an EtherCAT slave, the CoE parameters are set correctly in the new slave, CoE changes on the customer side must be entered in the start-up list.

i Changes in the CoE directory, program access

The CoE parameters can also be read and written during operation (insofar as it makes sense) by the PLC using corresponding control commands. For details, see the corresponding articles in the Beckhoff Information System, for example.

Notice: The CoE parameters should be written in status PreOp or SafeOp, after which the slave should be switched to status OP.

Measuring range

The specified terminal precision refers to the full-scale value for the resistance measurement.

It is not advisable to use the terminal in the range 0 to 10 Ω , due to the relatively low (relative) measuring accuracy in this range. Specialized resistance measuring terminals such as the [EL3692](#) with 4-wire measurement are to be used for such measurements.

Notes regarding accuracy

In addition to the accuracy of the terminal resistance measurement, the measurement result is influenced by further factors, as described below:

- Measuring error of the measuring sensor used
- Measuring error of the terminal: < +/- 0.3% of full-scale value (240 kΩ for resistance measurement); < +/- 0.5°C for Pt sensors
- Implementation of the calculation parameters in the firmware, e.g. number of sampling points, accuracy of the calculated parameters,...

We recommend verifying the settings against a calibrated temperature source.

Programmed characteristic sensor curves

• Direct resistance measurement

The EL3204-0200 can be used for direct resistance measurement in 3 measuring ranges (n = channel 0... 3)

CoE 0x80n0:19 [▶ 169] setting	Measuring range	Resolution/digit
0x101	0 ...6,553.5 Ω	0.1 Ω
0x102	0 ..0.650.535 Ω	1 Ω
0x103	0 ...240,000 Ω	10 Ω

Notice: The way in which the EL3204-0200 measures is electrically the same with these 3 settings. Only the displayed resolution is changed in the firmware. The full-scale value is therefore 240 kΩ for all 3 measurements.

• Fixed characteristic sensor curves

Further programmed characteristic sensor curves in the defined measuring range can be selected individually in the CoE entry [0x80n0:19](#) [[▶ 169](#)] for each channel (n = channel 0... 3).

Type	Resistance range	Implemented temperature range	Resolution/digit
Pt1000 (0.00385 Ω/Ω/°C, IEC60751 characteristic curve Pt385) (PTC)	~180 ... ~ 3,900 Ω	-200°C to 850°C	0.1°C
Ni1000 (PTC)	~700 ... ~ 2,300 Ω	-60°C to 250°C	0.1°C
Ni1000 TK1500 (at 100°C: 1500 Ω) (PTC)		-30 to 160°C	0.1°C
NTC 5k (0°C: 16325 Ω)	~167,000 ... ~340 Ω	-40...100°C	0.1°C
NTC 10k (0°C: 32650 Ω)	~176,000 ... ~180 Ω	-30...150°C	0.1°C
NTC 10k (0°C: 29490 Ω)	~135,000 ... ~240 Ω	-30...150°C	0.1°C
NTC 10k (0 °C: 27280 Ω, same as KL3204-0030)	~190,000 ... ~760 Ω	-40...110°C	0.1°C
NTC 20k (0°C: 70200 Ω)	~221,000 ... ~270 Ω	-20...150°C	0.1°C
FeT (0°C: 2226 Ω)	~1,900 ... ~ 3,900 Ω	-30...150°C	0.1°C

Freely programmable linearization

To connect a resistive sensor with a characteristic curve that is known but not implemented in the EL32xx-xxxx, the characteristic curve can be entered in the terminal via one of four different options. Each linearization and each NTC/PTC characteristic curve can be realized within the measuring range. The suitability of the different implementation procedures depends on the sensor type.

CoE entry in <u>0x80n0:19</u> [▶ <u>169</u>]	Characteristic curve type	PTC sensors (e.g. Pt100, Pt1000, Ni100, Ni1000,...)	NTC sensors (NTC 5k, NTC 10k,...)
0x110	RTD table/sampling points table	X	X
0x111	B-parameter equation	-	X
0x112	DIN EN 60751 equation	X (standard for Pt sensors)	-
0x113	Steinhart-Hart equation	-	X

Select the procedure for the channel in CoE entry 0x80n0:19 [▶ 169]. Please note that for each procedure only 1 parameter set can be stored in the terminal, which is then available for all channels.

Example: 2 sensors are connected to channels 1 and 2. Both are to be linearized with the B-parameter equation. The two sensors have to be "identical", since it is not possible to specify different sets of B-parameters for channels 1 and 2. The same applies to the sampling points table.

• **Sampling points table (0x80n0:19, entry 0x110)**

The required characteristic curve for a PTC or NTC sensor can be stored as a table in CoE 0x8001 [▶ 170]. It can either be loaded directly from the PLC into the CoE, or the user can enter it in the startup list for the terminal as an XML file. The entries from the TwinCAT startup list are automatically written to the terminal whenever EtherCAT starts.

The sampling points table generally has the following structure: A temperature is allocated to the corresponding resistance. The temperature increment must be constant. Between these values linearization is applied.

The maximum number of sampling points is 100, the maximum measuring range is 0...240 kΩ.

The precise structure of the sampling points table is described in the CoE overview [▶ 168] and additionally in an example file (<https://infosys.beckhoff.com/content/1033/el32xx/Resources/zip/1525761547.zip>) for a Pt1000 sensor.

The sample file illustrates the structure of an XML file for import. An Excel file (Excel 2010 format) for creating an XML file is also included. Explanations about the use of the Excel file can be taken from the attached description.

The XML file can be imported directly via the Startup field and activated via "Activate configuration".

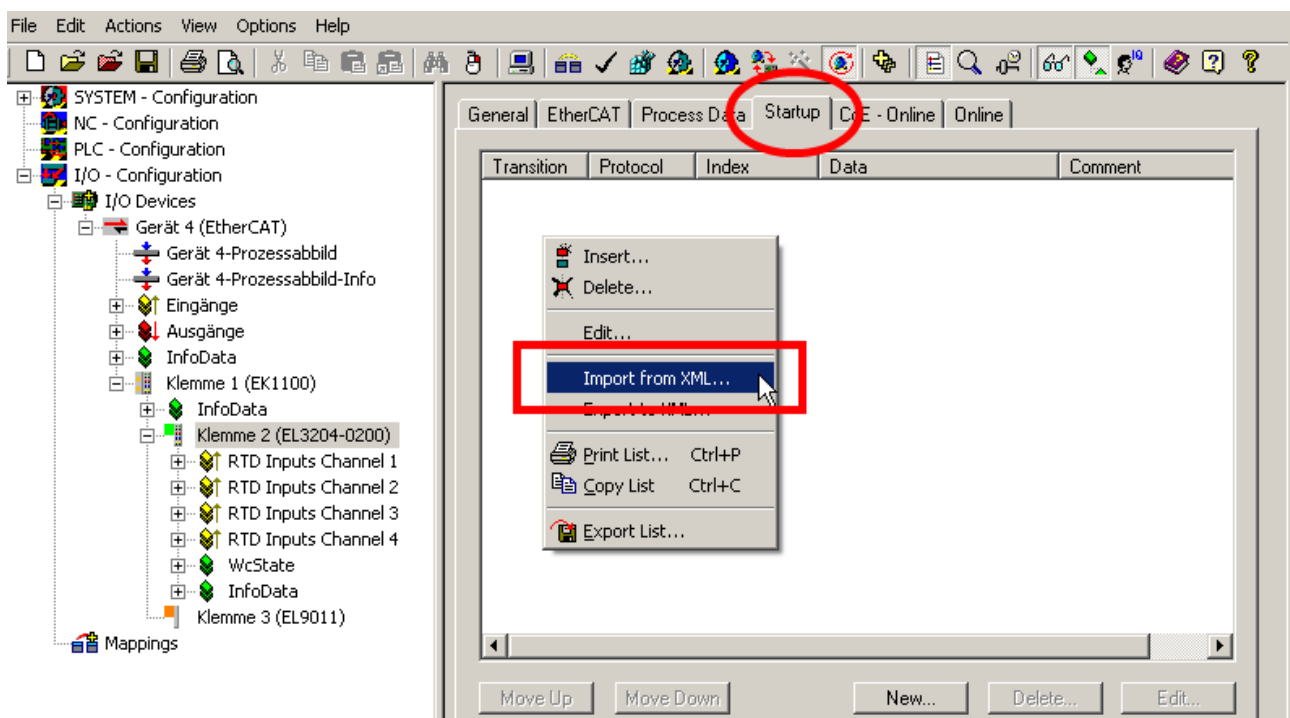


Fig. 153: Importing the XML file

- **B-parameter equation (0x80n0:19, entry 0x111)**

The B-parameter equation can be used for NTC sensors (thermistors), i.e. RTD elements with negative coefficient k.

$$R_T = R(T) = A \cdot e^{\frac{B}{T}} = R_{T_0} \cdot e^{B\left(\frac{1}{T} - \frac{1}{T_0}\right)}$$

The coefficient R_{T_0} indicates the resistance at temperature T_0 . The B-parameter can be taken from the information provided by the sensor manufacturer, or it can be determined by measuring the resistance at two known temperatures.

The following example file (<https://infosys.beckhoff.com/content/1033/el32xx/Resources/zip/1525763723.zip>) can be used for the calculation of the B-parameter. Enter the calculated parameters in the CoE directory 0x8002 [▶ 171].

	A	B	D
1	To solve the B-Parameter Equation 2 resistor parameters at 2 defined		
2	temperatures are required:		
3			
4		Temperatur [°C]	Resistance [Ω]
5	Value 1	25	10000
6	Value 2	85	1450
7			
8	CoE Index	Name	CoE Value
9	8002:0	RTD B-Parameter Equation	
10	8002:01	Min Temperature	-50
11	8002:02	Max Temperature	110
12	8002:03	Rn	10000
13	8002:04	Tn	25
14	8002:05	B-Parameter	3437

Fig. 154: Table for calculating the B-parameter

- **Programmable according to DIN 60751 for Pt sensors (0x80n0:19, entry 0x112)**

IEC 751 (and the German version DIN EN 60751) defines the electrical properties of platinum temperature sensors. The characteristic curve is described as follows:

for temperatures -200.. 0°C:

$$R(T) = R_0(1 + AT + BT^2 + C(T - 100^\circ\text{C})T^3)$$

for temperatures 0.. 850°C:

$$R(T) = R_0(1 + AT + BT^2)$$

With parameters:

$$A = 3.9083 \times 10^{-3} \text{ }^\circ\text{C}^{-1}$$

$$B = -5.775 \times 10^{-7} \text{ }^\circ\text{C}^{-2}$$

$$C = -4.183 \times 10^{-12} \text{ }^\circ\text{C}^{-4}$$

The parameter R_0 indicates the resistance in ohm of the platinum sensor at $T = 0 \text{ }^\circ\text{C}$. The sensor designations are based on these characteristics, e.g. for Pt100 $R_0 = 100 \text{ } \Omega$ at $T=0 \text{ }^\circ\text{C}$.

The parameters A, B and C defined in the standard are already stored in CoE directory 0x8003:yy [▶ 171].

The Pt sensor is specified via parameter 0x8003:03 [▶ 171], which corresponds to R_0 .

• **Steinhart-Hart equation (0x80n0:19, entry 0x113)**

The Steinhart-Hart equation can be used for NTC sensors (thermistors), i.e. RTD elements with negative coefficient k.

$$\frac{1}{T} = C_1 + C_2 \ln(R) + C_4(\ln(R))^3$$

The coefficients C_1 , C_2 and C_4 can either be taken directly from the manufacturer data or calculated. An example file (<https://infosys.beckhoff.com/content/1033/el32xx/Resources/zip/1525765899.zip>) is available for the calculation of the Steinhart-Hart parameters. For determining the parameters the resistance values at three known temperatures are required. These can either be taken from the manufacturer data or measured directly at the sensor. Enter the calculated parameters in the CoE directory 0x8004 [► 172]. In most cases the parameter C_3 is close to zero, i.e. negligible. It is therefore not used in the sample file calculation.

EL3204-0200_Steinhart-Hart.xlsx

	A	B	D
1	To solve the Steinhart-Hart Equation 3 resistor parameters at 3 defined		
2	temperatures are required:		
3		Temperatur [°C]	Resistance [Ω]
4	Value 1	0	27280
5	Value 2	25	10000
6	Value 3	85	1450
7			
8	CoE Index	Name	CoE Value
9	8004:0	RTD Steinhart-Hart Equation	
10	8004:01	Min Temperature	-50
11	8004:02	Max Temperature	110
12	8004:03	C1	8,84E-04
13	8004:04	C2	2,52E-04
14	8004:05	C3	0,00E+00
15	8004:06	C4	1,90E-07

Fig. 155: Fig. 2: Table for calculating the Steinhart-Hart parameters

Sample implementation of an NTC sensor

The procedure below illustrates the implementation of an NTC sensor whose characteristic curve is not already stored in the EL3204-0200 based on three methods, i.e. RTD table, B-parameter equation and Steinhart-Hart equation.

An NTC030WP00 sensor is used in the example.

In this case the manufacturer provides a table with temperature and corresponding resistance values and the following sensor characteristics: NTC 10kΩ +/- 1% at 25°C, Beta 3435.

• **Implementation based on sampling points table, RTD table**

The general procedure is described in section [Sampling points table \[► 143\]](#). This example illustrates some special cases. An example XML file (<https://infosys.beckhoff.com/content/1033/el32xx/Resources/zip/1525768075.zip>) for the NTC030WP00 sensor is available for download.

Description	Problem	Approach																		
The table covers the temperature range -50 °C... 110 °C, in 1 °C steps	Total number of sampling points is 160. Once hundred sampling points can currently be entered with the aid of the RTD table.	<ol style="list-style-type: none"> 1. Limit the measuring range to correspond with 100 sampling points in total. 2. Increase the step size, e.g. implement a table with 5 °C steps. The step size should not be too large, since NTC sensors have an exponential characteristic curve (i.e. not linear such as the PTC sensors, for example). 																		
Resistance values in the temperature range -50 °C.. -44 °C are in the range 330 kΩ.. 240 kΩ	Measurement readings at low temperatures exceed the total measuring range of 240 kΩ	<ol style="list-style-type: none"> 1. Check whether the measuring range is required 2. Select an alternative sensor with lower resistance values at low temperatures. 																		
<p>The range -40 °C...110°C should be mapped in 2°C steps. In addition to the table the following settings are implemented in CoE 0x8001:</p> <table border="1"> <tr> <td>0x8001:01 [▶ 170] Distance in Degrees</td> <td>2</td> </tr> <tr> <td>0x8001:02 [▶ 170] Amount of Entries</td> <td>75</td> </tr> <tr> <td>0x8001:03 [▶ 170] Start Temperature</td> <td>-40</td> </tr> <tr> <td>0x8001:04 [▶ 170] Conversion Factor</td> <td>10</td> </tr> <tr> <td>0x8001:05 [▶ 170] Value 1</td> <td>188500</td> </tr> </table>	0x8001:01 [▶ 170] Distance in Degrees	2	0x8001:02 [▶ 170] Amount of Entries	75	0x8001:03 [▶ 170] Start Temperature	-40	0x8001:04 [▶ 170] Conversion Factor	10	0x8001:05 [▶ 170] Value 1	188500	The default settings of conversion factor 10 result in errors: The measuring range was exceeded. Adjust the unit of the resistance values to the conversion factor	<p>Note the measuring range. For the range 0...240 kΩ a conversion factor of 100 should be entered in 0x8001:04 [▶ 170]. The resistance values should then be specified in 1/10 Ω.</p> <p>The ranges are defined as follows:</p> <table border="1"> <thead> <tr> <th>Conversion Factor 0x8001:04 [▶ 170]</th> <th>Measuring range</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>0.. 6,533.5 Ω</td> </tr> <tr> <td>10</td> <td>0..65,535.0 Ω</td> </tr> <tr> <td>100</td> <td>0..240,000.0 Ω</td> </tr> </tbody> </table>	Conversion Factor 0x8001:04 [▶ 170]	Measuring range	1	0.. 6,533.5 Ω	10	0..65,535.0 Ω	100	0..240,000.0 Ω
0x8001:01 [▶ 170] Distance in Degrees	2																			
0x8001:02 [▶ 170] Amount of Entries	75																			
0x8001:03 [▶ 170] Start Temperature	-40																			
0x8001:04 [▶ 170] Conversion Factor	10																			
0x8001:05 [▶ 170] Value 1	188500																			
Conversion Factor 0x8001:04 [▶ 170]	Measuring range																			
1	0.. 6,533.5 Ω																			
10	0..65,535.0 Ω																			
100	0..240,000.0 Ω																			

• Implementation based on B-parameter equation

The B-parameter (Beta 3435) and the parameters R_n (10 kΩ) and T_n (25°C) can be taken from the manufacturer's data and entered directly in the CoE directory 0x8002 [▶ 171].

CoE Entry	Entry
0x8002:01 Min Temperature	-40
0x8002:02 Max Temperature	110
0x8002:03 R_n	10000
0x8002:04 T_n	25
0x8002:05 B-Parameter	3435

If one compares the implementation of the table and the B-parameter equation, as shown in the diagram below, it becomes clear that the B-parameter equation is only valid in a limited range (large deviations at low temperatures). If the whole temperature range is required, it is preferable to use the Steinhart-Hart equation for the implementation.

• Implementation via the Steinhart-Hart equation

The Steinhart-Hart parameters can be calculated based on the sample file. The required resistance values at three defined temperatures (e.g. 0°C, 25°C and 85°C) can be taken from the manufacturer's table or measured direct at the sensor.

CoE Entry	Entry
0x8004:01 Min Temperature	-40
0x8004:02 Max Temperature	110
0x8004:03 C1	8.8424E-4

CoE Entry	Entry
0x8004:04 C2	2.5202E-4
0x8002:04 C3	0
0x8002:04 C4	1.9018E-7

The diagram shows that the implementation via the Steinhart-Hart equation is closer to the values from the manufacturer's table than the implementation via the B-parameter equation. If a manufacturer's table is available, this method is always preferable.

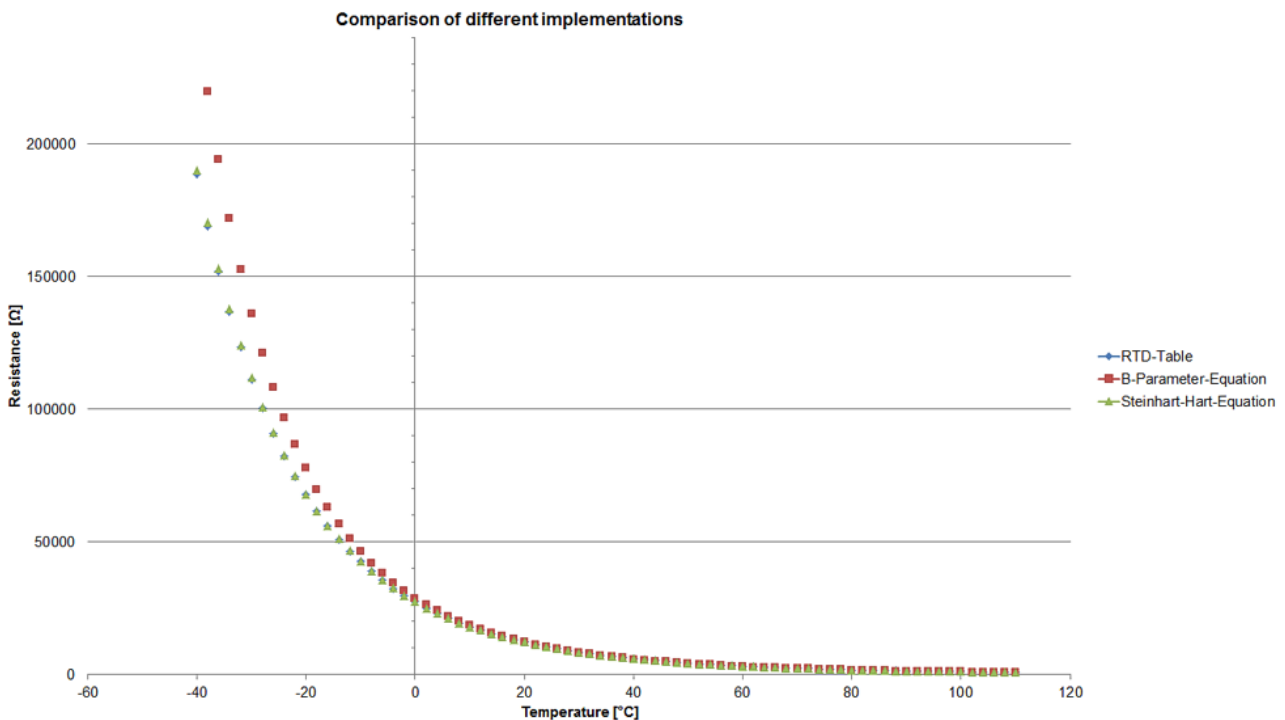


Fig. 156: Comparison of implementation via B-parameters and Steinhart-Hart parameters

Sample implementation of an PTC sensor

Sample implementation of an PTC sensor

The example below illustrates the implementation of a PTC sensor, in particular a platinum sensor (Pt sensor), based on the RTD table and according to DIN EN 60751.

• **Implementation based on sampling points table, RTD table**

The general procedure is explained in section [Sampling points table](#) [▶ 143]. The example describes the implementation of table for a Pt500 sensor.

Description	Problem	Approach
For Pt sensors tables with several 100 entries in 1 °C steps are available.	The total number of sampling points is exceeded. Currently, up to 100 sampling points can be entered based on the RTD table.	<ol style="list-style-type: none"> 1. Limit the measuring range to correspond with 100 sampling points in total. 2. Increase the step size, e.g. implement a table with 5 °C or 10 °C steps. Since the Pt sensor curves are almost linear, an increment value only has a small effect on the measuring accuracy. This is in contrast to NTC sensors, where the curve is exponential and a smaller increment is therefore preferable.

Description	Problem	Approach								
The manufacturer tables contain universal values	The table values should be adjusted to the specific sensor type	For different Pt sensors the table entries have to be multiplied with the R_0 value. R_0 indicates the resistance at 0°C. The sensor name is based on the temperature value.								
		<table border="1"> <thead> <tr> <th>Pt sensor</th> <th>R (resistance at 0°C)₀</th> </tr> </thead> <tbody> <tr> <td>Pt100</td> <td>100</td> </tr> <tr> <td>Pt500</td> <td>500</td> </tr> <tr> <td>Pt1000</td> <td>1000</td> </tr> </tbody> </table>	Pt sensor	R (resistance at 0°C) ₀	Pt100	100	Pt500	500	Pt1000	1000
		Pt sensor	R (resistance at 0°C) ₀							
		Pt100	100							
Pt500	500									
Pt1000	1000									

For the Pt500 sensor the following CoE entries can be selected and edited with the following XML file (<https://infosys.beckhoff.com/content/1033/el32xx/Resources/zip/1525770251.zip>):

CoE Entry	Entry
0x8001:01 Distance in Degrees	10
0x8001:02 Amount of Entries	60
0x8001:03 Start Temperature	-40
0x8001:04 Conversion Factor	10
0x8001:05 Value 1	4247
0x8001:0x further table entries

• Programmable according to DIN 60751 for Pt sensors

The implementation according to DIN EN 60751 is described in [Programmable according to DIN 60751 for Pt sensors \[▶ 144\]](#). The parameters A, B, C are already stored in the CoE. Only the temperature range and the R_0 parameter for the specific sensor has to be added. R_0 indicates the resistance at 0°C. The sensor name is based on the temperature value. The CoE entry has to be adjusted for the specific sensor.

Pt sensor	CoE Entry 0x8003:03
Pt100	100
Pt200	200
Pt500	500
Pt1000	1000

If one compare curve showing the resistance values over temperature, the RTD table and according to DIN EN 60751, it is clear that both implementation methods deliver almost the same result. Parameterization according to DIN 60571 is simpler and therefore preferable.

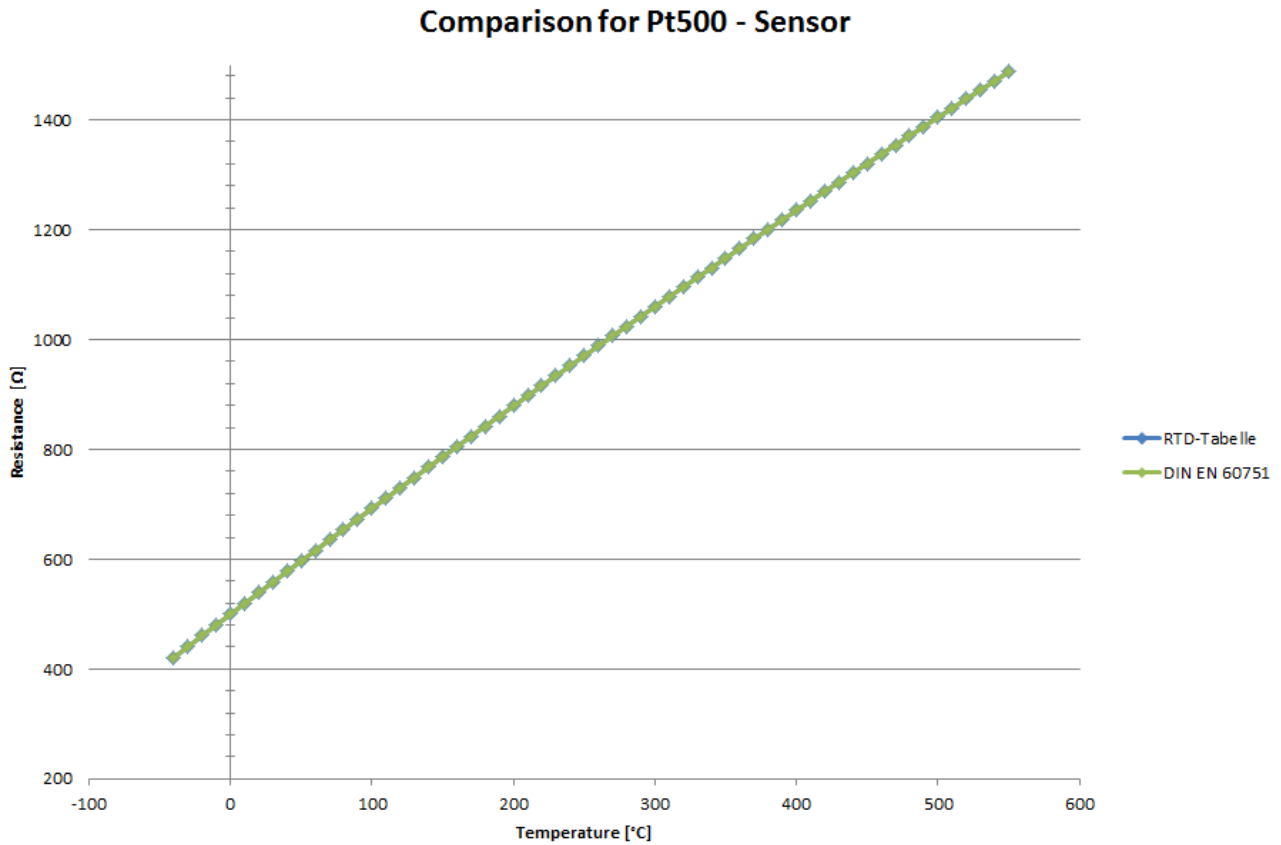


Fig. 157: Comparison of Pt500 sensor implementation

5.6 Process data and operating modes

Table of contents
• Data flow [▶ 150]
• Sync Manager [▶ 150]
• Operating modes and settings [▶ 150]
- Presentation [▶ 150]
- Siemens Bits [▶ 151]
- Underrange, Overrange [▶ 151]
- Notch filter (conversion times) [▶ 151]
- Limit 1 and Limit 2 [▶ 153]
- Calibration [▶ 153]
- Producer Codeword [▶ 154]
• Interference from equipment [▶ 154]

Data flow

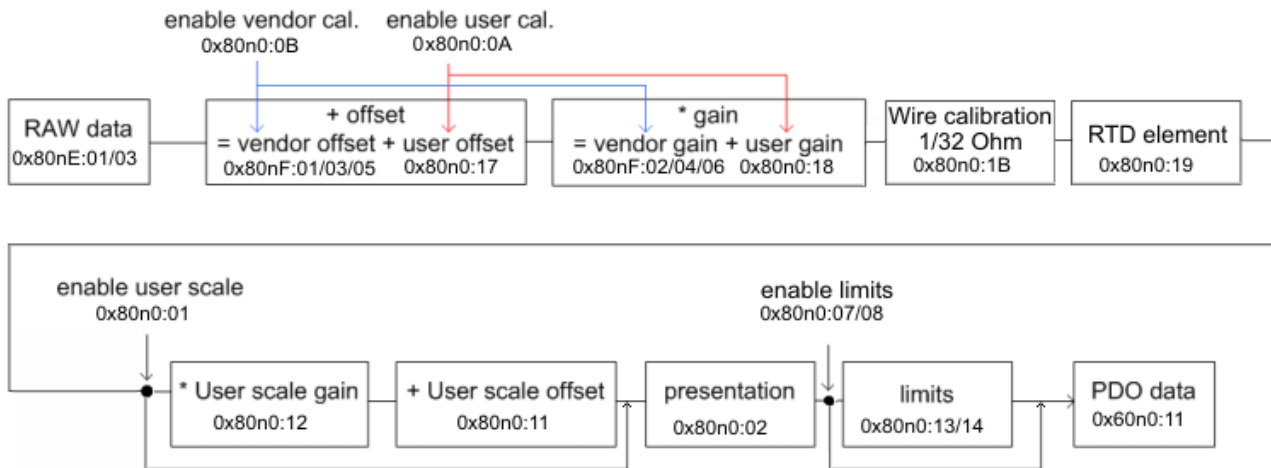


Fig. 158: Data diagram

Sync Manager (SM)

• **PDO Assignment**

SM3, PDO assignment 0x1C13, for 0 ≤ n ≤ 7 (Ch. 1 - 8) (depending on number of channels)				
Index	Index of excluded PDOs	Size (byte.bit)	Name	PDO content
0x1A0n (default)	-	4.0	RTD Inputs Channel 1 - 8	Index 0x60n0:01 [▶ 167] - Underrange Index 0x60n0:02 [▶ 167] - Overrange Index 0x60n0:03 [▶ 167] - Limit 1* Index 0x60n0:05 [▶ 167] - Limit 2* Index 0x6000:07 [▶ 167] - Error Index 0x180n:07 - TxPDO Status Index 0x180n:09 - TxPDO Toggle Index 0x60n0:11 [▶ 167] - Value

*) not for EL3214, EL3208

Operating modes and settings

Presentation, index 0x80n0:02 [▶ 159]

In the delivery state, the measured value is output in increments of 1/10° C in two's complement format (signed integer).

The complete measuring range is output for each resistance sensor. Index 0x80n0:02 [▶ 159] offers the possibility to change the method of representation of the measured value.

Measured value	Output (hexadecimal)	Output (signed integer, decimal)
-250.0 °C	0xF63C	-2500
-200.0 °C	0xF830	-2000
-100.0 °C	0xFC18	-1000
-0.1 °C	0xFFFF	-1
-0.0 °C	0x0000	0
-0.1 °C	0x0001	1
100.0 °C	0x03E8	1000
200.0 °C	0x07D0	2000
500.0 °C	0x1388	5000

Measured value	Output (hexadecimal)	Output (signed integer, decimal)
850.0 °C	0x2134	8500

• **Signed Integer**

The measured value is presented in two's complement format.
 Maximum presentation range for 16 bit = -32768 .. +32767

Example:

$1000\ 0000\ 0000\ 0000_{bin} = 8000_{hex} = -32768_{dec}$
 $1111\ 1111\ 1111\ 1110_{bin} = FFFE_{hex} = -2_{dec}$
 $1111\ 1111\ 1111\ 1111_{bin} = FFFF_{hex} = -1_{dec}$
 $0000\ 0000\ 0000\ 0001_{bin} = 0001_{hex} = +1_{dec}$
 $0000\ 0000\ 0000\ 0010_{bin} = 0002_{hex} = +2_{dec}$
 $0111\ 1111\ 1111\ 1111_{bin} = 7FFF_{hex} = +32767_{dec}$

• **Absolute value with MSB as sign**

The measured value is output in magnitude-sign format.
 Maximum presentation range for 16 bit = -32767 .. +32767

Example:

$1111\ 1111\ 1111\ 1111_{bin} = FFFF_{hex} = -32767_{dec}$
 $1000\ 0000\ 0000\ 0010_{bin} = 8002_{hex} = -2_{dec}$
 $1000\ 0000\ 0000\ 0001_{bin} = 8001_{hex} = -1_{dec}$
 $0000\ 0000\ 0000\ 0001_{bin} = 0001_{hex} = +1_{dec}$
 $0000\ 0000\ 0000\ 0010_{bin} = 0002_{hex} = +2_{dec}$
 $0111\ 1111\ 1111\ 1111_{bin} = 7FFF_{hex} = +32767_{dec}$

• **High resolution (1/100 C°)**

The measured value is output in 1/100 °C steps.
 As a result, the maximum measuring range is limited to $32767_{dec} * (1/100) °C = 327.67 °C$, but can be extended by the user scaling [[▶ 153](#)]!

Siemens bits, index 0x80n0:05 [[▶ 159](#)]

If this bit is set, status displays are superimposed on the lowest three bits. In the error case "overrange" or "underrange", bit 0 is set.

Undershoot and overshoot of the measuring range (under range, over range), index 0x60n0:0 [[▶ 167](#)] **0x60n0:02 [[▶ 167](#)]**

Pt100: $T > 850 °C$ ($R > 400 \Omega$): Index 0x60n0:02 and index 0x60n0:07 (over range and error bit) are set. The linearization of the characteristic curve is continued with the coefficients of the upper range limit up to the limit stop of the A/D converter (approx. 500 Ω for Pt100).

Pt100: $T < -200 °C$ ($R < 18 \Omega$): Index 0x60n0:01 and index 0x60n0:07 (under range and error bit) are set. The smallest negative number is displayed (0x8001 corresponds to -32767_{dec}).

For over range or under range the red error LED is switched on.

Notch filter, index 0x80n0:06 [[▶ 159](#)]

The ELX320x terminals are equipped with a digital filter. The filter performs a notch filter function and determines the conversion time of the terminal. It is parameterized via the index 0x8000:15 [[▶ 159](#)]. The higher the filter frequency, the faster the conversion time.

Notch filter means that the filter has zeros (notches) in the frequency response at the specified filter frequency and multiples thereof, i.e. it masks out these frequencies. In order to filter out 50 Hz, therefore, the 50 or the 10 Hz filter can be used. Since all frequencies lower than the notch frequency usually have to be measured, however, the 50 Hz setting is to be used in this case.

typ. Frequency response Notch filter

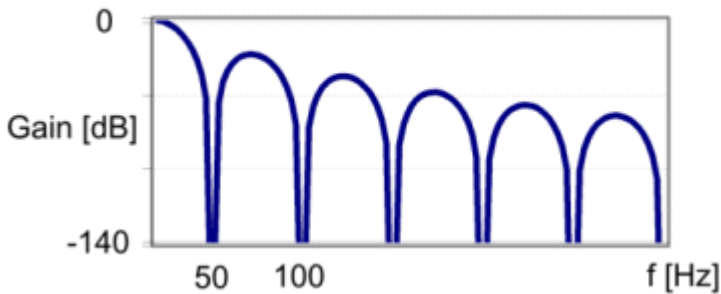


Fig. 159: Typical frequency response of a notch filter set to 50 Hz

● Index 80n0:06

i The filter function is always active even if the bit is not set, since this is obligatory for the measurement process!

Details of the conversion time

- the conversion time depends on *number of active channels, number of measurements (2/4-wire), filter time, const. computing time*
- the conversion time for the 2-wire measurement is about twice as long as for the 4-wire measurement
- the resistance measurement corresponds to a 4-wire measurement
- the process data *TxPDO Toggle* changes its state if a new measured value is present

● The filter characteristics are set via index 0x8000:15 [[▶ 159](#)]

i The filter frequencies are set for all channels of the ELX320x terminals centrally via index 0x8000:15 (channel 1). The corresponding indices 0x8010:15 of the ELX3202 or 0x8010:15, 0x8020:15, 0x8030:15 of the ELX3204 have no parameterization function.

● Switching off the unused channels of multi-channel terminals

i Unused channels should be switched off if filter frequencies are ≥ 1 kHz (optimization of conversion times)!
The channel concerned is completely disabled and the conversion time of the ELX320x is correspondingly shortened.

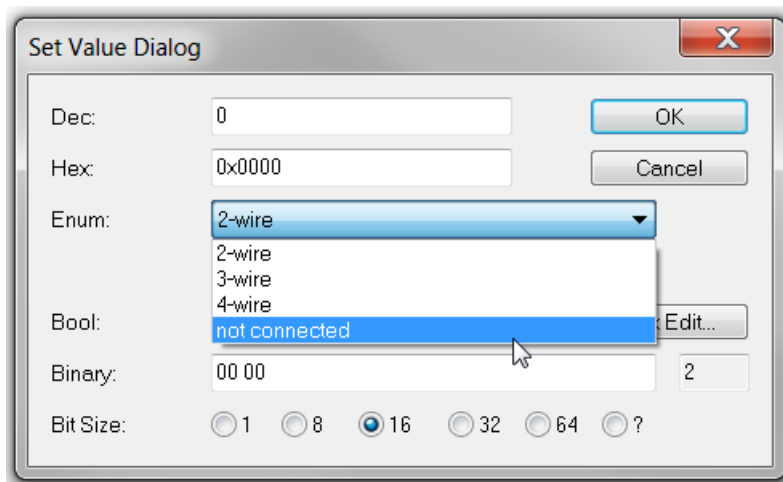


Fig. 160: Set Value Dialog

By means of a double click on the corresponding object [0x80n0:1A \[▶ 159\]](#), the respective channels can be switched off in the Set Value Dialog by selecting the Enum value *not connected*; see fig. *Set Value Dialog*.

Limit 1 and Limit 2, [0x80n0:13, 0x80n0:14 \[▶ 159\]](#) (not for EL3214 and EL3208)

A temperature range can be set that is compared with the values in the indices [0x80n0:13 \[▶ 159\]](#) and [0x80n0:14 \[▶ 159\]](#). The bits are set in the process data if the limit values are undershot or exceeded.

The limit evaluation takes place after taking into account the set characteristic curve and negative values.

Limit [2-bit] interpretation (default), index [0x60n0:03 \[▶ 167\]](#), [0x60n0:05 \[▶ 167\]](#):

- 0x00, 0: not active
- 0x01, 1: Value is larger than the limit value
- 0x10, 2: Value is smaller than the limit value
- 0x11, 3: Value is equal to the limit value

Calibration

• **User scaling, index [0x80n0:01 \[▶ 159\]](#)**

The user scaling is enabled via index [0x80n0:01 \[▶ 159\]](#). Parameterization takes place via the indices

- [0x80n0:11 \[▶ 159\]](#) User scaling offset
- [0x80n0:12 \[▶ 159\]](#) User scaling gain

• **Manufacturer calibration, [0x80nF \[▶ 168\]](#)**

The manufacturer compensation is enabled via index [0x80n0:0B \[▶ 159\]](#). Parameterization takes place via the indices

- [0x80nF:01 \[▶ 168\]](#) Calibration offset
- [0x80nF:02 \[▶ 168\]](#) Calibration gain
- [0x80nF:03 \[▶ 168\]](#) Calibration offset, input RL
- [0x80nF:04 \[▶ 168\]](#) Calibration gain, input RL
- [0x80nF:05 \[▶ 168\]](#) (not for EL3214 and EL3208), calibration offset, 4-wire connection technique
- [0x80nF:06 \[▶ 168\]](#) (not for EL3214 and EL3208), calibration gain, 4-wire connection technique

User calibration, index [0x80n0:18 \[▶ 159\]](#), [0x80n0:17 \[▶ 159\]](#)

The user calibration is enabled via index [0x80n0:0A \[▶ 159\]](#). Parameterization takes place via the indices

- [0x80n0:17 \[▶ 159\]](#) User calibration offset
- [0x80n0:18 \[▶ 159\]](#) User calibration gain

Process data calculation

The terminal records measured values continuously and places the raw values of its A/D converter into the ADC raw value object [0x80nE:01 \[▶ 167\]](#) or [0x80nE:02 \[▶ 167\]](#). The calculation of the correction with the vendor calibration values takes place after each acquisition of the analog signal. User scaling then follows (optionally):

$Y_H = (X_{ADC} - B_K) \times A_K$	<p>Measured value following manufacturer calibration (corresponds to Y_A if index 0x80n0:0A [▶ 159] is inactive)</p>
$Y_A = Y_H \times A_W \times 2^{-16} + B_W$	<p>Measured value following user scaling</p>

Legend

Name	Name	Index
X _{ADC}	Output value of the A/D converter	0x80nE:01 [▶ 167], 0x80nE:02 [▶ 167]
B _K	Manufacturer calibration offset (only changeable if the object <u>Producer codeword</u> [▶ 154] 0xF008 [▶ 168] is set)	0x80nF:01 [▶ 168], 0x80nF:03 [▶ 168]
A _K	Manufacturer calibration gain (only changeable if the object <u>Producer codeword</u> [▶ 154] 0xF008 [▶ 168] is set)	0x80nF:02 [▶ 168], 0x80nF:04 [▶ 168]
B _W	User scaling offset (can be activated via index 0x80n0:0A [▶ 159])	0x80n0:17 [▶ 159]
A _W	User scaling gain (can be activated via index 0x80n0:0A [▶ 159])	0x80n0:18 [▶ 159]
Y _A	Process data for controller	-

Producer Codeword

● Producer Codeword



The vendor reserves the authority for the basic calibration of the terminals. The Producer codeword is therefore at present reserved.

Interference from equipment

When operating the EL320x analog EtherCAT Terminals, high frequency superimposed signals from interfering devices (e.g. proportional valves, stepper motors or DC motor output stages) can be picked up by the terminal. In order to guarantee interference-free operation, we recommend the use of separate power supply units for the terminals and the interference-causing devices. The integrated filter of the terminal (see index 0x80n0:06 [▶ 159]) should be switched on as an additional measure.

5.7 TwinSAFE SC

5.7.1 TwinSAFE SC - operating principle

The TwinSAFE SC (Single Channel) technology enables the use of standard signals for safety tasks in any networks of fieldbuses. To do this, EtherCAT Terminals from the areas of analog input, angle/displacement measurement or communication (4...20 mA, incremental encoder, IO-Link, etc.) are extended by the TwinSAFE SC function. The typical signal characteristics and standard functionalities of the I/O components are retained. TwinSAFE SC I/Os have a yellow strip at the front of the housing to distinguish them from standard I/Os.

The TwinSAFE SC technology enables communication via a TwinSAFE protocol. These connections can be distinguished from the usual safe communication via Safety over EtherCAT.

The data of the TwinSAFE SC components are transferred via a TwinSAFE protocol to the TwinSAFE logic, where they can be used in the context of safety-relevant applications. Detailed examples for the correct application of the TwinSAFE SC components and the respective normative classification, which were confirmed/calculated by TÜV SÜD, can be found in the [TwinSAFE application manual](#).

5.7.2 TwinSAFE SC - configuration

The TwinSAFE SC technology enables communication with standard EtherCAT terminals via the Safety over EtherCAT protocol. These connections use another checksum, in order to be able to distinguish between TwinSAFE SC and TwinSAFE. Eight fixed CRCs can be selected, or a free CRC can be entered by the user.

By default the TwinSAFE SC communication channel of the respective TwinSAFE SC component is not enabled. In order to be able to use the data transfer, the corresponding TwinSAFE SC module must first be added under the Slots tab. Only then is it possible to link to a corresponding alias device.

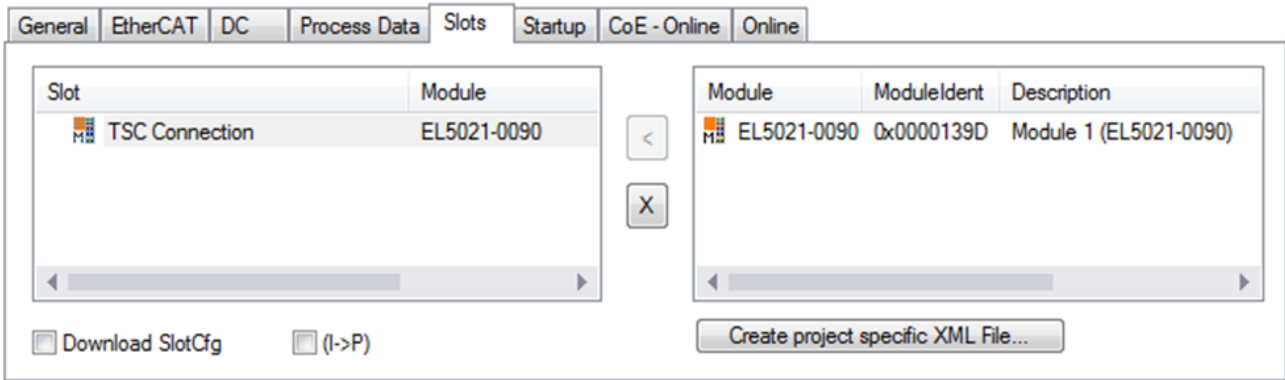


Fig. 161: Adding the TwinSAFE SC process data under the component, e.g. EL5021-0090

Additional process data with the ID TSC Inputs, TSC Outputs are generated (TSC - TwinSAFE Single Channel).

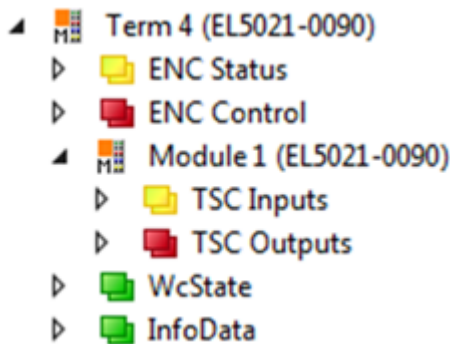


Fig. 162: TwinSAFE SC component process data, example EL5021-0090

A TwinSAFE SC connection is added by adding an alias devices in the safety project and selecting TSC (TwinSAFE Single Channel)

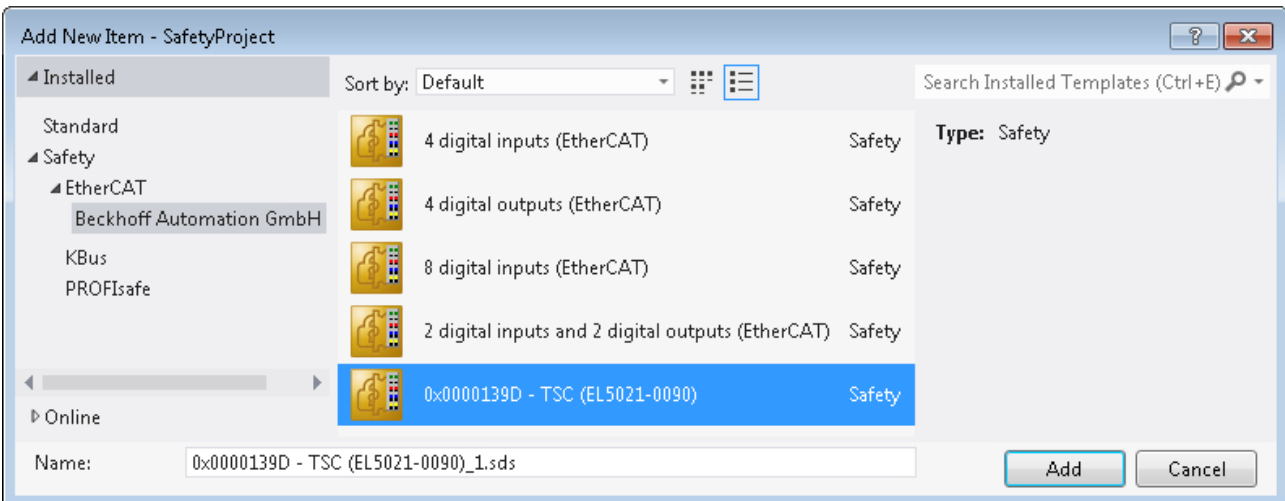



Fig. 163: Adding a TwinSAFE SC connection

After opening the alias device by double-clicking, select the Link button  next to *Physical Device*, in order to create the link to a TwinSAFE SC terminal. Only suitable TwinSAFE SC terminals are offered in the selection dialog.

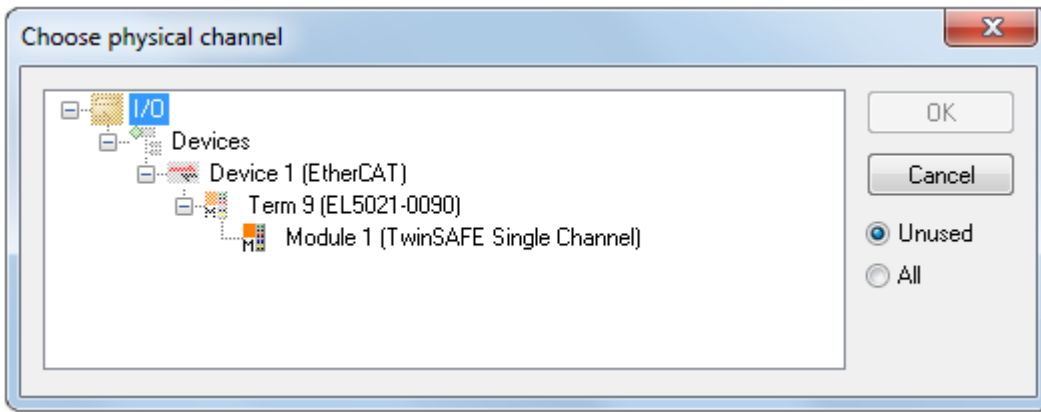


Fig. 164: Creating a link to TwinSAFE SC terminal

The CRC to be used can be selected or a free CRC can be entered under the Connection tab of the alias device.

Entry Mode	Used CRCs
TwinSAFE SC CRC 1 master	0x17B0F
TwinSAFE SC CRC 2 master	0x1571F
TwinSAFE SC CRC 3 master	0x11F95
TwinSAFE SC CRC 4 master	0x153F1
TwinSAFE SC CRC 5 master	0x1F1D5
TwinSAFE SC CRC 6 master	0x1663B
TwinSAFE SC CRC 7 master	0x1B8CD
TwinSAFE SC CRC 8 master	0x1E1BD

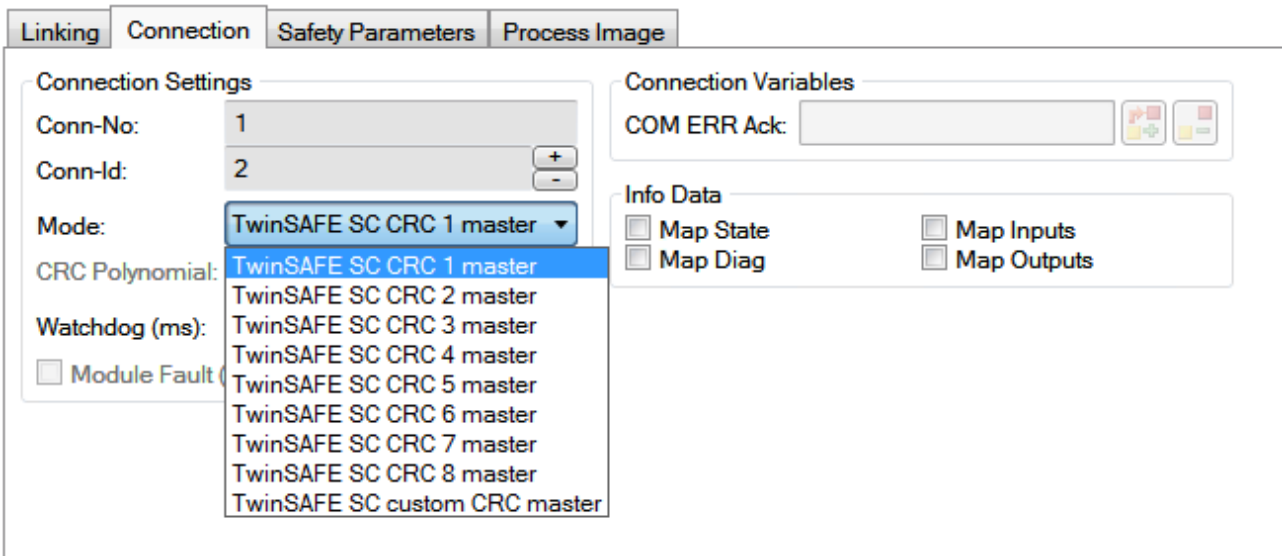


Fig. 165: Selecting a free CRC

These settings must match the settings in the CoE objects of the TwinSAFE SC component. The TwinSAFE SC component initially makes all available process data available. The *Safety Parameters* tab typically contains no parameters. The process data size and the process data themselves can be selected under the *Process Image* tab.

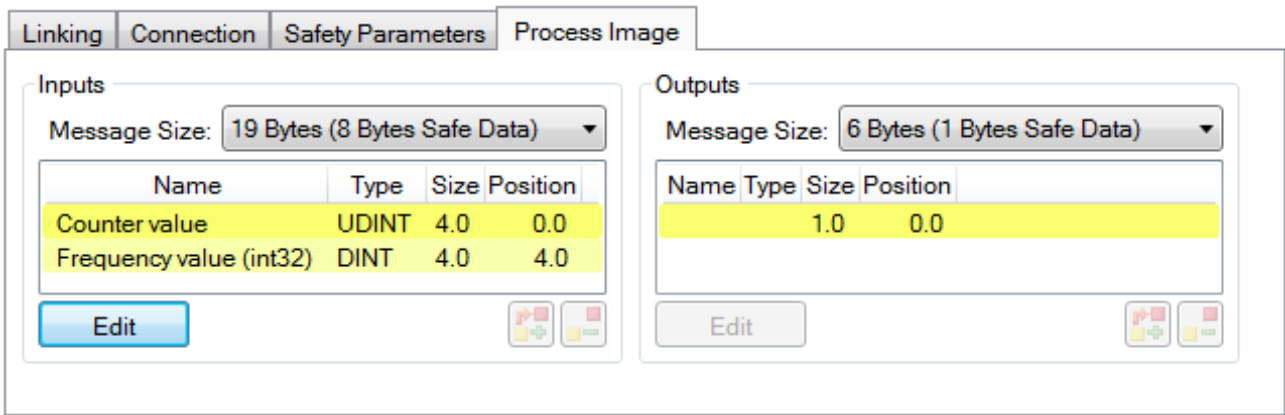


Fig. 166: Selecting the process data size and the process data

The process data (defined in the ESI file) can be adjusted to user requirements by selecting the *Edit* button in the dialog *Configure I/O element(s)*.

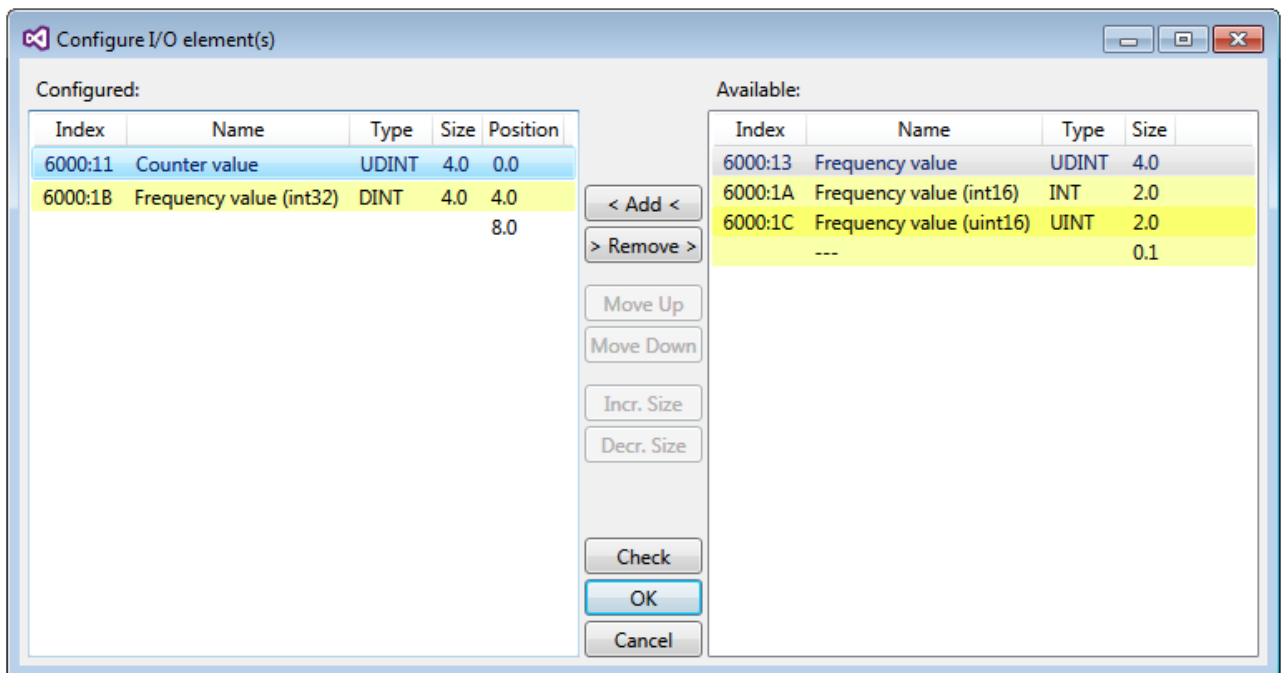


Fig. 167: Selection of the process data

The safety address together with the CRC must be entered on the TwinSAFE SC slave side. This is done via the CoE objects under *TSC settings* of the corresponding TwinSAFE SC component (here, for example, EL5021-0090, 0x8010: 01 and 0x8010: 02). The address set here must also be set in the *alias device* as *FSoE* address under the *Linking* tab.

Under the object 0x80n0:02 Connection Mode the CRC to be used is selected or a free CRC is entered. A total of 8 CRCs are available. A free CRC must start with 0x00ff in the high word.

8010:0	TSC Settings	RW	> 2 <
8010:01	Address	RW	0x0000 (0)
8010:02	Connection Mode	RW	TwinSAFE SC CRC1 master (97039)

Fig. 168: CoE objects 0x8010:01 and 0x8010:02

● Object *TSC Settings*

i Depending on the terminal, the index designation of the configuration object *TSC Settings* can vary. Example:

- EL3214-0090 and EL3314-0090, TSC Settings, Index 8040
- EL5021-0090, TSC Settings, Index 8010
- EL6224-0090, TSC Settings, Index 800F

Fig. 169: Entering the safety address and the CRC

● TwinSAFE SC connections

i If several TwinSAFE SC connections are used within a configuration, a different CRC must be selected for each TwinSAFE SC connection.

5.8 TwinSAFE SC process data EL3214-0090

The EL3214-0090 transmits the following process data to the TwinSAFE logic:

Index	Name	Type	Size
6000:11	RTD Module 1.Value	INT	2.0
6010:11	RTD Module 2.Value	INT	2.0
6020:11	RTD Module 3.Value	INT	2.0
6030:11	RTD Module 4.Value	INT	2.0

The process data of all four channels are transmitted by default. Via the „Process Image“ tab, the other data types of the frequency value can be selected or completely deselected in the Safety Editor.

Depending on the TwinCAT 3.1 version, process data can be renamed automatically when linking to the Safety Editor.

5.9 CoE object description EL32xx-00xx

● EtherCAT XML Device Description

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

● **Parameterization via the CoE list (CAN over EtherCAT)**

i The terminal is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE information when using/manipulating the CoE parameters: - Keep a startup list if components have to be replaced - Differentiation between online/offline dictionary, existence of current XML description - Use "CoE reload" for resetting changes

● **Relevant objects**

i The object description refers to the analog input terminals for Pt100 (RTD) in 1 to 8 channel versions. Observe the indices with regard to the objects relevant for the respective terminal (channel dependent).

Introduction

The CoE overview contains objects for different intended applications:

- [Objects required for parameterization \[► 159\]](#) and [profile-specific objects \[► 166\]](#) required during commissioning
- [Objects for indicating internal settings \[► 162\]](#) (may be fixed)

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

Objects for commissioning

Index 1011 Restore default parameters

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

Index 80n0 RTD settings for 0 ≤ n ≤ 7 (Ch. 1 - 8)

● **The filter properties are set only via index 0x8000:15**

i The filter frequencies are set for all channels of the EL32xx terminals centrally via index 0x8000:15 (channel 1).
The corresponding indices 8010:15 of the EL3202, EL3204 and EL3208 have no parameterization function!

Index (hex)	Name	Meaning	Data type	Flags	Default value
80n0:0	RTD Settings	Maximum subindex	UINT8	RO	0x1B (27 _{dec})
80n0:01	Enable user scale [►_153]	User scale is active.	BOOLEAN	RW	0x00 (0 _{dec})
80n0:02	Presentation [►_150]	0: Signed presentation 1: Absolute value with MSB as sign Signed amount representation 2: High resolution (1/100 C°) [default for EL320x-00x0 high-precision terminals, see note [►_20] !]	BIT3	RW	0x00 (0 _{dec})
80n0:05	Siemens bits [►_151]	The S5 bits are superimposed on the three low-order bits (value 60n0:11 [►_167]) Bit 0 = 1 ("overrange" or "underrange") Bit 1 (not used) Bit 2 (not used)	BOOLEAN	RW	0x00 (0 _{dec})
80n0:06*	Enable filter [►_151]	Enable filter, which makes PLC-cycle-synchronous data exchange unnecessary	BOOLEAN	RW	0x00 (0 _{dec})
80n0:07*	Enable limit 1 [►_153]	The status bits are set in relation to Limit 1	BOOLEAN	RW	0x00 (0 _{dec})
80n0:08*	Enable limit 2 [►_153]	The status bits are set in relation to Limit 2	BOOLEAN	RW	0x00 (0 _{dec})
8010:09*	Enable automatic calibration	A calibration is cyclically started. (optional)	BOOLEAN	RW	0x00 (0 _{dec})
80n0:0A	Enable user calibration [►_153]	Enabling of the user calibration	BOOLEAN	RW	0x00 (0 _{dec})
80n0:0B	Enable vendor calibration [►_153]	Enabling of the vendor calibration	BOOLEAN	RW	0x01 (1 _{dec})
80n0:11	User scale offset [►_153]	User scaling offset	INT16	RW	0x0000 (0 _{dec})
80n0:12	User scale gain [►_153]	This is the user scaling gain. The gain is represented in fixed-point format, with the factor 2 ⁻¹⁶ . The value 1 corresponds to 65535 (0x00010000).	INT32	RW	0x00010000 (65536 _{dec})
80n0:13*	Limit 1 [►_153]	First limit value for setting the status bits (resolution 0.1 °C)	INT16	RW	0x0000 (0 _{dec})
80n0:14*	Limit 2 [►_153]	Second limit value for setting the status bits (resolution 0.1 °C)	INT16	RW	0x0000 (0 _{dec})
80n0:15	Filter settings [►_151]	This object determines the digital filter settings, if it is active via Enable filter (index 0x80n0:06 [►_159]). The possible settings are sequentially numbered. 0: 50 Hz 1: 60 Hz 2: 100 Hz 3: 500 Hz 4: 1 kHz 5: 2 kHz 6: 3.75 kHz 7: 7.5 kHz 8: 15 kHz 9: 30 kHz 10: 5 Hz 11: 10 Hz	UINT16	RW	0x0000 (0 _{dec})
80n0:17	User calibration offset [►_153]	User offset calibration	INT16	RW	0x0000 (0 _{dec})
80n0:18	User calibration gain [►_153]	User gain compensation	UINT16	RW	0xFFFF (65535 _{dec})

*) not for EL3208 and EL3214-00x0

Index (hex)	Name	Meaning	Data type	Flags	Default value
80n0:19	RTD element [► 21]	<p>RTD element</p> <p>0: Pt100 (for high-precision terminals see note [► 20]!)</p> <p>1: Ni100 -60°C to 250°</p> <p>2: Pt1000 -200°C to 850°C</p> <p>3: Pt500 -200°C to 850°C</p> <p>4: Pt200 -200°C to 850°C</p> <p>5: Ni1000 -60°C to 250°C</p> <p>6: Ni1000 TK1500, 100°C: 1500 Ohm (-30 to 160°C)</p> <p>7: Ni120 -60°C to 320°C</p> <p>8: Output in Ohm Resolution 1/16 Ohm (0 to 4096 Ohm)***</p> <p>9: Output in Ohm Resolution 1/64 Ohm (0 to 1024 Ohm)***</p> <p>10-32:</p> <p>KT100/110/130/210/230 KTY10/11/13/16/19 KTY81/82-110,120,150 (-50...150°C) KTY81-121 KTY81-122 KTY81-151 KTY81-152 KTY81/82-210,220,250 KTY81-221 KTY81-222 KTY81-251 KTY81-252 KTY83-110,120,150 (-50...175°C) KTY83-121 KTY83-122 KTY83-151 KTY83-152 KTY84-130,150 (-40...300°C) KTY84-151 KTY21/23-6 (-50...150°C) KTY1x-5 KTY1x-7 KTY21/23-5 KTY21/23-7</p> <p>***) please note chapter "function [► 19]!"</p>	UINT16	RW	0x0000 (0 _{dec})
80n0:1A	Connection technology [► 20]	<p>Connection technology</p> <p>0: Two-wire connection (EL3214-0090 from firmware 02)</p> <p>1*: Three-wire connection</p> <p>2*: Four-wire connection [default for EL320x-00x0 high-precision terminals, see note [► 20]!]</p> <p>3*: not connected</p>	UINT16	RW	0x0000 (0 _{dec})
80n0:1B	Wire calibration 1/32 Ohm [► 21]	Calibration of the supply lines	INT16	RW	0x0000 (0 _{dec})

*) not for EL3208 and EL3214-00x0

Index 8040 TSC Settings (only EL3214-0090)

Index (hex)	Name	Meaning	Data type	Flags	Default value
8040:0	TSC Settings [▶ 154]	Max. ubindex	UINT8	RO	0x02 (2 _{dec})
8040:01	Address	TwinSAFE SC Address	UINT16	RW	0x0000 (0 _{dec})
8040:02	Connection Mode	Selection of TwinSAFE SC CRC	UINT32	RO	0x00000000 (0 _{dec})

Complete overview**Standard objects (0x1000-0x1FFF)**

The standard objects have the same meaning for all EtherCAT slaves.

Index 1000 Device type

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

Index 1008 Device name

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

Index 1009 Hardware version

Index (hex)	Name	Meaning	Data type	Flags	Default value
1009:0	Hardware version	Hardware version of the EtherCAT slave	STRING	RO	00

Index 100A Software version

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

Index 1018 Identity

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

Index 10F0 Backup parameter handling

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

Index 1604 TSC RxPDO-Map Master Message (only EL3214-0090)

Index (hex)	Name	Meaning	Data type	Flags	Default value
1604:0	TSC RxPDO-Map Master Message	PDO Mapping RxPDO	UINT8	RO	0x04 (4 _{dec})
1604:01	SubIndex 001	1. PDO Mapping entry (object 0x7040 (TSC Master Frame Elements), entry 0x01 (TSC__Master Cmd))	UINT32	RO	0x7040:01, 8
1604:02	SubIndex 002	2. PDO Mapping entry (8 bits align)	UINT32	RO	0x0000:00, 8
1604:03	SubIndex 003	3. PDO Mapping entry (object 0x7040 (TSC Master Frame Elements), entry 0x03 (TSC__Master CRC_0))	UINT32	RO	0x7040:03, 16
1604:04	SubIndex 004	4. PDO Mapping entry (object 0x7040 (TSC Master Frame Elements), entry 0x02 (TSC__Master Con-nID))	UINT32	RO	0x7040:02, 16

Index 1A0n TxPDO Map for 0 ≤ n ≤ 3 (Ch. 1 - 4) [for EL3201, EL3202, EL3204-00x0]

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A0n:0	TxPDO Map Ch.1	PDO Mapping TxPDO 1	UINT8	RW	0x09 (9 _{dec})
1A0n:01	SubIndex 001	1. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x01 (Underrange))	UINT32	RW	0x60n0:01, 1
1A0n:02	SubIndex 002	2. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x02 (Overrange))	UINT32	RW	0x60n0:02, 1
1A0n:03	SubIndex 003	3. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x03 (Limit 1))	UINT32	RW	0x60n0:03, 2
1A0n:04	SubIndex 004	4. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x05 (Limit 2))	UINT32	RW	0x60n0:05, 2
1A0n:05	SubIndex 005	5. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x07 (Error))	UINT32	RW	0x60n0:07, 1
1A0n:06	SubIndex 006	6. PDO Mapping entry (7 bits align)	UINT32	RW	0x0000:00, 7
1A0n:07	SubIndex 007	7. PDO Mapping entry (object 0x180n (TxPDO-Par Ch.1), entry 0x07 (TxPDO-State))	UINT32	RW	0x180n:07, 1
1A0n:08	SubIndex 008	8. PDO Mapping entry (object 0x180n (TxPDO-Par Ch.1), entry 0x09 (TxPDO-Toggle))	UINT32	RW	0x180n:09, 1
1A0n:09	SubIndex 009	9. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x11 (Value))	UINT32	RW	0x60n0:11, 16

Index 1A0n TxPDO Map for 0 ≤ n ≤ 3 (Ch. 1 - 4) [for EL3214-00x0]

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A0n:0	TxPDO Map Ch.1	PDO Mapping TxPDO 1	UINT8	RW	0x09 (9 _{dec})
1A0n:01	SubIndex 001	1. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x01 (Underrange))	UINT32	RW	0x60n0:01, 1
1A0n:02	SubIndex 002	2. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x02 (Overrange))	UINT32	RW	0x60n0:02, 1
1A0n:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RW	0x0000:00, 4
1A0n:04	SubIndex 004	4. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x07 (Error))	UINT32	RW	0x60n0:07, 1
1A0n:05	SubIndex 005	5. PDO Mapping entry (7 bits align)	UINT32	RW	0x0000:00, 7
1A0n:06	SubIndex 006	6. PDO Mapping entry (object 0x180n (TxPDO-Par Ch.1), entry 0x07 (TxPDO-State))	UINT32	RW	0x180n:07, 1
1A0n:07	SubIndex 007	7. PDO Mapping entry (object 0x180n (TxPDO-Par Ch.1), entry 0x09 (TxPDO-Toggle))	UINT32	RW	0x180n:09, 1
1A0n:08	SubIndex 008	8. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x11 (Value))	UINT32	RW	0x60n0:11, 16

Index 1A0n TxPDO Map for $0 \leq n \leq 7$ (Ch. 1 - 8) [for EL3208]

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A0n:0	TxPDO Map Ch.1	PDO Mapping TxPDO 1	UINT8	RW	0x09 (9 _{dec})
1A0n:01	SubIndex 001	1. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x01 (Underrange))	UINT32	RW	0x60n0:01, 1
1A0n:02	SubIndex 002	2. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x02 (Overrange))	UINT32	RW	0x60n0:02, 1
1A0n:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RW	0x0000:00, 4
1A0n:04	SubIndex 004	4. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x07 (Error))	UINT32	RW	0x60n0:07, 1
1A0n:05	SubIndex 005	5. PDO Mapping entry (7 bits align)	UINT32	RW	0x0000:00, 7
1A0n:06	SubIndex 006	6. PDO Mapping entry (object 0x180n (TxPDO-Par Ch.1), entry 0x07 (TxPDO-State))	UINT32	RW	0x180n:07, 1
1A0n:07	SubIndex 007	7. PDO Mapping entry (object 0x180n (TxPDO-Par Ch.1), entry 0x09 (TxPDO-Toggle))	UINT32	RW	0x180n:09, 1
1A0n:08	SubIndex 008	8. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x11 (Value))	UINT32	RW	0x60n0:11, 16

Index 1A04 TSC TxPDO-Map Slave Message (only EL3214-0090)

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A04:0	TSC TxPDO-Map Slave Message	PDO Mapping TxPDO	UINT8	RW	0x0A (10 _{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6040 (TSC Slave Frame Elements), entry 0x01 (TSC__Slave Cmd))	USINT8	RW	0x6040:01, 8
1A04:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (ENC Inputs), entry 0x11 (Counter value))	INT16	RW	0x6000:11, 16
1A04:03	SubIndex 003	3. PDO Mapping entry (object 0x6040 (TSC Slave Frame Elements), entry 0x03 (TSC__Slave CRC_0))	UINT16	RW	0x6040:03, 16
1A04:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (ENC Inputs), entry 0x11 (Counter value))	INT16	RW	0x6010:11, 16
1A04:05	SubIndex 005	5. PDO Mapping entry (object 0x6040 (TSC Slave Frame Elements), entry 0x04 (TSC__Slave CRC_1))	UINT16	RW	0x6040:04, 16
1A04:06	SubIndex 006	6. PDO Mapping entry (object 0x6020 (ENC Inputs), entry 0x11 (Counter value))	INT16	RW	0x6020:11, 16
1A04:07	SubIndex 007	7. PDO Mapping entry (object 0x6040 (TSC Slave Frame Elements), entry 0x05 (TSC__Slave CRC_2))	UINT16	RW	0x6040:05, 16
1A04:08	SubIndex 008	8. PDO Mapping entry (object 0x6030 (ENC Inputs), entry 0x11 (Counter value))	INT16	RW	0x6030:11, 16
1A04:09	SubIndex 009	9. PDO Mapping entry (object 0x6040 (TSC Slave Frame Elements), entry 0x06 (TSC__Slave CRC_3))	UINT16	RW	0x6040:06, 16
1A04:0A	SubIndex 010	10. PDO Mapping entry (object 0x6040 (TSC Slave Frame Elements), entry 0x02 (TSC__Slave Con-nID))	UINT16	RW	0x6040:02, 16

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x00 (0 _{dec})

Index 1C13 TxPDO assign***

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x08 (8 _{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A00 (6656 _{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A01 (6657 _{dec})
1C13:03	Subindex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A02 (6658 _{dec})
1C13:04	Subindex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A03 (6659 _{dec})
1C13:05	Subindex 005	5. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A04 (6660 _{dec})
1C32:06	Subindex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A05 (6661 _{dec})
1C13:07	Subindex 007	7. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A06 (6662 _{dec})
1C13:08	Subindex 008	8. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A07 (6663 _{dec})

***) for EL3201: subindex x01, for EL3202: subindex x01, x02 and so on.

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x07 (7 _{dec})
1C33:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> • 0: Free Run • 1: Synchronous with SM 3 event (no outputs available) • 2: DC - Synchronous with SYNC0 Event • 3: DC - Synchronous with SYNC1 Event • 34: Synchronous with SM 2 event (outputs available) 	UINT16	RW	0x0000 (0 _{dec})
1C33:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC-Mode: SYNC0/SYNC1 Cycle Time 	UINT32	RW	0x00000000 (0 _{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RW	0x00000000 (0 _{dec})
1C33:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> • Bit 0: free run is supported • Bit 1: synchronous with SM 2 event is supported (outputs available) • Bit 1: synchronous with SM 3 event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: input shift through local event (outputs available) • Bit 4-5 = 10: input shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C33:08) 	UINT16	RO	0x8007 (32775 _{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00000000 (0 _{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:08	Command	<ul style="list-style-type: none"> • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started <p>The entries 0x1C33:03, 0x1C33:06, 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p>	UINT16	RW	0x0000 (0 _{dec})
1C33:09	Delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 _{dec})
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0 _{dec})
1C33:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0 _{dec})
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0 _{dec})

Profile-specific objects (0x6000-0xFFFF)

The profile-specific objects have the same meaning for all EtherCAT slaves that support the profile 5001.

Index 60n0 RTD Inputs for $0 \leq n \leq 7$ (Ch. 1 - 8)

Index (hex)	Name	Meaning	Data type	Flags	Default value
60n0:0	RTD Inputs	Maximum subindex	UINT8	RO	0x11 (17 _{dec})
60n0:01	Underrange	The measuring range is undershot.	BOOLEAN	RO	0x00 (0 _{dec})
60n0:02	Overrange	The measuring range is overshoot. ("open circuit" detection if "error" [index 0x60n0:07] is set	BOOLEAN	RO	0x00 (0 _{dec})
60n0:03*	Limit 1	Limit value monitoring 0: not active 1: Value is larger than the limit value 2: Value is smaller than the limit value 3: Value is equal to the limit value	BIT2	RO	0x00 (0 _{dec})
60n0:05*	Limit 2	Limit value monitoring 0: not active 1: Value is larger than the limit value 2: Value is smaller than the limit value 3: Value is equal to the limit value	BIT2	RO	0x00 (0 _{dec})
60n0:07	Error	The error bit is set if the data is invalid.	BOOLEAN	RO	0x00 (0 _{dec})
60n0:0F	TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).	BOOLEAN	RO	0x00 (0 _{dec})
60n0:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0 _{dec})
60n0:11	Value	The analog input data	INT16	RO	0x0000 (0 _{dec})

*) not for EL3208 and EL3214-00x0

Index 6040 TSC Slave Frame Elements (only EL3214-0090)

Index (hex)	Name	Meaning	Data type	Flags	Default value
6040:0	TSC Slave Frame Elements [▶_154]	Max. subindex	UINT8	RO	0x06 (7 _{dec})
6040:01	TSC__Slave Cmd	reserved	UINT8	RO	0x00 (0 _{dec})
6040:02	TSC__Slave ConnID	reserved	UINT16	RO	0x0000 (0 _{dec})
6040:03	TSC__Slave CRC_0	reserved	UINT16	RO	0x0000 (0 _{dec})
6040:04	TSC__Slave CRC_1	reserved	UINT16	RO	0x0000 (0 _{dec})
6040:05	TSC__Slave CRC_2	reserved	UINT16	RO	0x0000 (0 _{dec})
6040:06	TSC__Slave CRC_3	reserved	UINT16	RO	0x0000 (0 _{dec})

Index 7040 TSC Master Frame Elements (only EL3214-0090)

Index (hex)	Name	Meaning	Data type	Flags	Default value
7040:0	TSC Master Frame Elements	Max. subindex	UINT8	RO	0x03 (3 _{dec})
7040:01	TSC__Master Cmd	reserved	UINT8	RO	0x00 (0 _{dec})
7040:02	TSC__Master ConnID	reserved	UINT16	RO	0x0000 (0 _{dec})
7040:03	TSC__Master CRC_0	reserved	UINT16	RO	0x0000 (0 _{dec})

Index 80nE RTD Internal data for $0 \leq n \leq 7$ (Ch. 1 - 8)

Index (hex)	Name	Meaning	Data type	Flags	Default value
80nE:0	RTD Internal data	Maximum subindex	UINT8	RO	0x04 (4 _{dec})
80nE:01	ADC raw value 1	ADC raw value 1	INT32	RO	0x00000000 (0 _{dec})
80nE:02	Resistor 1	Resistance 1 (measured value of resistance sensor, resolution 1/32 Ohm)	UINT16	RO	0x0000 (0 _{dec})
80nE:03*	ADC raw value 2 (RL)	ADC raw value 2 (RL)	INT32	RO	0x00000000 (0 _{dec})
80nE:04*	Resistor 2 (RL)	Resistance 2 (RL) (measured value of the supply line, resolution 1/32 Ohm)	UINT16	RO	0x0000 (0 _{dec})

*) not for EL3208

Index 80nF RTD Vendor data for $0 \leq n \leq 3$ (Ch. 1 - 4) [not for EL3208 and EL3214-00x0]

Index (hex)	Name	Meaning	Data type	Flags	Default value
80nF:0	RTD Vendor data	Maximum subindex	UINT8	RO	0x06 (6 _{dec})
80nF:01	Calibration offset	Manufacturer calibration offset	INT16	RW	0x0000 (0 _{dec})
80nF:02	Calibration gain	Manufacturer calibration gain	UINT16	RW	0x9E50 (40528 _{dec})
80nF:03	Calibration offset RL	Manufacturer calibration offset (input RL)	INT16	RW	0x0000 (0 _{dec})
80nF:04	Calibration gain RL	Manufacturer calibration gain (input RL)	UINT16	RW	0x9E50 (40528 _{dec})
80nF:05	Calibration offset 4-wire	Manufacturer calibration offset (4-wire connection technology)	INT16	RW	0x0000 (0 _{dec})
80nF:06	Calibration gain 4-wire	Manufacturer calibration gain (4-wire connection technology)	UINT16	RW	0x9E50 (40528 _{dec})

Index 80nF RTD Vendor data for $0 \leq n \leq 7$ (Ch. 1 - 8) [EL3208 and EL3214-00x0 only]

Index (hex)	Name	Meaning	Data type	Flags	Default value
80nF:0	RTD Vendor data	Maximum subindex	UINT8	RO	-
80nF:01	Calibration offset	Manufacturer calibration offset	INT16	RW	-
80nF:02	Calibration gain	Manufacturer calibration gain	UINT16	RW	-
80nF:03	Calibration offset Pt1000	Manufacturer calibration offset Pt1000	INT16	RW	-
80nF:04	Calibration gain Pt1000	Manufacturer calibration gain Pt1000	UINT16	RW	-
80nF:05*	Calibration gain RL	Manufacturer calibration gain (input RL)	UINT16	RW	-
80nF:06*	Calibration gain RL	Manufacturer calibration gain (input RL)	UINT16	RW	-

*) not for EL3208

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default value
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2 _{dec})
F000:01	Module index distance	Index spacing of the objects of the individual channels	UINT16	RO	0x0010 (16 _{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0004 (4 _{dec})

Index F008 Code word

Index (hex)	Name	Meaning	Data type	Flags	Default value
F008:0	Code word [► 154]	currently reserved	UINT32	RW	0x00000000 (0 _{dec})

Index F010 Module list [for {n=1} (1-channel) to {n=1,...,n=8} (8-channel)]

Index (hex)	Name	Meaning	Data type	Flags	Default value
F010:0	Module list	Maximum subindex	UINT8	RO	0x08 (n _{dec})
F010:0n	Subindex 00n	Profile 320	INT32	RO	0x00000140 (320 _{dec})

5.10 CoE object description EL3204-0200



EtherCAT XML Device Description

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

i Parameterization via the CoE list (CAN over EtherCAT)

The terminal is parameterized via the CoE - Online tab (double-click on the respective object) or via the Process Data tab (allocation of PDOs). Please note the following general CoE information when using/manipulating the CoE parameters: - Keep a startup list if components have to be replaced - Differentiation between online/offline dictionary, existence of current XML description - Use "CoE reload" for resetting changes

Introduction

The CoE overview contains objects for different intended applications:

- [Objects required for parameterization \[▶ 169\]](#) and [profile-specific objects \[▶ 176\]](#) required during commissioning
- [Objects for indicating internal settings \[▶ 172\]](#) (may be fixed)

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

Objects for commissioning

Index 1011 Restore default parameters

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

Index 80n0 RTD settings for 0 ≤ n ≤ 3 (Ch. 1 - 4)

i The filter properties are set only via index 0x8000:15

The filter frequencies are set for all channels of the EL3204-0200 terminals centrally via index 0x8000:15 (channel 1). The corresponding indices 0x80n0:15 have no parameterization function!

Index (hex)	Name	Meaning	Data type	Flags	Default value
80n0:0	RTD Settings	Maximum subindex	UINT8	RO	0x1B (27 _{dec})
80n0:01	Enable user scale	User scale is active.	BOOLEAN	RW	0x00 (0 _{dec})
80n0:02	Process data Presentation	permitted values 0: Signed 1: Absolute MSB sign 2: High resolution (1/100°C)	BIT3	RW	0x00 (0 _{dec})
80n0:05	Siemens bits	The S5 bits are superimposed on the three low-order bits (value 0x60n0:11 [▶ 176]) Bit 0 = 1 ("overrange" or "underrange") Bit 1 (not used) Bit 2 (not used)	BOOLEAN	RW	0x00 (0 _{dec})
80n0:06	Enable filter	Enable filter, which makes PLC-cycle-synchronous data exchange unnecessary	BOOLEAN	RW	0x00 (0 _{dec})
80n0:0A	Enable user calibration	Enabling of the user calibration	BOOLEAN	RW	0x00 (0 _{dec})
80n0:0B	Enable vendor calibration	Enabling of the vendor calibration	BOOLEAN	RW	0x01 (1 _{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default value
80n0:11	User scale offset	User scaling offset	INT16	RW	0x0000 (0 _{dec})
80n0:12	User scale gain	This is the user scaling gain. The gain is represented in fixed-point format, with the factor 2 ⁻¹⁶ . The value 1 corresponds to 65535 (0x00010000).	INT32	RW	0x00010000 (65536 _{dec})
80n0:15	Filter settings	This object determines the digital filter settings, if it is active via Enable filter (index 0x80n0:06 [▶ 169]). The possible settings are sequentially numbered. permitted values: 0: 50 Hz 1: 60 Hz 2: 100 Hz 3: 500 Hz 4: 1 kHz 5: 2 kHz 6: 3.75 kHz 7: 7.5 kHz 8: 15 kHz 9: 30 kHz 10: 5 Hz 11: 10 Hz	UINT16	RW	0x0000 (0 _{dec})
80n0:17	User calibration offset	User offset calibration	INT16	RW	0x0000 (0 _{dec})
80n0:18	User calibration gain	User gain compensation	UINT16	RW	0xFFFF (65535 _{dec})
80n0:19	RTD element	permitted values: 0: Pt1000 (-200...850°C) 2: NI1000 100 °C (TK5000, at 100 °C 1500 Ohm, -30 to 160 °C) 3: NI1000 (-60...250°C) 4: NTC 5k (0°C: 16325 Ohm, -40...100°C) 5: NTC 10k (0°C: 32650 Ohm, -30...150°C) 6: NTC 10k (0°C: 29490 Ohm, -30...150°C) 7: NTC 10k (0°C: 27280 Ohm, -40...110°C, as for KL3204-0030) 8: NTC 20k (0°C: 70200 Ohm, -20...150°C) 9: Fet (-30...150°C) 101: Resolution 1/10 Ohm (0 to 6553.5 Ohm) 102: Resolution 1/10 Ohm (0 to 65535 Ohm) 103: Resolution 1/10 Ohm (0 to 240 kOhm) 110: RTD table (for further parameters see 0x80n1 [▶ 170]) 111: B-parameter equation (for further parameters see 0x80n2 [▶ 171]) 112: DIN IEC 60751 equation (for further parameters see 0x80n3 [▶ 171]) 113: Steinhart-Hart equation (for further parameters see 0x80n4 [▶ 172])	UINT16	RW	0x0000 (0 _{dec})
80n0:1A	Connection technology	permitted values 0: Two-wire connection 3: n. c. (channel deactivated)	UINT16	RW	0x0000 (0 _{dec})
80n0:1B	Wire calibration	Offset value for the calibration of the supply lines [1/10] Ohm	INT16	RW	0x0000 (0 _{dec})

Index 80n1 RTD Table

The standard values in the following table show by way of example the conversion values of a Pt1000 within the range from 0 °C to 140 °C with a resolution of 1/10 Ohm

Index (hex)	Name	Meaning	Data type	Flags	Default value
80n1:0	RTD Table	The sensor characteristic can be programmed with this table	UINT8	RO	105
80n1:01	Distance in Degrees	The value set describes the distance of the entries in °C	UINT16	RW	10
80n1:02	Amount of Entries	The value entered describes the number of entries with which the dependence on distance in °C, start temperature of the measuring range and final temperature are displayed according to the equation below Final temperature = start temperature + ((number of entries - 1) * distance in °C)	UINT16	RW	15
80n1:03	Start Temperature	The value entered describes the start of the measuring range Standard value 0 °C	INT16	RW	0
80n1:04	Conversion Factor	1 = 1/10 Ohm per digit in the table entries 10 = 1 Ohm per digit in the table entries 100 = 10 Ohm per digit in the table entries	UINT16	RW	1
80n1:05	Value 1		UINT16	RW	0x2710 (10000 _{dec})
80n1:06	Value 2		UINT16	RW	0x2896 (10390 _{dec})
80n1:07	Value 3		UINT16	RW	0x2A1B (10779 _{dec})
80n1:08	Value 4		UINT16	RW	0x2B9F (11167 _{dec})
80n1:09	Value 5		UINT16	RW	0x2D22 (11554 _{dec})
80n1:0A	Value 6		UINT16	RW	0x2EA3 (11939 _{dec})
80n1:0B	Value 7		UINT16	RW	0x3024 (12324 _{dec})
80n1:0C	Value 8		UINT16	RW	0x31A3 (12707 _{dec})
80n1:0D	Value 9		UINT16	RW	0x3321 (13089 _{dec})
80n1:0E	Value 10		UINT16	RW	0x349E (13470 _{dec})
80n1:0F	Value 11		UINT16	RW	0x361A (13850 _{dec})
80n1:10	Value 12		UINT16	RW	0x3795 (14229 _{dec})
80n1:11	Value 13		UINT16	RW	0x390E (14606 _{dec})
80n1:13	Value 14		UINT16	RW	0x3A87 (14983 _{dec})
80n1:14	Value 15		UINT16	RW	0x3BF5 (15358 _{dec})
80n1:15	Value 16		UINT16	RW	0x0000 (0 _{dec})
...
80n1:69	Value 100		UINT16	RW	0x0000 (0 _{dec})

Index 80n2 RTD B-parameter equation for $0 \leq n \leq 3$ (Ch. 1 - 4)

Index (hex)	Name	Meaning	Data type	Flags	Default value
80n2:0	RTD B-Parameter Equation		UINT8	RO	0x05 (5 _{dec})
80n2:01	Min Temperature	Start of the measuring range in °C.	INT16	RW	-30
80n2:02	Max Temperature	End of the measuring range in °C.	INT16	RW	150
80n2:03	Rn	Resistance value at Tn. In most cases = 25 °C.	INT16	RW	1000
80n2:04	Tn	Reference temperature for Rn [°C]	INT16	RW	25
80n2:05	B-parameter	B-parameter [Kelvin]	REAL32	RW	3710

Index 80n3 RTD DIN EN 60751 equation for $0 \leq n \leq 3$ (Ch. 1 - 4) 1 - 4)

Index (hex)	Name	Meaning	Data type	Flags	Default value
80n3:0	RTD DIN EN 60751 equation	Standard parameter for Pt1000 within the range 0 to 850 °C.	UINT8	RO	0x06 (6 _{dec})
80n3:01	Min Temperature	Start of the measuring range in °C.	INT16	RW	0
80n3:02	Max Temperature	End of the measuring range in °C.	INT16	RW	850
80n3:03	R0	see equation [► 144]	UINT16	RW	1000
80n3:04	A parameter	see equation [► 144]	REAL32	RW	3.908E-3
80n3:05	B parameter	see equation [► 144]	REAL32	RW	-5.775E-7
80n3:06	C parameter	see equation [► 144]	REAL32	RW	-4.183E-12

Index 80n4 RTD Steinhart-Hart equation for $0 \leq n \leq 3$ (Ch. 1 - 4)

Index (hex)	Name	Meaning	Data type	Flags	Default value
80n4:0	RTD Steinhart Hart Equation	Standard values for an NTC10K with 32650 Ohm at 0 °C: C1 default = 1.12119E-03 C2 default = 2.35346E-04 C3 default = 0 C4 default = 8.34620E-08	UINT8	RO	0x06 (6 _{dec})
80n4:01	Min Temperature	Start of the measuring range in °C.	INT16	RW	-30
80n4:02	Max Temperature	End of the measuring range in °C.	INT16	RW	150
80n4:03	C1	see equation [► 142]	REAL32	RW	1.1211E-03
80n4:04	C2	see equation [► 142]	REAL32	RW	2.35346E-04
80n4:05	C3	see equation [► 142]	REAL32	RW	0
80n4:06	C4	see equation [► 142]	REAL32	RW	8.24620E-08

Index 80nE RTD Internal Data for $0 \leq n \leq 3$ (Ch. 1 - 4)

Index (hex)	Name	Meaning	Data type	Flags	Default value
80nE:0	RTD Internal Data	Max. subindex	UINT8	RO	
80nE:01	ADC raw value	ADC raw value.	INT32	RO	
80nE:02	Resistor (1/10 Ohm)	resulting measured resistance	UINT32	RO	

Index 80nF RTD Vendor Data for $0 \leq n \leq 3$ (Ch. 1 - 4)

Index (hex)	Name	Meaning	Data type	Flags	Default value
80nF:0	RTD Vendor Data	Max. subindex	UINT8	RO	0x03 (3 _{dec})
80nF:01	Calibration offset	Calibration offset	INT32	RW	0x00000DAC (3500 _{dec})
80nF:02	Calibration gain	Gain value for measuring range 0 to 5000 Ohm	UINT16	RW	0x3D68 (15720 _{dec})
80nF:03	Calibration gain ext.	Gain value for measuring range 5 to 240 kOhm	UINT16	RW	0x05A0 (1440 _{dec})

Complete overview**Standard objects**

The standard objects have the same meaning for all EtherCAT slaves.

Index 1000 Device type

Index (hex)	Name	Meaning	Data type	Flags	Default value
1000:0	Device type	Device type of the EtherCAT slave: the Lo-Word contains the CoE profile used (5001). The Hi-Word contains the module profile according to the modular device profile.	UINT32	RO	0x01401389 (20976521 _{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default value
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL3204-0200

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default value
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4 _{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2 _{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x0C843052 (209989714 _{dec})
1018:03	Revision	Revision number of the EtherCAT slave; the low word (bit 0-15) indicates the special terminal number, the high word (bit 16-31) refers to the device description	UINT32	RO	0x00000000 (0 _{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	0x00000000 (0 _{dec})

Index 10F0 Backup parameter handling

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

Index 180n RTD TxPDO-Par for 0 ≤ n ≤ 3 (Ch. 1 - 4)

Index (hex)	Name	Meaning	Data type	Flags	Default value
180n:0	RTD TxPDO-Par	PDO parameter TxPDO 1	UINT8	RO	0x09 (9 _{dec})
180n:06	Exclude TxPDOs	This entry specifies the TxPDOs (index of TxPDO mapping objects) that must not be transferred together with TxPDO 1	OCTET-STRING[0]	RO	
180n:07	TxPDO State	The TxPDO state is set if it was not possible to correctly read in the associated input data	BOOLEAN	RO	0x00 (0 _{dec})
180n:09	TxPDO Toggle	The TxPDO toggle is toggled with each update the corresponding input data	BOOLEAN	RO	0x00 (0 _{dec})

Index 1A0n RTD TxPDO-Map 0 ≤ n ≤ 3 (Ch. 1 - 4)

Index (hex)	Name	Meaning	Data type	Flags	Default value
1A0n:0	RTD TxPDO-Map	PDO Mapping TxPDO 1	UINT8	RO	0x08 (8 _{dec})
1A0n:01	SubIndex 001	1. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x01 (Underrange))	UINT32	RO	0x60n0:01, 1
1A0n:02	SubIndex 002	2. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x02 (Overrange))	UINT32	RO	0x60n0:02, 1
1A0n:03	SubIndex 003	3. PDO Mapping entry (4 bits align)	UINT32	RO	0x0000:00, 4
1A0n:04	SubIndex 004	4. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x07 (Error))	UINT32	RO	0x60n0:07, 1
1A0n:05	SubIndex 005	5. PDO Mapping entry (7 bits align)	UINT32	RO	0x00n0:00, 7
1A0n:06	SubIndex 006	6. PDO Mapping entry (object 0x180n (RTD TxPDO-Par Ch.1), entry 0x07 (TxPDO State))	UINT32	RO	0x180n:07, 1
1A0n:07	SubIndex 007	7. PDO Mapping entry (object 0x180n (RTD TxPDO-Par Ch.1), entry 0x09 (TxPDO Toggle))	UINT32	RO	0x180n:09, 1
1A0n:08	SubIndex 008	8. PDO Mapping entry (object 0x60n0 (RTD Inputs Ch.1), entry 0x11 (Value))	UINT32	RO	0x60n0:11, 16

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RO	0x00 (0 _{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RO	0x04 (4 _{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A00 (6656 _{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A01 (6657 _{dec})
1C13:03	Subindex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A02 (6658 _{dec})
1C13:04	Subindex 004	4. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RO	0x1A03 (6659 _{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default value
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32 _{dec})
1C33:01	Sync mode	Current synchronization mode: <ul style="list-style-type: none"> • 0: Free Run • 1: Synchronous with SM 3 event (no outputs available) • 2: DC - Synchronous with SYNC0 Event • 3: DC - Synchronous with SYNC1 Event • 34: Synchronous with SM 2 event (outputs available) 	UINT16	RW	0x0000 (0 _{dec})
1C33:02	Cycle time	Cycle time (in ns): <ul style="list-style-type: none"> • Free Run: Cycle time of the local timer • Synchronous with SM 2 event: Master cycle time • DC-Mode: SYNC0/SYNC1 Cycle Time 	UINT32	RW	0x00000000 (0 _{dec})
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:04	Sync modes supported	Supported synchronization modes: <ul style="list-style-type: none"> • Bit 0: free run is supported • Bit 1: synchronous with SM 2 event is supported (outputs available) • Bit 1: synchronous with SM 3 event is supported (no outputs available) • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 01: input shift through local event (outputs available) • Bit 4-5 = 10: input shift with SYNC1 event (no outputs available) • Bit 14 = 1: dynamic times (measurement through writing of 1C33:08) 	UINT16	RO	0xC007 (49159 _{dec})
1C33:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00000000 (0 _{dec})
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:07	Minimum delay time		UINT32	RO	0x00000000 (0 _{dec})
1C33:08	Command	<ul style="list-style-type: none"> • 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started <p>The entries 0x1C33:03, 0x1C33:06, 0x1C33:09 are updated with the maximum measured values. For a subsequent measurement the measured values are reset</p>	UINT16	RW	0x0000 (0 _{dec})
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0 _{dec})
1C33:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0 _{dec})
1C33:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0 _{dec})
1C33:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0 _{dec})
1C33:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0 _{dec})

Profile-specific objects

Index 60n0 RTD Inputs for $0 \leq n \leq 3$ (Ch. 1 – 4)

Index (hex)	Name	Meaning	Data type	Flags	Default value
60n0:0	RTD Inputs	Maximum subindex	UINT8	RO	0x11 (17 _{dec})
60n0:01	Underrange	The measuring range is undershot.	BOOLEAN	RO	0x00 (0 _{dec})
60n0:02	Overrange	The measuring range is overshot.	BOOLEAN	RO	0x00 (0 _{dec})
60n0:07	Error	The error bit is set if the data is invalid.	BOOLEAN	RO	0x00 (0 _{dec})
60n0:0F	TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).	BOOLEAN	RO	0x00 (0 _{dec})
60n0:10	TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	BOOLEAN	RO	0x00 (0 _{dec})
60n0:11	Value	The analog input data	INT16	RO	0x0000 (0 _{dec})

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default value
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2 _{dec})
F000:01	Module index distance	Index spacing of the objects of the individual channels	UINT16	RO	0x0010 (16 _{dec})
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0004 (4 _{dec})

Index F008 Code word

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

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default value
F010:0	Module list	Maximum subindex	UINT8	RW	0x04 (4 _{dec})
F010:01	SubIndex 001	Analog Input Profile for RTD	UINT32	RW	0x00000140 (320 _{dec})
F010:02	SubIndex 002	Analog Input Profile for RTD	UINT32	RW	0x00000140 (320 _{dec})
F010:03	SubIndex 003	Analog Input Profile for RTD	UINT32	RW	0x00000140 (320 _{dec})
F010:04	SubIndex 004	Analog Input Profile for RTD	UINT32	RW	0x00000140 (320 _{dec})

5.11 Status word

The status information for each channel of the EL32xx and EL33xx is transmitted cyclically from the terminal to the EtherCAT Master as process data (PDO). Two versions of the device description are available for the EL32xx and EL33xx, representing the process image in individual and extended forms.

The difference can be seen in the revision number EL3xxx-xxxx-**XXXX**.

The EL32xx or EL33xx transmit the following process data:

- **Underrange:** Measurement is below range
- **Overrange:** Range of measurement exceeded ("Cable break" together with "Error")
- **Limit 1:** Limit value monitoring 0: ok, 1: Limit value overshoot, 2: Limit value undershot
- **Limit 2:** Limit value monitoring 0: ok, 1: Limit value overshoot, 2: Limit value undershot
- **Error:** The error bit is set if the process data is invalid (cable break, overrange, underrange)
- **TxPDO State:** Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).
- **TxPDO Toggle:** The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated. This allows the currently required conversion time to be derived.

The limit evaluation is set in the "8000" objects in the CoE directory.

Differences in the versions of the EL32xx and EL33xx series

The revision differences are illustrated below, using the EL32xx series as an example. The principles of the description apply equally to the EL33xx series.

Revision -0016 (EL32xx-xxxx-0016)

These terminal revisions have the **single process image**, see “EL32xx-0000-0016 process image in the TwinCAT 2.11 representation”.

Each item of status information is transmitted as a single, linkable process data.

Term 2 (EL3202-0000-0016)	Name	Type	Size
RTD Inputs Channel 1	Underrange	BOOL	0.1
Underrange	Overrange	BOOL	0.1
Overrange	Limit 1	BIT2	0.2
Limit 1	Limit 2	BIT2	0.2
Limit 2	Error	BOOL	0.1
Error	TxPDO State	BOOL	0.1
TxPDO State	TxPDO Toggle	BOOL	0.1
TxPDO Toggle	Value	INT	2.0
Value	Underrange	BOOL	0.1
RTD Inputs Channel 2	Overrange	BOOL	0.1
WcState	Limit 1	BIT2	0.2
WcState	Limit 2	BIT2	0.2
InfoData	Error	BOOL	0.1
State	TxPDO State	BOOL	0.1
AdsAddr	TxPDO Toggle	BOOL	0.1
	Value	INT	2.0
	WcState	BOOL	0.1
	State	UINT	2.0
	AdsAddr	AMSADD...	8.0

Fig. 170: EL32xx-0000-0016 process image in the TwinCAT 2.11 representation

Revisions -0017 (EL32xx-xxxx-0017) and higher

These terminal revisions also have the **summarized process image**, see “EL32xx-0000-0017 process image in the TwinCAT 2.11 representation”.

The individual units of information are assembled here in the usual Beckhoff representation as a 16-bit status word, and can be linked into the controller in this way.

Status word

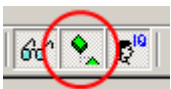
Bit	SW.15	SW.14	SW.13	SW.12	SW.11	SW.10	SW.9	SW.8	SW.7	SW.6	SW.5	SW.4	SW.3	SW.2	SW.1	SW.0
Name	Tx-PDO Toggle	Tx-PDO State	-	-	-	-	-	-	-	Error	Limit 2		Limit 1		Over-range	Under-range

In addition to this, the consolidated "status" can be folded out through the "+" symbol, and the items of process data linked individually.

Name	Type	Size
§↑ Status	Status_4...	2.0
◆↑ Value	INT	2.0
§↑ Status	Status_4...	2.0
◆↑ Value	INT	2.0
◆↑ WcState	BOOL	0.1
◆↑ State	UINT	2.0
§↑ AdsAddr	AMSADD...	8.0

Fig. 171: EL32xx-0000-0017 process image in the TwinCAT 2.11 representation

The individual items of information can also be displayed in the overview window (A) on the right. By clicking on the button



in the menu bar the information can also be displayed there.

Term 3 (EL3202-0000-0017)		Name	Type	Size
RTD Inputs Channel 1	Status	Status	Status_4096	2.0
	Underrange	Underrange	BOOL	0.1
	Overrange	Overrange	BOOL	0.1
	Limit 1	Limit 1	BIT2	0.2
	Limit 2	Limit 2	BIT2	0.2
	Error	Error	BOOL	0.1
	TxPDO State	TxPDO State	BOOL	0.1
	TxPDO Toggle	TxPDO To...	BOOL	0.1
	Value	Value	INT	2.0
RTD Inputs Channel 2	Status	Status	Status_4096	2.0
	Underrange	Underrange	BOOL	0.1
	Overrange	Overrange	BOOL	0.1
	Limit 1	Limit 1	BIT2	0.2
	Limit 2	Limit 2	BIT2	0.2
	Error	Error	BOOL	0.1
	TxPDO State	TxPDO State	BOOL	0.1
	TxPDO To...	TxPDO To...	BOOL	0.1
	Value	Value	INT	2.0
	WcState	WcState	BOOL	0.1
	State	State	UINT	2.0
	AdsAddr	AdsAddr	AMSADDRESS	8.0
	netId	netId	ARRAY [0..5] OF USINT	6.0
	netId[0]	netId[0]	USINT	1.0
	netId[1]	netId[1]	USINT	1.0
	netId[2]	netId[2]	USINT	1.0
	netId[3]	netId[3]	USINT	1.0
	netId[4]	netId[4]	USINT	1.0
	netId[5]	netId[5]	USINT	1.0
	port	port	UINT	2.0

Fig. 172: Consolidated process image in the extended representation under TwinCAT 2.11

Notes

- The **consolidated representation** is only visible from TwinCAT 2.11 and above. For reasons of compatibility, if a EL32xx-xxxx-0017 (or later) is operated in earlier TwinCAT configurations, the individual process image is displayed, prepended with the identifier "Status__".

Name	Type	Size
♦ Status__Underrange	BOOL	0.1
♦ Status__Overrange	BOOL	0.1
♦ Status__Limit 1	BIT2	0.2
♦ Status__Limit 2	BIT2	0.2
♦ Status__Error	BOOL	0.1
♦ Status__TxPDO State	BOOL	0.1
♦ Status__TxPDO Toggle	BOOL	0.1
♦ Value	INT	2.0
♦ Status__Underrange	BOOL	0.1
♦ Status__Overrange	BOOL	0.1
♦ Status__Limit 1	BIT2	0.2
♦ Status__Limit 2	BIT2	0.2
♦ Status__Error	BOOL	0.1
♦ Status__TxPDO State	BOOL	0.1
♦ Status__TxPDO Toggle	BOOL	0.1
♦ Value	INT	2.0
♦ WcState	BOOL	0.1
♦ State	UINT	2.0
♦ AdsAddr	AMSADD...	8.0

Fig. 173: Consolidated process image represented under TwinCAT 2.10

- Revisions -0016 and -0017 do not depend on the revision of the firmware installed in the terminal. This means that terminals that were supplied as EL32xx-xxxx-0016 can also be operated with a "newer" -0017 configuration, and therefore can be addressed using the consolidated process image. This case of "upwards compatibility" is permitted for the EL32xx-xxxx-0016 and -0017.
- The easiest way to determine the revision that is installed in the terminal is through a scan of the EtherCAT system. The comparison report shows the differences.

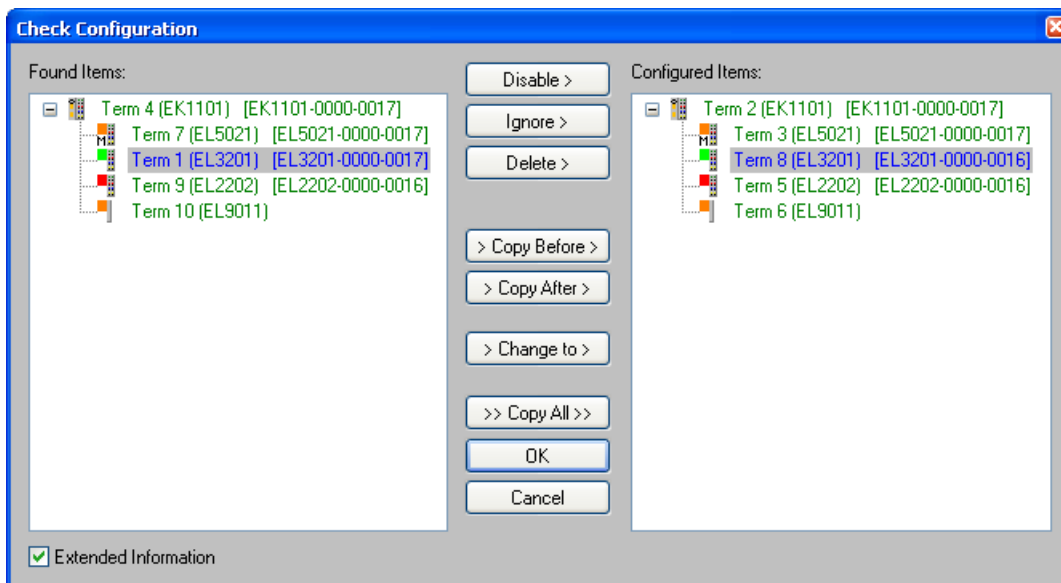


Fig. 174: Typical result after scanning an EtherCAT system

Explanation about fig. "Typical result after scanning an EtherCAT system": According to the overview on the right, an EL3201-0000-0016 was found in the configuration (*.tsm file), whereas the overview on the left shows revision -0017. The general downward compatibility of the EL terminals ensures that this kind of application is possible.

5.12 Notices on analog specifications

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

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

5.12.1 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.

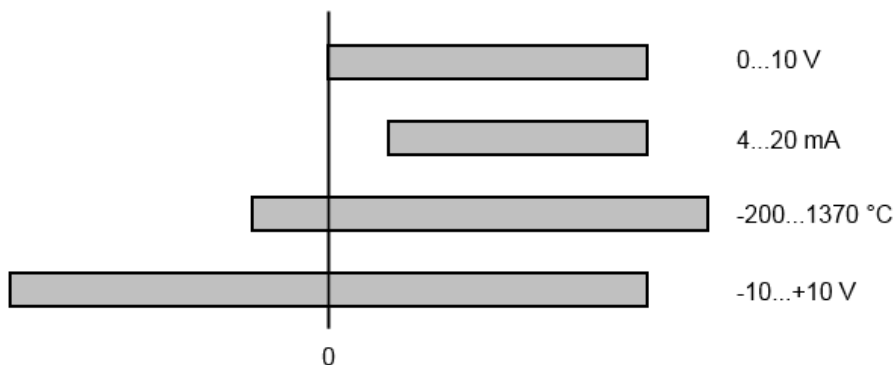


Fig. 175: Full scale value, measuring span

For the above **examples** this means:

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

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

5.12.2 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.

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

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

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

The maximum deviation can also be specified directly.

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

● 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.

5.12.3 Temperature coefficient tK [ppm/K]

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

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

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

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

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

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

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

5.12.4 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.

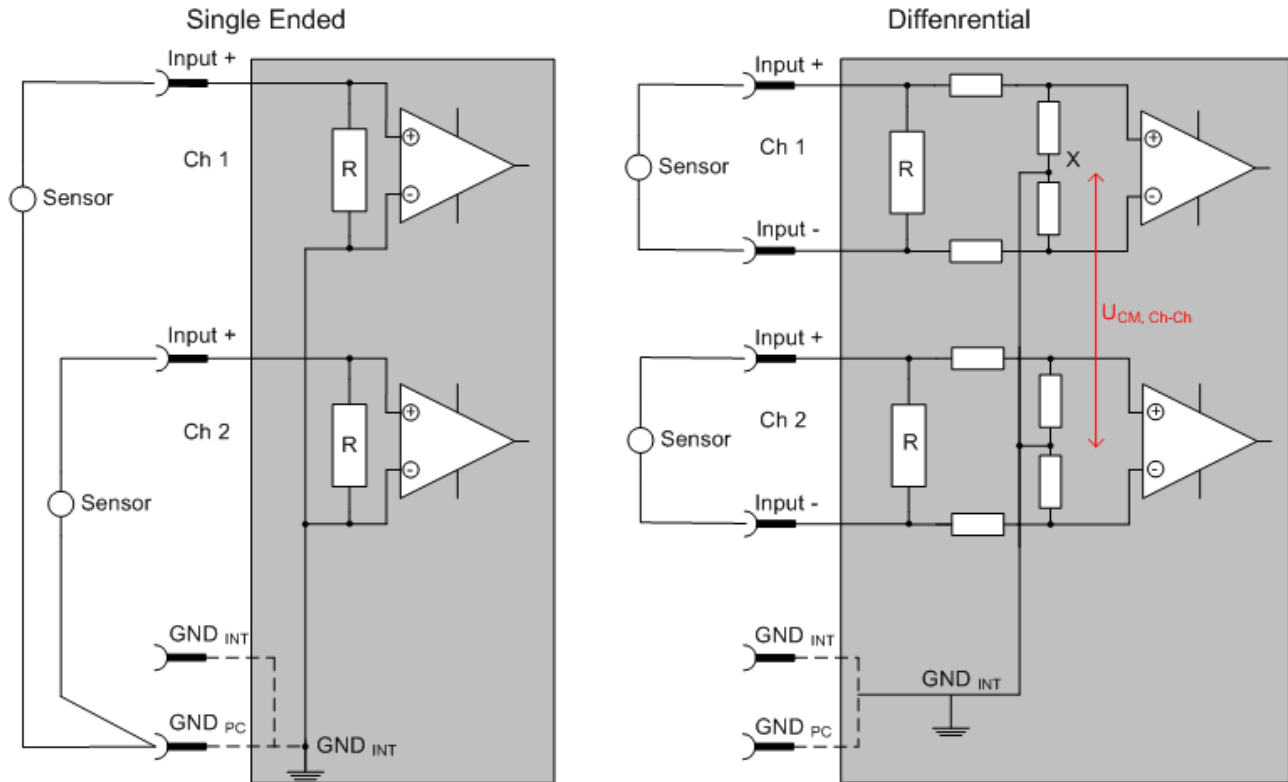


Fig. 176: SE and DIFF module as 2-channel version

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.

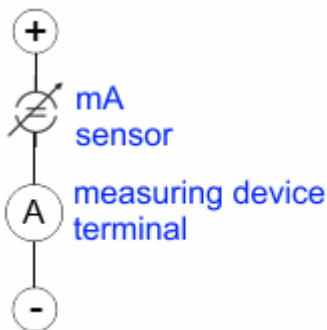


Fig. 177: 2-wire connection

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.

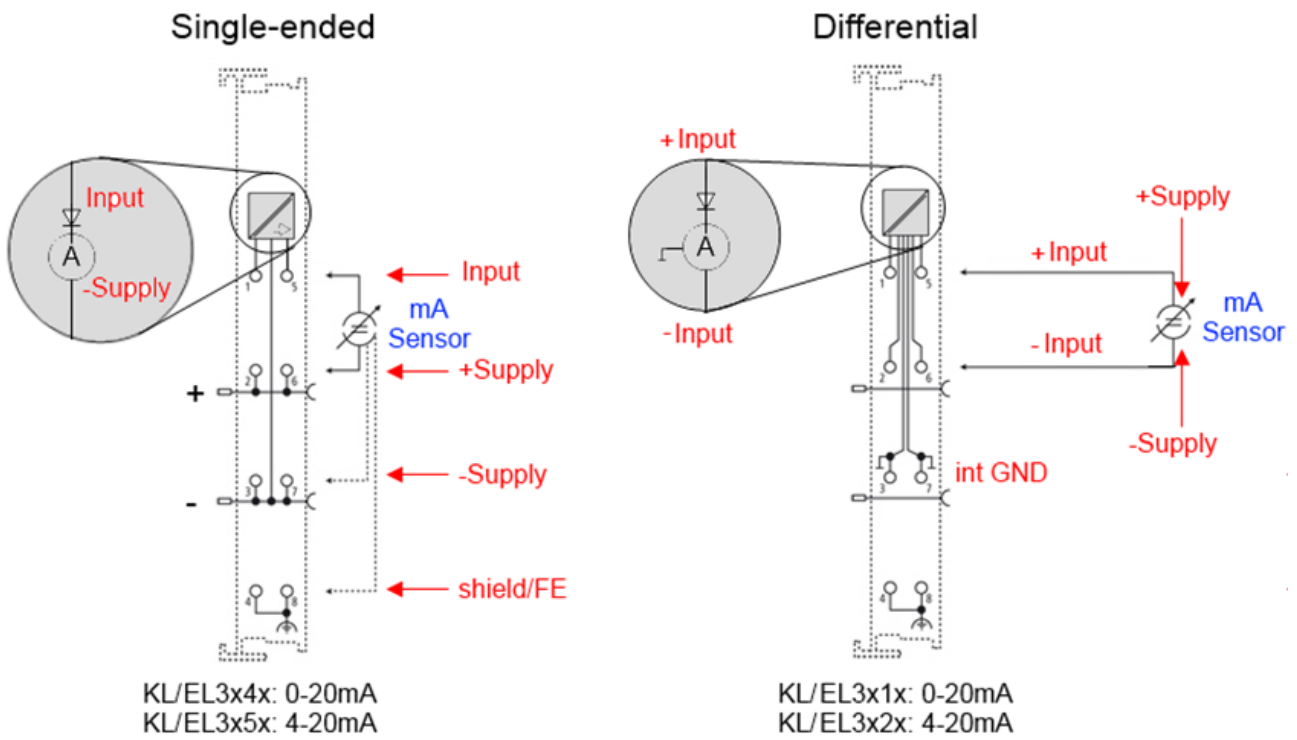


Fig. 178: Connection of externally supplied sensors

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.

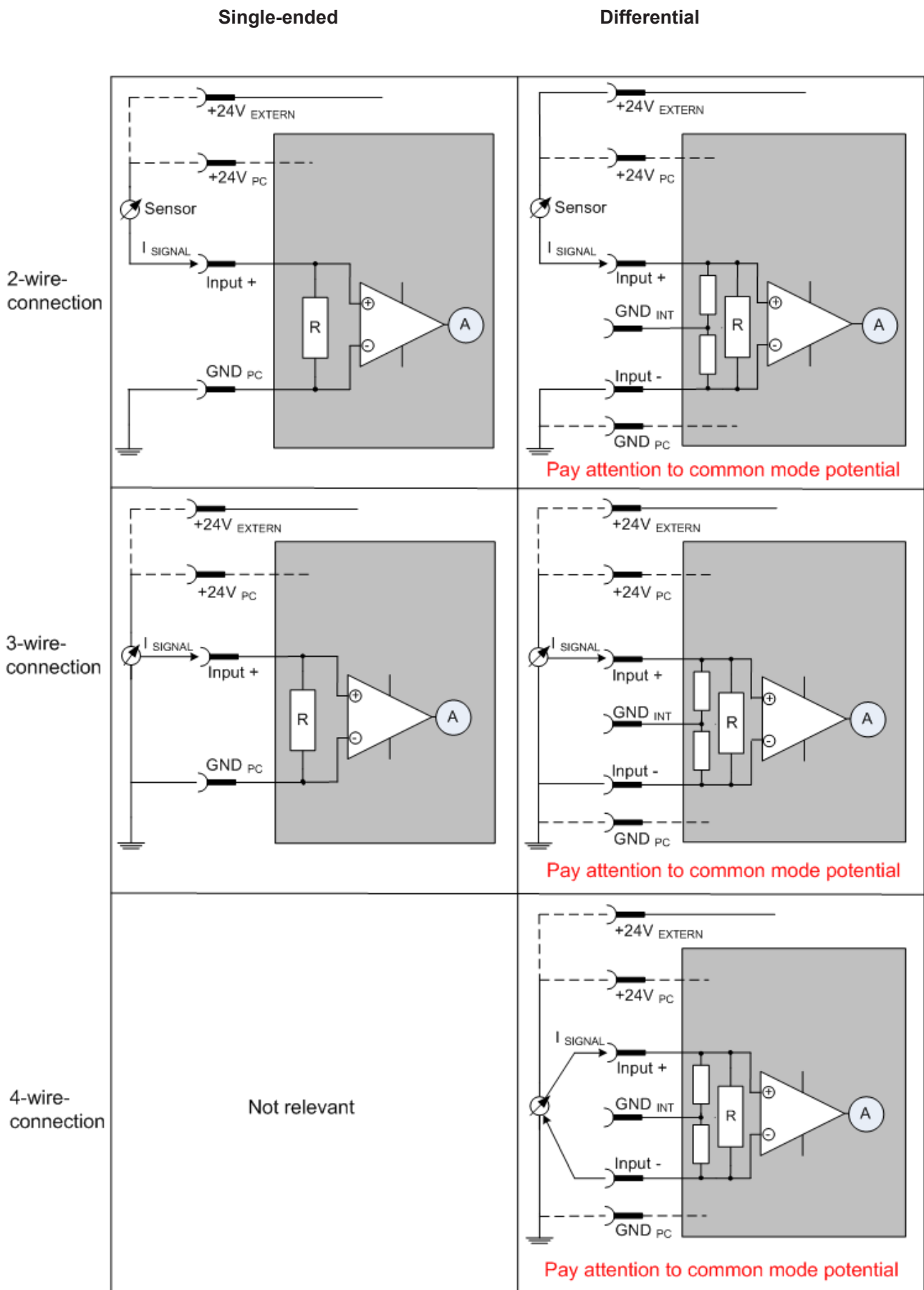


Fig. 179: 2-, 3- and 4-wire connection at single-ended and differential inputs

5.12.5 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.

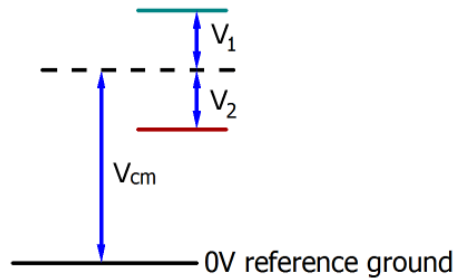


Fig. 180: Common-mode voltage (V_{cm})

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

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

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

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

Reference ground samples for Beckhoff IO devices:

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

5.12.6 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

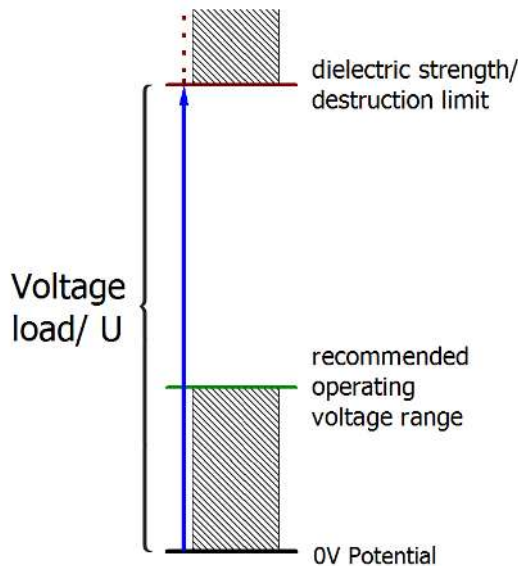


Fig. 181: Recommended operating voltage range

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

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

5.12.7 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.)

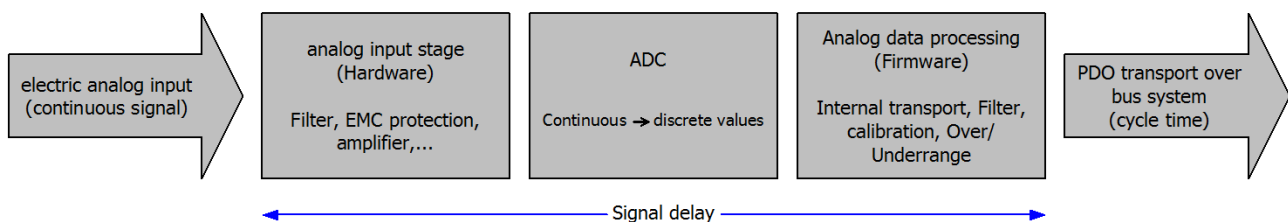


Fig. 182: Signal processing analog input

Two aspects are crucial from a user perspective:

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

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

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

1. Minimum conversion time [ms, µs]

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

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

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

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

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

2. Typical signal delay

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

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

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

A typical characterization in the device documentation may be:

2.1 Signal delay (step response)

Keywords: Settling time

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

The 90 % limit is used as detection threshold.

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

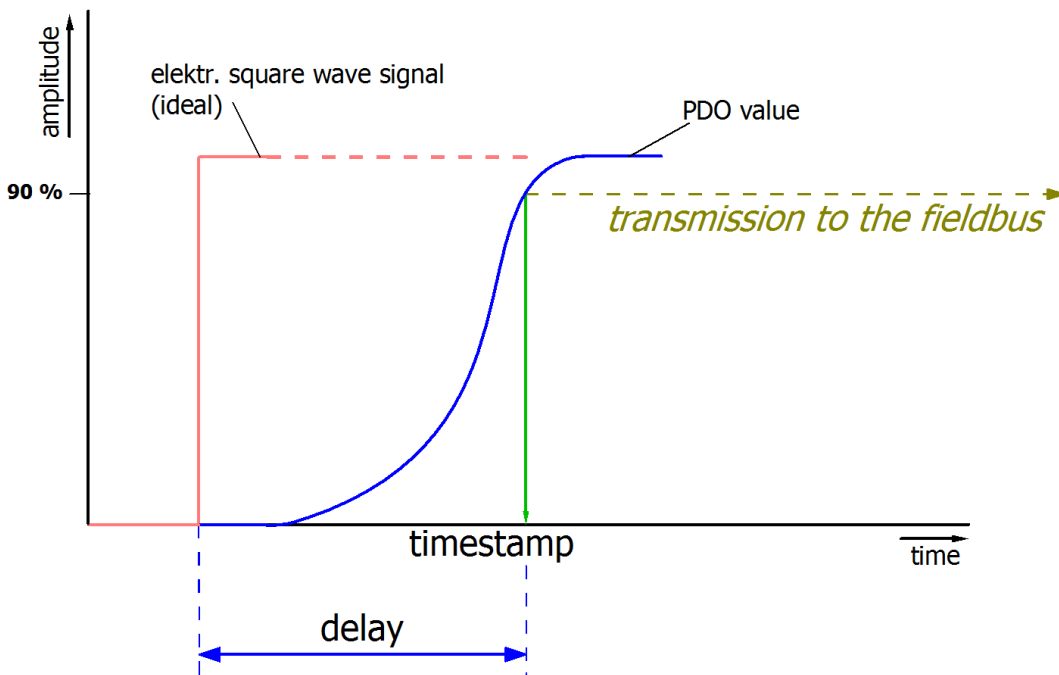


Fig. 183: Diagram signal delay (step response)

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.

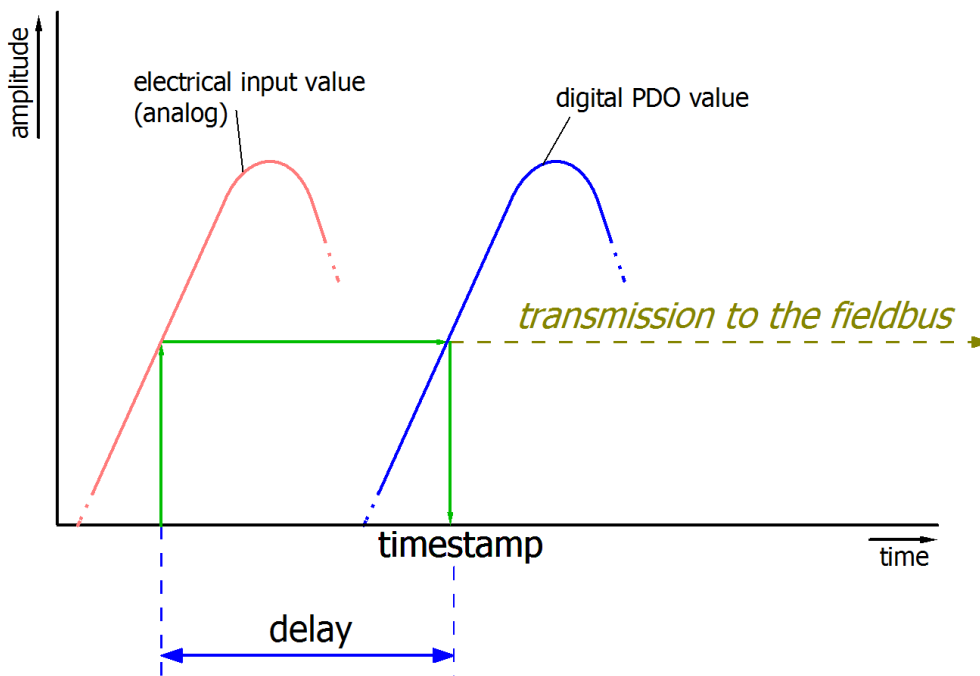


Fig. 184: Diagram signal delay (linear)

3. Additional Information

May be provided in the specification, e.g.

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

6 Appendix

6.1 EtherCAT AL Status Codes

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

6.2 Calibration certificate

For the EL320x-0020 high-precision terminals an individual calibration certificate is issued that you can download from the download area on the Beckhoff website.

The certificate can be identified on the basis of an ID at <http://www.beckhoff.com/certificates>. This ID is laser-engraved onto the terminal (see fig. "Laser-engraved ID")



Fig. 185: Laser-engraved ID

6.3 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. If a device is placed in BOOT-STRAP 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!

EL3201			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 02	03	EL3201-0000-0016	2008/03
	04		2009/04
	05		2009/05
02	06	EL3201-0000-0017	2010/04
03 - 10*	07*		2010/06
		EL3201-0000-0018	2012/06
		EL3201-0000-0019	2012/07
		EL3201-0000-0020	2014/07
		EL3201-0000-0021	2015/01
		EL3201-0000-0022	2016/01

EL3201-0010			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 02	03	EL3201-0010-0016	2008/11
	04		2009/04
	05		2009/05
02	06	EL3201-0010-0017	2010/04
03 - 09*	07*		2010/06
		EL3201-0010-0018	2012/06
		EL3201-0010-0019	2012/07
		EL3201-0010-0020	2014/07
		EL3201-0010-0021	2015/01
		EL3201-0010-0022	2016/02

EL3201-0020			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 02	03	EL3201-0020-0016	2008/11
	04		2009/04
	05		2009/05
02	06	EL3201-0020-0017	2010/04
03 - 10*	07*		2010/06
		EL3201-0020-0018	2012/06
		EL3201-0020-0019	2012/07
		EL3201-0020-0020	2014/07
		EL3201-0020-0021	2015/01
		EL3201-0020-0022	2016/02

EL3202			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 02	03	EL3202-0000-0016	2008/03
	04		2009/04
	05		2009/05
03 - 11*	06	EL3202-0000-0017	2010/04
	07*	EL3202-0000-0018	2012/06
		EL3202-0000-0019	2012/07
		EL3202-0000-0020	2014/07
		EL3202-0000-0021	2015/01
		EL3202-0000-0022	2016/01

EL3202-0010				
Hardware (HW)	Firmware (FW)	Revision No.	Date of release	
00 - 01	05	EL3202-0010-0016	2009/07	
	06	EL3202-0010-0017	2010/04	
01 - 08*	07		2010/06	
	08		2010/10	
	09		2011/07	
	10*			2011/09
		EL3202-0010-0018	2012/06	
		EL3202-0010-0019	2012/07	
		EL3202-0010-0020	2014/07	
		EL3202-0010-0021	2015/01	
EL3202-0010-0022	2016/01			

EL3202-0020			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
01 - 08*	09	EL3202-0020-0017	2011/02
	10*		2011/09
		EL3202-0020-0018	2012/06
		EL3202-0020-0019	2012/07
		EL3202-0020-0020	2014/07
		EL3202-0020-0021	2015/01
		EL3202-0020-0022	2016/01

EL3204			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 02	03	EL3204-0000-0016	2008/03
	04		2009/04
	05		2009/05
03 - 11*	06	EL3204-0000-0017	2010/04
	07*		2011/06
		EL3204-0000-0018	2012/07
		EL3204-0000-0019	2014/07
		EL3204-0000-0020	2015/01
		EL3204-0000-0021	2016/01

EL3204-0200			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 02*	01	EL3204-0200-0016	2013/08
	02*	EL3204-0200-0017	2014/04
		EL3204-0200-0018	2016/02

EL3208			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 03*	01*	EL3208-0000-0017	2014/04
		EL3208-0000-0018	2016/02

EL3214			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 01*	01*	EL3214-0000-0016	2015/01
		EL3214-0000-0017	2017/06

EL3214-0090			
Hardware (HW)	Firmware (FW)	Revision No.	Date of release
00 - 01*	01	EL3214-0090-0016	2017/02
	02*		2018/09

*) 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.

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

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

Storage locations

An EtherCAT slave stores operating data in up to 3 locations:

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

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

The TwinCAT System Manager offers mechanisms for programming all 3 parts with new data, if the slave is set up for this purpose. Generally the slave does not check whether the new data are suitable, i.e. it may no longer be able to operate if the data are unsuitable.

Simplified update by bundle firmware

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

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

Following the update, its success should be verified

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

NOTE

Risk of damage to the device!

Note the following when downloading new device files

- Firmware downloads to an EtherCAT device must not be interrupted
- Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.
- The power supply must adequately dimensioned. The signal level must meet the specification.

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

6.4.1 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:

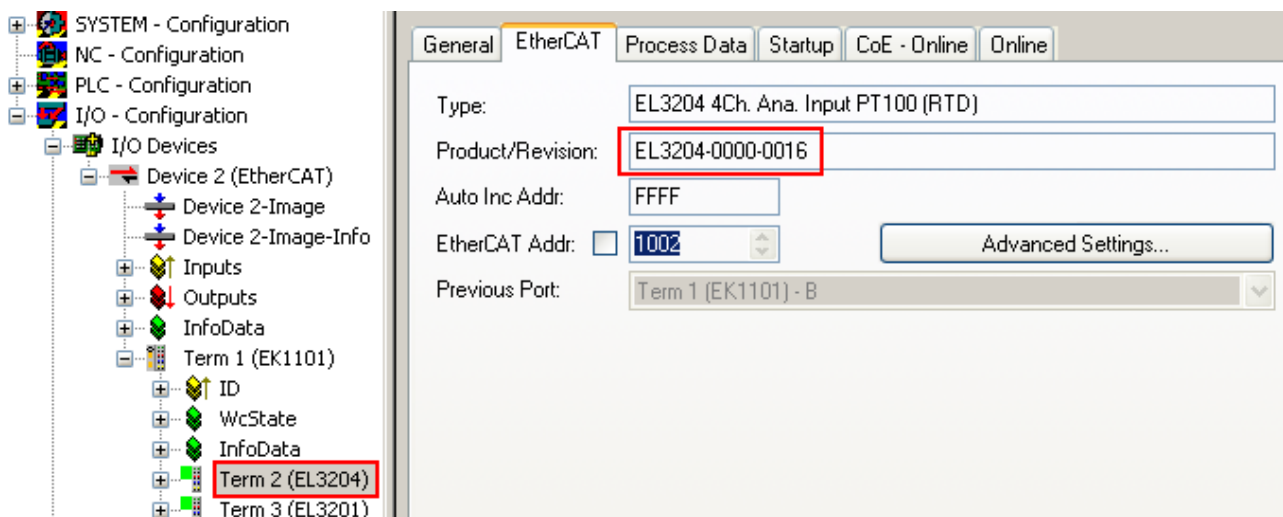


Fig. 186: Device identifier consisting of name EL3204-0000 and revision -0016

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:

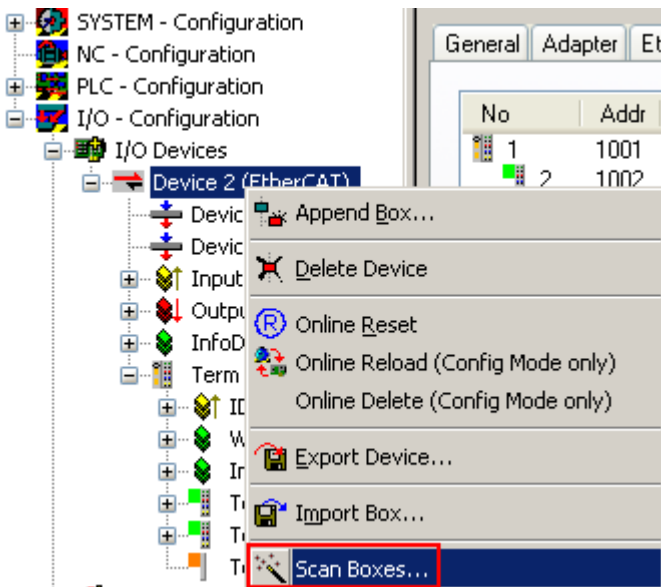


Fig. 187: Scan the subordinate field by right-clicking on the EtherCAT device

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



Fig. 188: Configuration is identical

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

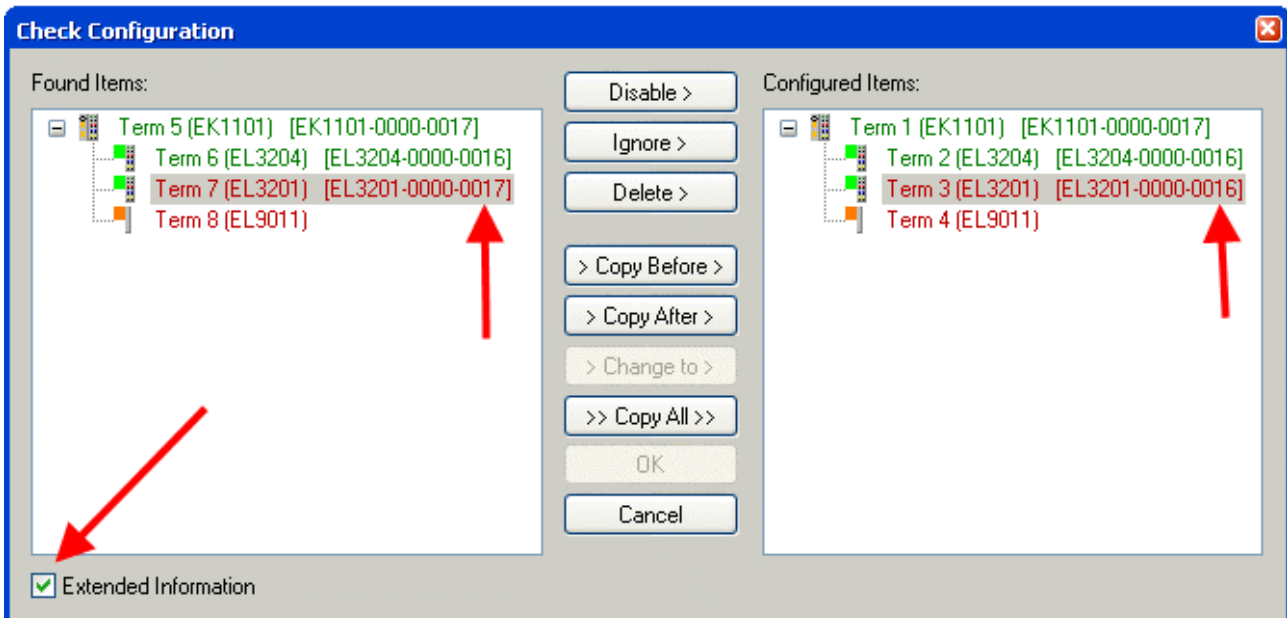


Fig. 189: Change dialog

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*

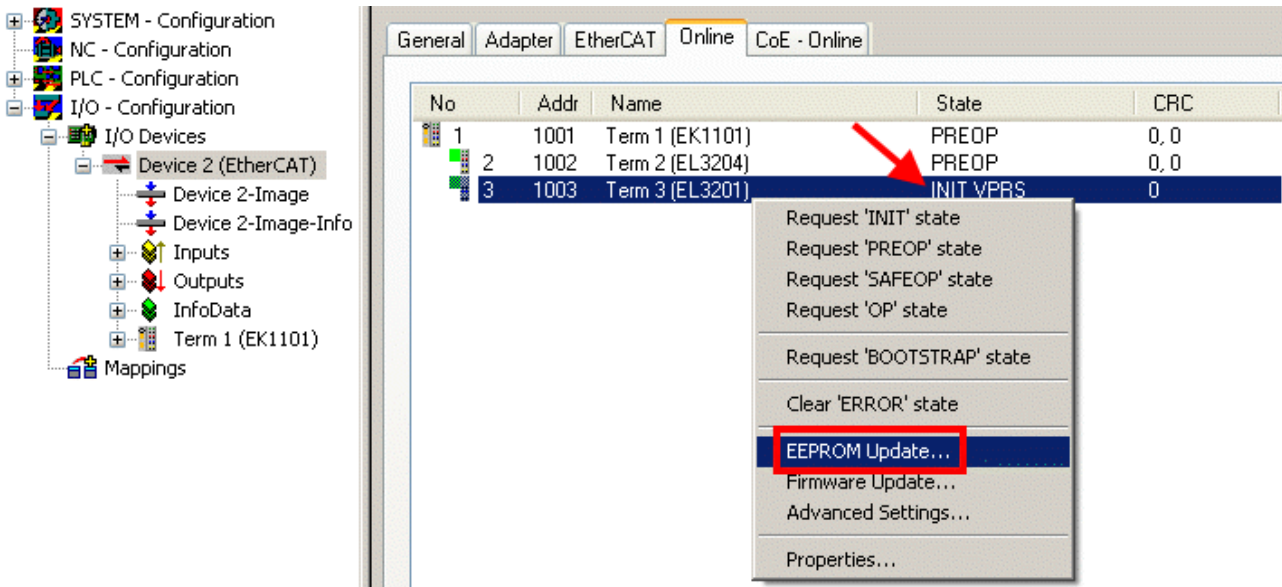


Fig. 190: EEPROM Update

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.

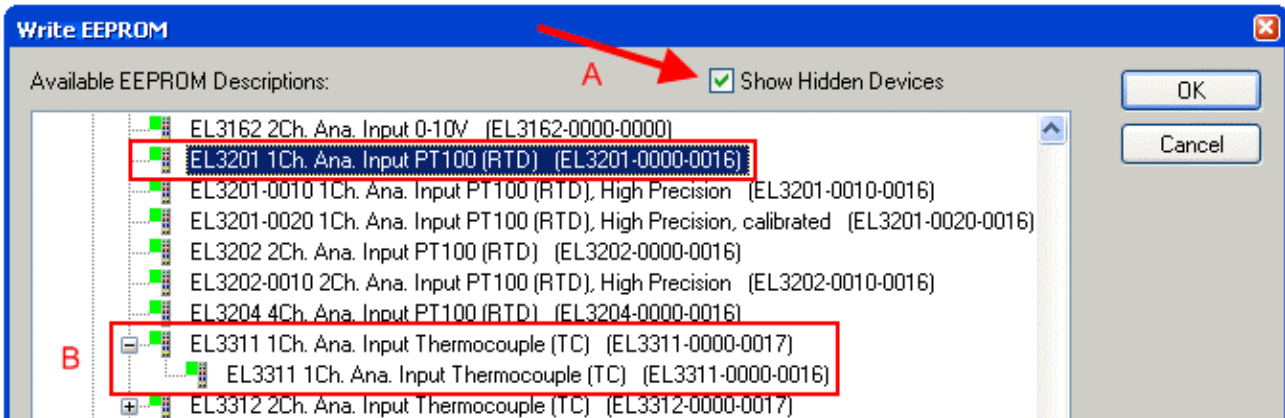


Fig. 191: Selecting the new ESI

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.

6.4.2 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 for 'Device 2 (EtherCAT)', with 'Term 2 (EL3204)' selected. The main window displays the 'CoE - Online' directory for this device. A table lists 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 <

The 'Advanced Settings' dialog is open, showing the 'Online' radio button selected and the 'All Objects' button highlighted. The 'Online Data' button is also visible in the main window.

Fig. 192: Display of EL3204 firmware version

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

6.4.3 Updating controller firmware *.efw

i **CoE directory**

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

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

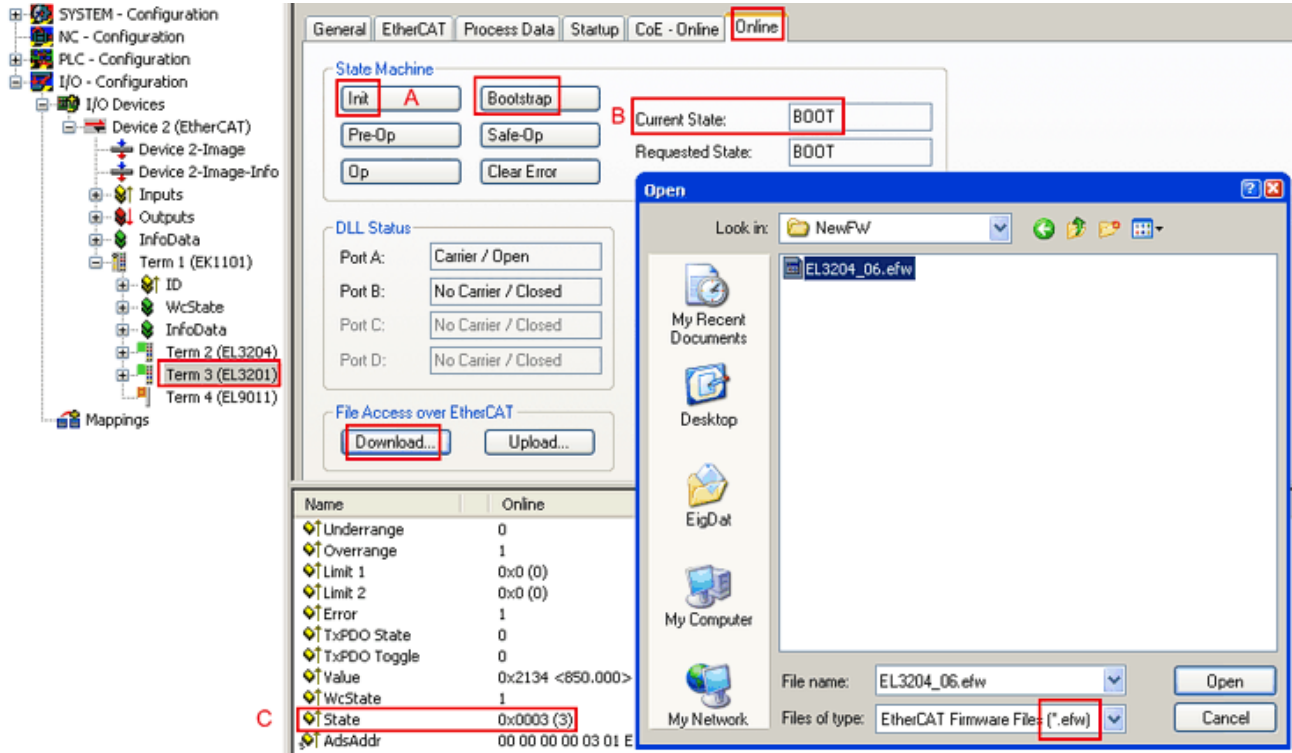
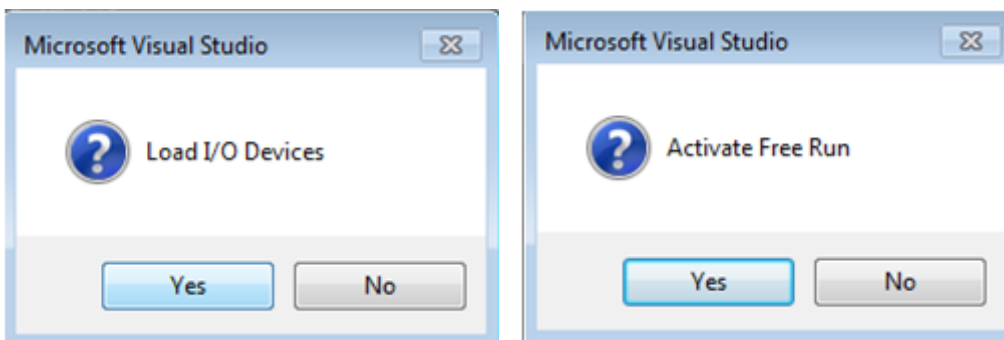


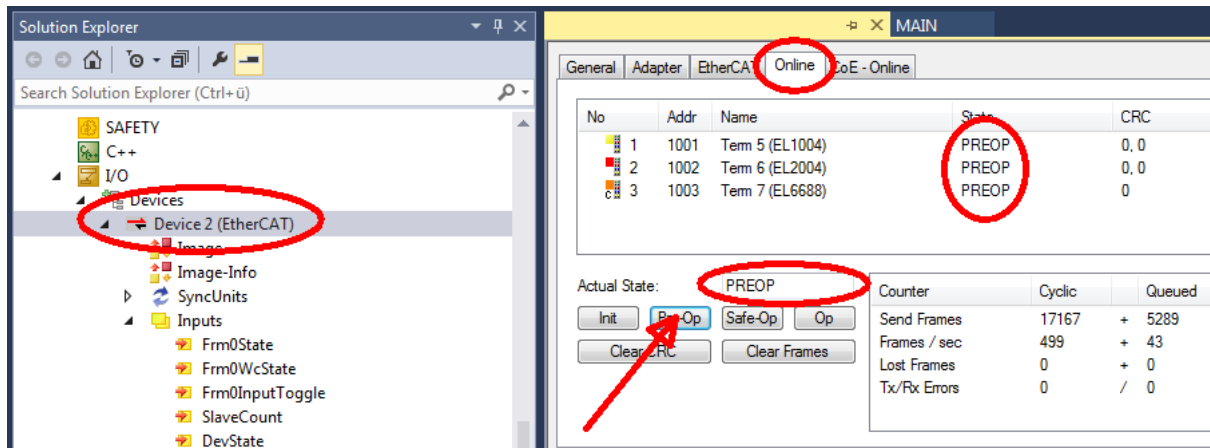
Fig. 193: Firmware Update

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

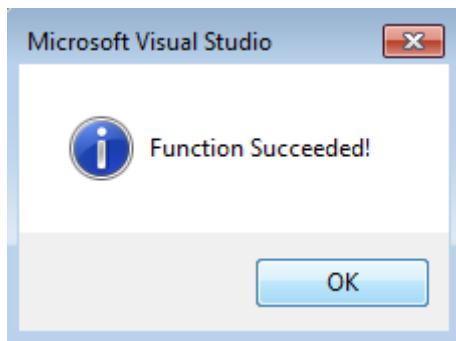
- Switch TwinCAT system to ConfigMode/FreeRun with cycle time ≥ 1 ms (default in ConfigMode is 4 ms). A FW-Update during real time operation is not recommended.



- 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.

6.4.4 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.

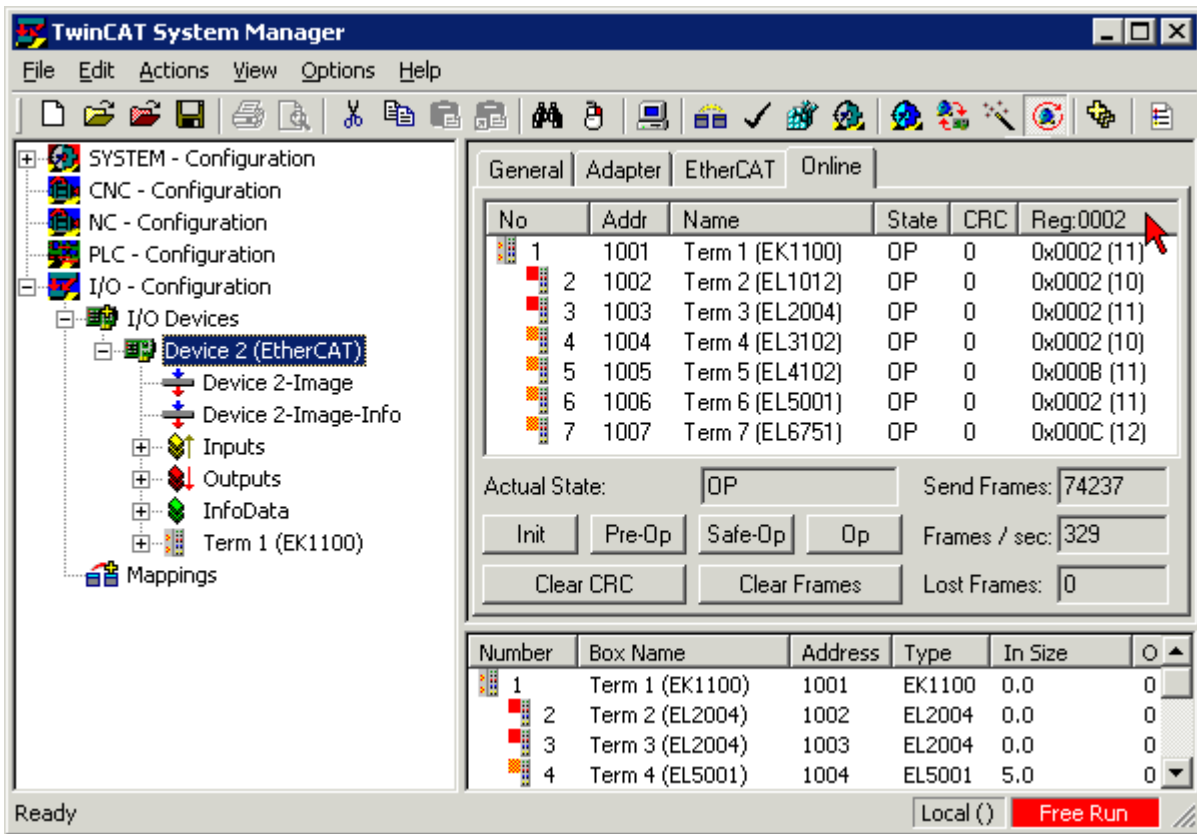


Fig. 194: FPGA firmware version definition

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

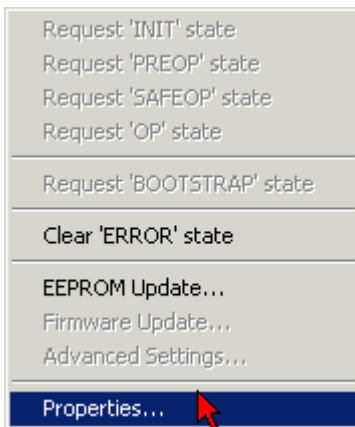


Fig. 195: Context menu *Properties*

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.

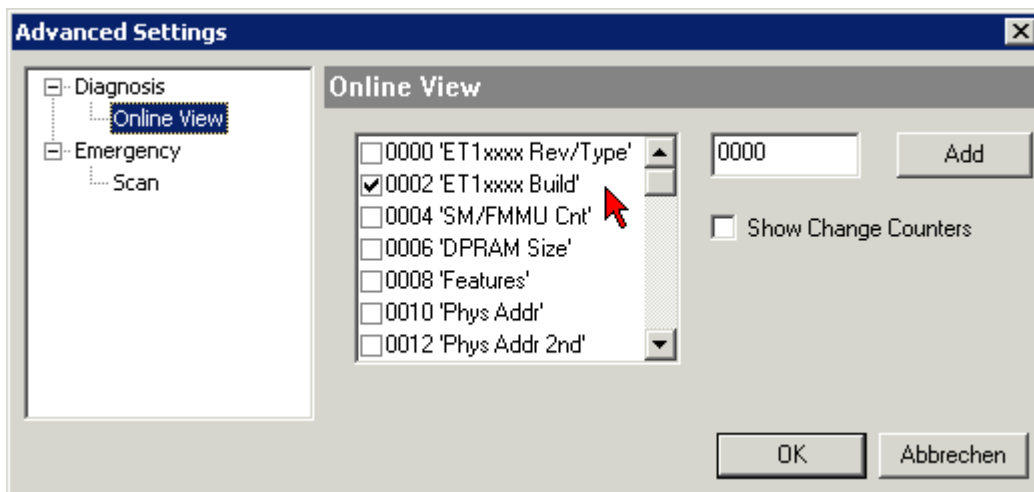


Fig. 196: Dialog *Advanced Settings*

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.

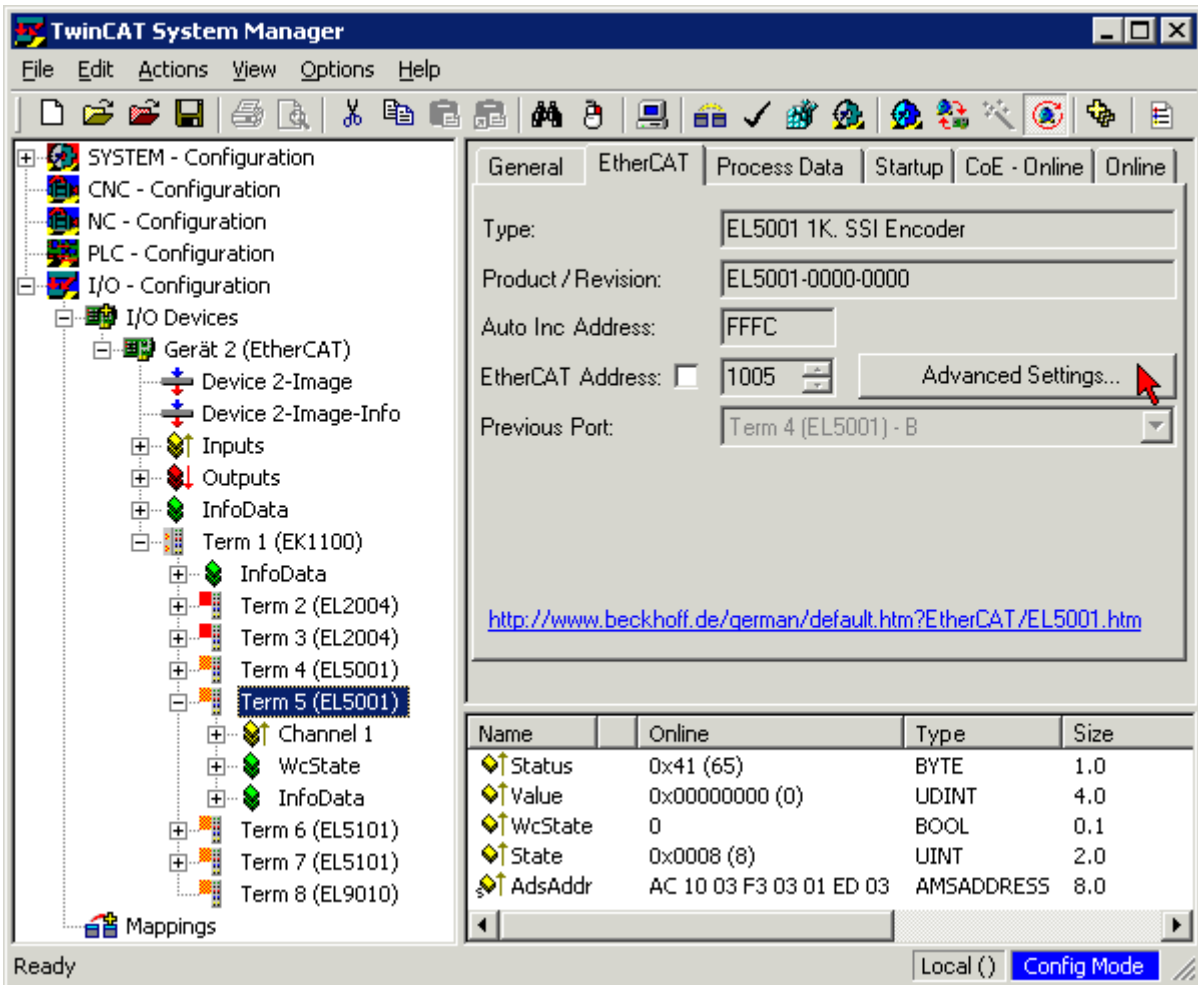
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

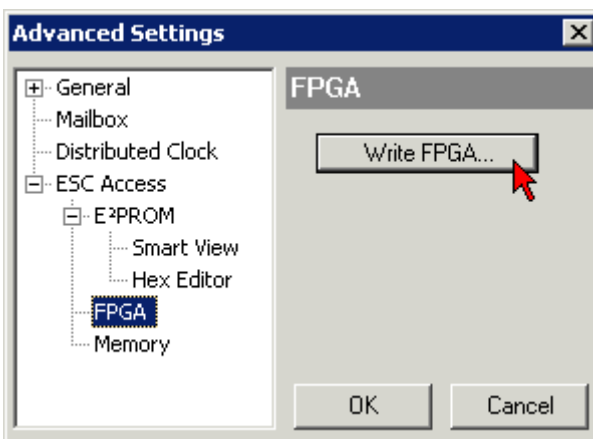
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

- 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.

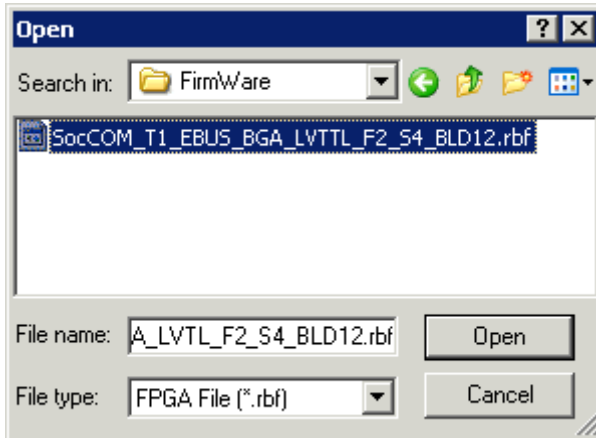
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and click the *Advanced Settings* button in the *EtherCAT* tab:



- 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!

6.4.5 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.

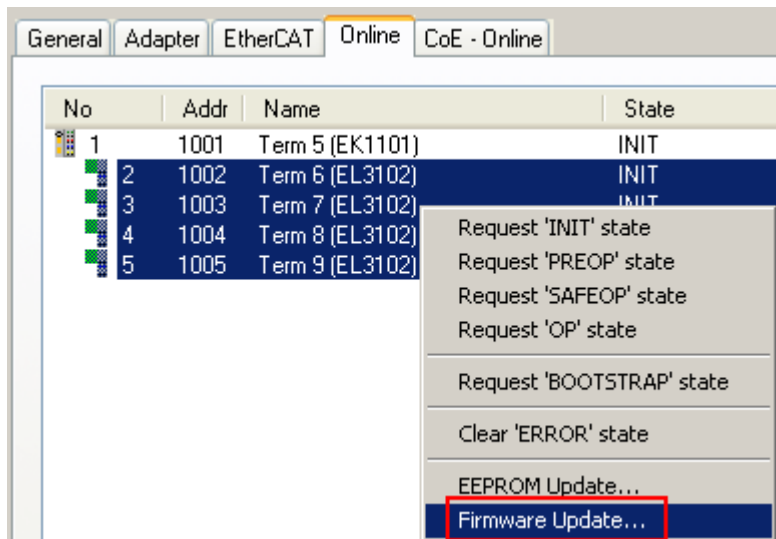


Fig. 197: Multiple selection and firmware update

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

6.5 Restoring the delivery state

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

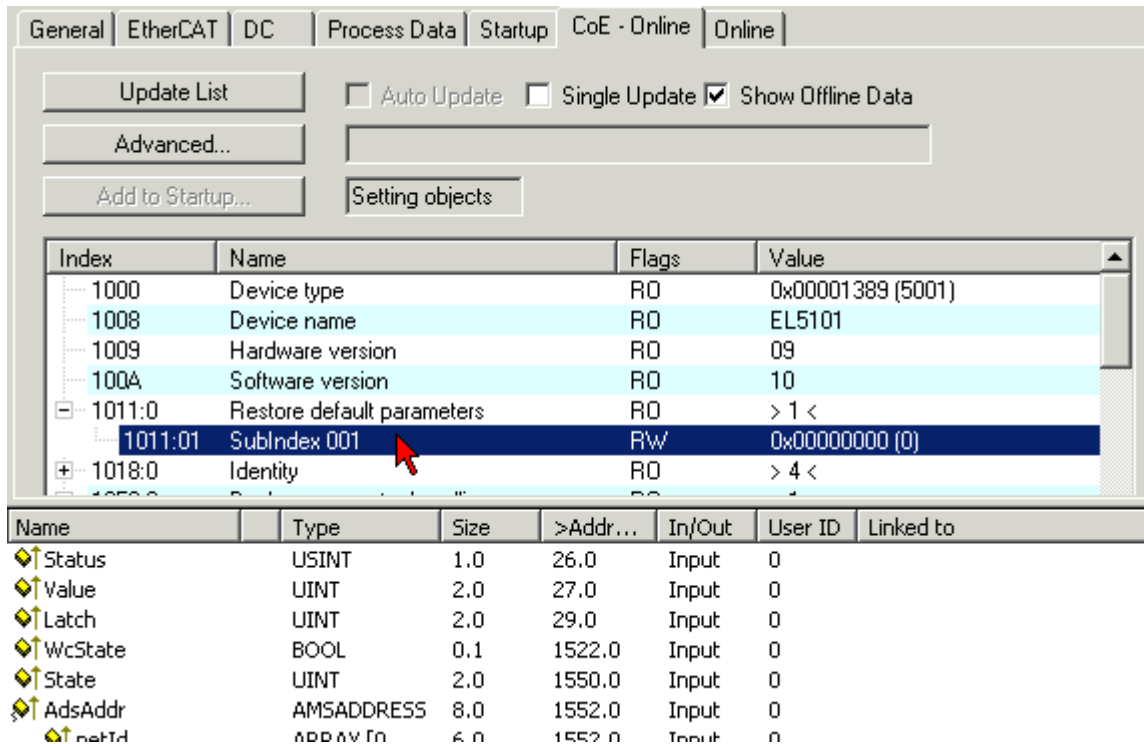


Fig. 198: Selecting the "Restore default parameters" PDO

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

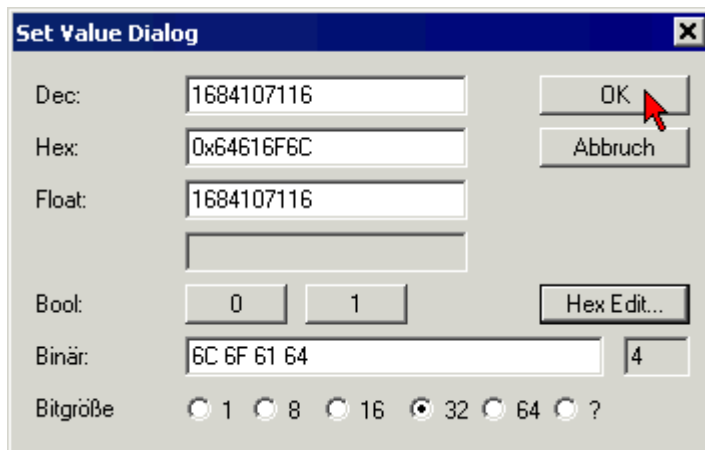


Fig. 199: Entering a restore value in the Set Value dialog

Alternative restore value

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

6.6 Support and Service

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

Beckhoff's branch offices and representatives

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

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

<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 5246 963 0
Fax: +49 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 5246 963 157
Fax: +49 5246 963 9157
e-mail: support@beckhoff.com

Beckhoff Service

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

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

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

Table of figures

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.....	12
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.....	13
Fig. 6	EP1908-0002 IP67 EtherCAT Safety Box with batch number/ date code 071201FF and unique serial number 00346070	13
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	EL3201	14
Fig. 10	EL3202	15
Fig. 11	EL3204	15
Fig. 12	EL3214	16
Fig. 13	EL3208	17
Fig. 14	EL3214-0090	18
Fig. 15	Display of the measurement and calculation of the resistance/temperature	20
Fig. 16	4-wire and 3-wire connection technology	21
Fig. 17	Examples for temperature-dependent resistance values	24
Fig. 18	System manager current calculation	34
Fig. 19	EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog	35
Fig. 20	States of the EtherCAT State Machine.....	37
Fig. 21	"CoE Online " tab	39
Fig. 22	Startup list in the TwinCAT System Manager	40
Fig. 23	Offline list.....	41
Fig. 24	Online list	41
Fig. 25	Spring contacts of the Beckhoff I/O components.....	45
Fig. 26	Attaching on mounting rail	46
Fig. 27	Disassembling of terminal.....	47
Fig. 28	Power contact on left side.....	48
Fig. 29	Correct positioning.....	50
Fig. 30	Incorrect positioning.....	50
Fig. 31	Recommended distances for standard installation position	51
Fig. 32	Other installation positions	52
Fig. 33	Standard wiring.....	53
Fig. 34	Pluggable wiring	53
Fig. 35	High Density Terminals.....	54
Fig. 36	Connecting a cable on a terminal point	55
Fig. 37	EL3201 pin assignment	60
Fig. 38	EL3202 pin assignment	60
Fig. 39	EL3201-00x0 pin assignment	62
Fig. 40	EL3202-00x0 pin assignment	62
Fig. 41	EL3204-0x00 pin assignment	64

Fig. 42	EL3214 pin assignment	65
Fig. 43	EL3214-0090 pin assignment	65
Fig. 44	EL3208 pin assignment	67
Fig. 45	Example of EL3202-0000 shield connection with shield contact, in the case of potential interference sources inside the control cabinet	70
Fig. 46	Example of EL3202-0000 shield connection with shield contact, in the case of potential interference sources inside and outside the control cabinet	71
Fig. 47	Relationship between user side (commissioning) and installation	73
Fig. 48	Control configuration with Embedded PC, input (EL1004) and output (EL2008)	74
Fig. 49	Initial TwinCAT 2 user interface	75
Fig. 50	Selection of the target system	76
Fig. 51	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	76
Fig. 52	Select "Scan Devices..."	77
Fig. 53	Automatic detection of I/O devices: selection the devices to be integrated	77
Fig. 54	Mapping of the configuration in the TwinCAT 2 System Manager	78
Fig. 55	Reading of individual terminals connected to a device	78
Fig. 56	TwinCAT PLC Control after startup	79
Fig. 57	Sample program with variables after a compile process (without variable integration)	80
Fig. 58	Appending the TwinCAT PLC Control project	80
Fig. 59	PLC project integrated in the PLC configuration of the System Manager	81
Fig. 60	Creating the links between PLC variables and process objects	81
Fig. 61	Selecting PDO of type BOOL	82
Fig. 62	Selecting several PDOs simultaneously: activate "Continuous" and "All types"	82
Fig. 63	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample	83
Fig. 64	Choose target system (remote)	84
Fig. 65	PLC Control logged in, ready for program startup	85
Fig. 66	Initial TwinCAT 3 user interface	86
Fig. 67	Create new TwinCAT project	86
Fig. 68	New TwinCAT3 project in the project folder explorer	87
Fig. 69	Selection dialog: Choose the target system	87
Fig. 70	Specify the PLC for access by the TwinCAT System Manager: selection of the target system ..	88
Fig. 71	Select "Scan"	88
Fig. 72	Automatic detection of I/O devices: selection the devices to be integrated	89
Fig. 73	Mapping of the configuration in VS shell of the TwinCAT3 environment	89
Fig. 74	Reading of individual terminals connected to a device	90
Fig. 75	Adding the programming environment in "PLC"	91
Fig. 76	Specifying the name and directory for the PLC programming environment	91
Fig. 77	Initial "Main" program of the standard PLC project	92
Fig. 78	Sample program with variables after a compile process (without variable integration)	93
Fig. 79	Start program compilation	93
Fig. 80	Creating the links between PLC variables and process objects	94
Fig. 81	Selecting PDO of type BOOL	94
Fig. 82	Selecting several PDOs simultaneously: activate "Continuous" and "All types"	95
Fig. 83	Application of a "Goto Link" variable, using "MAIN.bEL1004_Ch4" as a sample	95
Fig. 84	TwinCAT development environment (VS shell): logged-in, after program startup	96
Fig. 85	System Manager "Options" (TwinCAT 2)	98

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

Fig. 130	Dialog “Change to Compatible Type...” (left: TwinCAT 2; right: TwinCAT 3).....	120
Fig. 131	TwinCAT 2 Dialog Change to Alternative Type	120
Fig. 132	Branch element as terminal EL3751.....	121
Fig. 133	“General” tab.....	121
Fig. 134	„EtherCAT“ tab.....	122
Fig. 135	“Process Data” tab.....	123
Fig. 136	Configuring the process data.....	124
Fig. 137	„Startup“ tab.....	125
Fig. 138	“CoE – Online” tab	126
Fig. 139	Dialog “Advanced settings”	127
Fig. 140	„Online“ tab	127
Fig. 141	"DC" tab (Distributed Clocks).....	128
Fig. 142	Selection of the diagnostic information of an EtherCAT Slave	130
Fig. 143	Basic EtherCAT Slave Diagnosis in the PLC.....	131
Fig. 144	EL3102, CoE directory	133
Fig. 145	Example of commissioning aid for a EL3204	134
Fig. 146	Default behaviour of the System Manager	135
Fig. 147	Default target state in the Slave	135
Fig. 148	PLC function blocks	136
Fig. 149	Illegally exceeding the E-Bus current	137
Fig. 150	Warning message for exceeding E-Bus current	137
Fig. 151	Accessing the CoE parameters in TwinCAT System Manager , EL3201	140
Fig. 152	“not connected” setting	141
Fig. 153	Importing the XML file	143
Fig. 154	Table for calculating the B-parameter	144
Fig. 155	Fig. 2: Table for calculating the Steinhart-Hart parameters	145
Fig. 156	Comparison of implementation via B-parameters and Steinhart-Hart parameters	147
Fig. 157	Comparison of Pt500 sensor implementation	149
Fig. 158	Data diagram	150
Fig. 159	Typical frequency response of a notch filter set to 50 Hz.....	152
Fig. 160	Set Value Dialog	152
Fig. 161	Adding the TwinSAFE SC process data under the component, e.g. EL5021-0090	155
Fig. 162	TwinSAFE SC component process data, example EL5021-0090	155
Fig. 163	Adding a TwinSAFE SC connection	155
Fig. 164	Creating a link to TwinSAFE SC terminal	156
Fig. 165	Selecting a free CRC	156
Fig. 166	Selecting the process data size and the process data	157
Fig. 167	Selection of the process data	157
Fig. 168	CoE objects 0x8010:01 and 0x8010:02.....	157
Fig. 169	Entering the safety address and the CRC	158
Fig. 170	EL32xx-0000-0016 process image in the TwinCAT 2.11 representation	177
Fig. 171	EL32xx-0000-0017 process image in the TwinCAT 2.11 representation	178
Fig. 172	Consolidated process image in the extended representation under TwinCAT 2.11.....	179
Fig. 173	Consolidated process image represented under TwinCAT 2.10	180
Fig. 174	Typical result after scanning an EtherCAT system.....	180
Fig. 175	Full scale value, measuring span	181

Fig. 176 SE and DIFF module as 2-channel version	183
Fig. 177 2-wire connection.....	185
Fig. 178 Connection of externally supplied sensors	186
Fig. 179 2-, 3- and 4-wire connection at single-ended and differential inputs	187
Fig. 180 Common-mode voltage (V _{cm}).....	188
Fig. 181 Recommended operating voltage range.....	189
Fig. 182 Signal processing analog input.....	189
Fig. 183 Diagram signal delay (step response)	191
Fig. 184 Diagram signal delay (linear)	191
Fig. 185 Laser-engraved ID	193
Fig. 186 Device identifier consisting of name EL3204-0000 and revision -0016	197
Fig. 187 Scan the subordinate field by right-clicking on the EtherCAT device	198
Fig. 188 Configuration is identical	198
Fig. 189 Change dialog	198
Fig. 190 EEPROM Update	199
Fig. 191 Selecting the new ESI.....	199
Fig. 192 Display of EL3204 firmware version	200
Fig. 193 Firmware Update	201
Fig. 194 FPGA firmware version definition	203
Fig. 195 Context menu Properties	203
Fig. 196 Dialog Advanced Settings	204
Fig. 197 Multiple selection and firmware update	206
Fig. 198 Selecting the "Restore default parameters" PDO	207
Fig. 199 Entering a restore value in the Set Value dialog.....	207