BECKHOFF New Automation Technology

Documentation | EN

EL3692

2 Channel Resistance Measurement Terminal, high-precision

2022-01-06 | Version: 3.1

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
It is the duty of the technical personnel to use the documentation published at the respective time of each installation and commissioning.

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

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

Copyright

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

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

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

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

Warning of damage to property or environment

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

Information on handling the product

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

1.3 Documentation issue status

Version	Comment
3.1	- Update chapter "Technical data" - Update structure
3.0	- Update chapter "Technical data" - Update chapter "Quick start" - Update structure
2.9	- Addenda EL3692-0020 and EL3692-0030 - Update chapter "Technical data" - Update revision status - Update structure
2.8	- Update chapter „Specific data“ - Update revision status - Update structure
2.7	- Update chapter "Introduction“ - Update chapter "Technical data" - Update chapter "Technology" - Update chapter „Specific data“ - Update chapter "CoE" - Update structure
2.6	- Update chapter "CoE” - Update structure
2.5	- Update chapter "CoE” - Update structure
2.4	- Update chapter "Specific data" - Update revision status - Update structure
2.3	- Update chapter "Example program" - Update structure
2.2	- Update of Technical data - Addenda chapter "Instructions for ESD protection" - Update chapter "Notices on Analog specification"
2.1	- Update chapter "Notes on the documentation" - Update of Technical data - Addenda chapter "Installation instructions for enhanced mechanical load capacity" - Update chapter "TwinCAT 2.1x" -> "TwinCAT Development Environment" and "TwinCAT Quick Start"
2.0	- Migration - Update structure
1.4	- Addenda chapter "Notices on analog specifications" - Update chapter "Operating behavior, diagnostics" - Update chapter "Technical data" - Update revision status - Update structure
1.3	- Update chapter "Technical data"
1.2	- Addenda AutoRange operation mode
1.1	- Addenda
1.0	- Addenda - First publication
0.1	- Preliminary documentation for EL3692

1.4 Version identification of EtherCAT devices

1.4.1 General notes on marking

Designation

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

- family key
- type
- version
- revision

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

Notes

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

1.4.2 Version identification of EL terminals

The serial number/ data code for Beckhoff IO devices is usually the 8-digit number printed on the device or on a sticker. The serial number indicates the configuration in delivery state and therefore refers to a whole production batch, without distinguishing the individual modules of a batch.

Structure of the serial number: **KK YY FF HH**	Example with serial number 12 06 3A 02:
KK - week of production (CW, calendar week)	12 - production week 12
YY - year of production	06 - production year 2006
FF - firmware version	3A - firmware version 3A
HH - hardware version	02 - hardware version 02

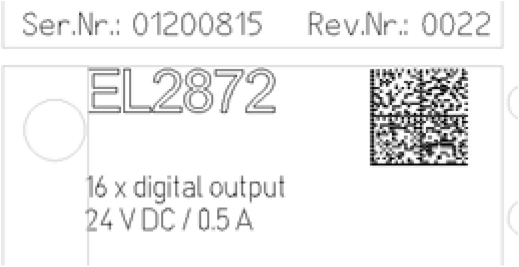


Fig. 1: EL2872 with revision 0022 and serial number 01200815

1.4.3 Beckhoff Identification Code (BIC)

The Beckhoff Identification Code (BIC) is increasingly being applied to Beckhoff products to uniquely identify the product. The BIC is represented as a Data Matrix Code (DMC, code scheme ECC200), the content is based on the ANSI standard MH10.8.2-2016.

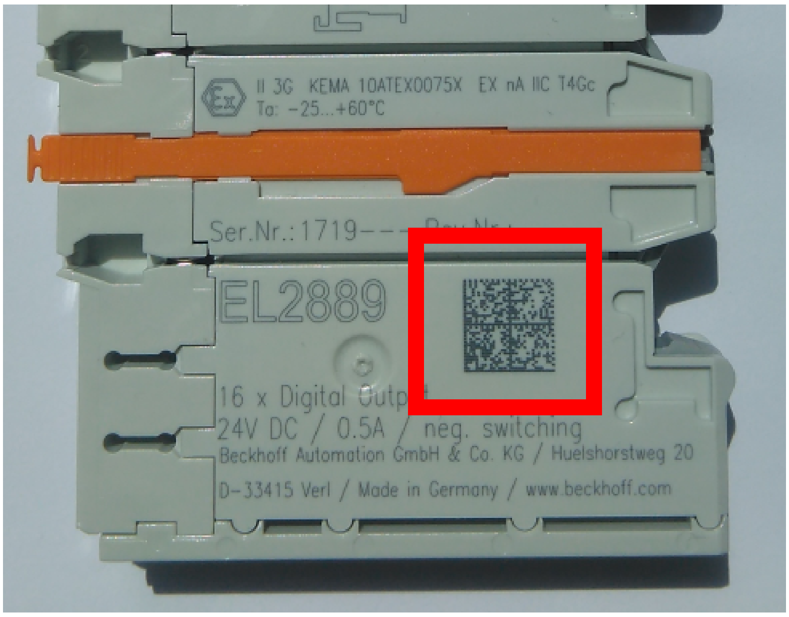

II 3G KEMA 10ATEX0075X EX nA IIC T4Gc
Ta: -25...+60°C
Ser.Nr.:1719---
EL2889
16 x Digital Output
24V DC / 0.5A / neg. switching
Beckhoff Automation GmbH & Co. KG / Huelshorstweg 20
D-33415 Verl / Made in Germany / www.beckhoff.com

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

The BIC will be introduced step by step across all product groups.

Depending on the product, it can be found in the following places:

- on the packaging unit
- directly on the product (if space suffices)
- on the packaging unit and the product

The BIC is machine-readable and contains information that can also be used by the customer for handling and product management.

Each piece of information can be uniquely identified using the so-called data identifier (ANSI MH10.8.2-2016). The data identifier is followed by a character string. Both together have a maximum length according to the table below. If the information is shorter, spaces are added to it.

Following information is possible, positions 1 to 4 are always present, the other according to need of production:

Posi-tion	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	1P072222
2	Beckhoff Traceability Number (BTN**)**	**Unique serial number, see note below**	SBTN	12	SBTNk4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	1KEL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	Q1
5	Batch number	Optional: Year and week of production	2P	14	2P401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	51S678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	32	30PF971, 2*K183
...					

Further types of information and data identifiers are used by Beckhoff and serve internal processes.

Structure of the BIC

Example of composite information from positions 1 to 4 and with the above given example value on position 6. The data identifiers are highlighted in bold font:

1P072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

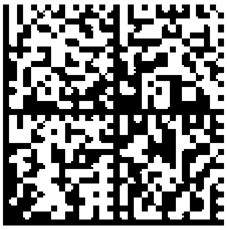


Fig. 3: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

An important component of the BIC is the Beckhoff Traceability Number (BTN, position 2). The BTN is a unique serial number consisting of eight characters that will replace all other serial number systems at Beckhoff in the long term (e.g. batch designations on IO components, previous serial number range for safety products, etc.). The BTN will also be introduced step by step, so it may happen that the BTN is not yet coded in the BIC.

NOTICE

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

1.4.4 Electronic access to the BIC (eBIC)

Electronic BIC (eBIC)

The Beckhoff Identification Code (BIC) is applied to the outside of Beckhoff products in a visible place. If possible, it should also be electronically readable.

Decisive for the electronic readout is the interface via which the product can be electronically addressed.

K-bus devices (IP20, IP67)

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

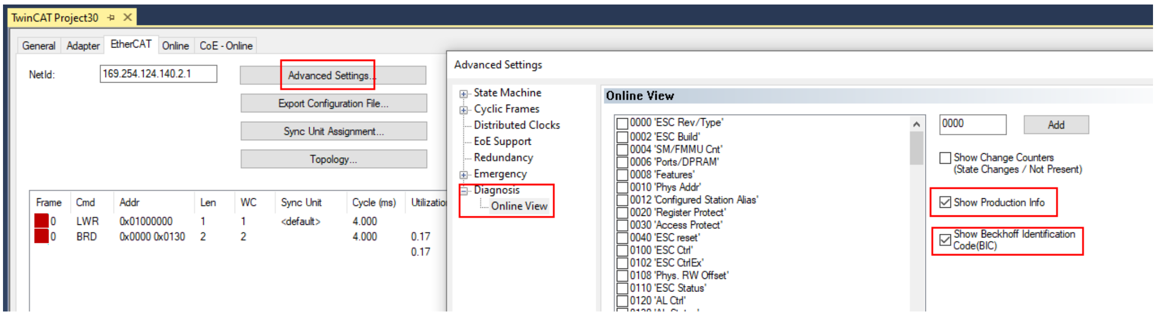
EtherCAT devices (IP20, IP67)

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

The eBIC is also stored in the ESI-EEPROM. The eBIC was introduced into the Beckhoff I/O production (terminals, box modules) from 2020; widespread implementation is expected in 2021.

The user can electronically access the eBIC (if existent) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM
    - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.
    - To do this,
      check the checkbox "Show Beckhoff Identification Code (BIC)" under
      EtherCAT → Advanced Settings → Diagnostics:



NetId: 169.254.124.140.2.1

Frame	Cmd	Addr	Len	WC	Sync Unit	Cycle (ms)	Utilizatio
0	LWR	0x01000000	1	1	<default>	4.000	
0	BRD	0x0000 0x0130	2	2		4.000	0.17
							0.17

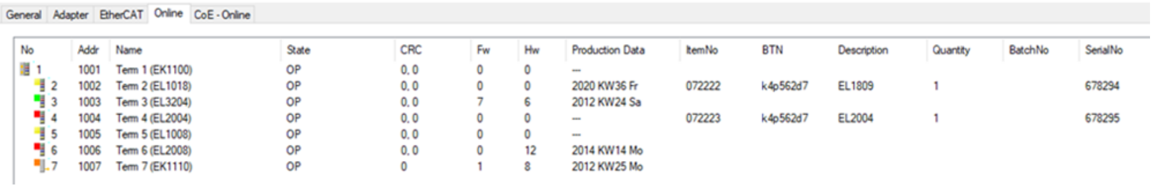
Online View

□ 0000 'ESC Rev/Type'
□ 0002 'ESC Build'
□ 0004 'SM/FMMU Cnt'
□ 0006 'Ports/DPRAM'
□ 0008 'Features'
□ 0010 'Phys Addr'
□ 0012 'Configured Station Alias'
□ 0020 'Register Protect'
□ 0030 'Access Protect'
□ 0040 'ESC reset'
□ 0100 'ESC Ctrl'
□ 0102 'ESC CtrlEx'
□ 0108 'Phys. RW Offset'
□ 0110 'ESC Status'
□ 0120 'AL Ctrl'

0000

□ Show Change Counters (State Changes / Not Present)
☑ Show Production Info
☑ Show Beckhoff Identification Code(BIC)

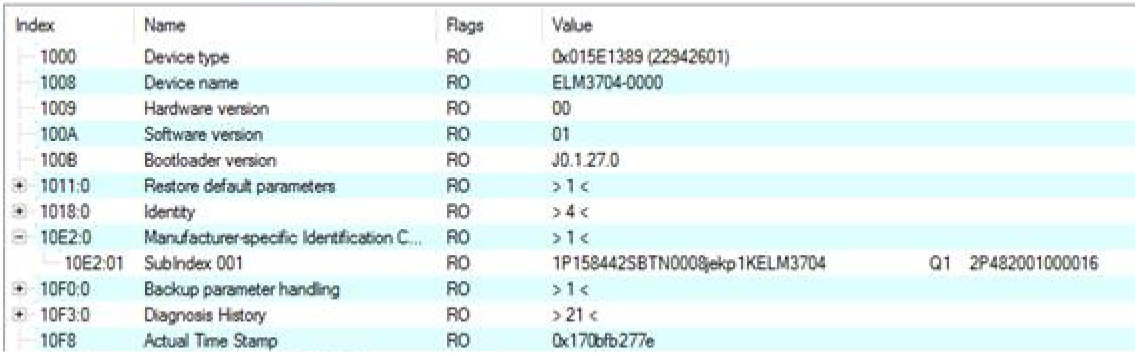
    - The BTN and its contents are then displayed:



No	Addr	Name	State	CRC	Fw	Hw	Production Data	ItemNo	BTN	Description	Quantity	BatchNo	SerialNo
1	1001	Term 1 (EK1100)	OP	0, 0	0	0	---						
2	1002	Term 2 (EL1018)	OP	0, 0	0	0	2020 KW36 Fr	072222	k4p562d7	EL1809	1		678294
3	1003	Term 3 (EL3204)	OP	0, 0	7	6	2012 KW24 Sa						
4	1004	Term 4 (EL2004)	OP	0, 0	0	0	---	072223	k4p562d7	EL2004	1		678295
5	1005	Term 5 (EL1008)	OP	0, 0	0	0	---						
6	1006	Term 6 (EL2008)	OP	0, 0	0	12	2014 KW14 Mo						
7	1007	Term 7 (EK1110)	OP	0	1	8	2012 KW25 Mo						

    - Note: as can be seen in the illustration, the production data HW version, FW version and production date, which have been programmed since 2012, can also be displayed with "Show Production Info".
    - Access from the PLC: From TwinCAT 3.1. build 4024.24 the functions *FB_EcReadBIC* and *FB_EcReadBTN* are available in the Tc2_EtherCAT Library from v3.3.19.0 for reading into the PLC..
- In the case of EtherCAT devices with CoE directory, the object 0x10E2:01 can additionally by used to display the device's own eBIC; the PLC can also simply access the information here:

  - The device must be in PREOP/SAFEOP/OP for access:



Index	Name	Flags	Value
1000	Device type	RO	0x015E1389 (22942601)
1008	Device name	RO	ELM3704-0000
1009	Hardware version	RO	00
100A	Software version	RO	01
100B	Bootloader version	RO	J0.1.27.0
+ 1011:0	Restore default parameters	RO	> 1 <
+ 1018:0	Identity	RO	> 4 <
− 10E2:0	Manufacturer-specific Identification C...	RO	> 1 <
10E2:01	SubIndex 001	RO	1P158442SBTN0008jekp1KELM3704 Q1 2P482001000016
+ 10F0:0	Backup parameter handling	RO	> 1 <
+ 10F3:0	Diagnosis History	RO	> 21 <
10F8	Actual Time Stamp	RO	0x170bfb277e

  - The object 0x10E2 will be introduced into stock products in the course of a necessary firmware revision.
  - From TwinCAT 3.1. build 4024.24 the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* are available in the Tc2_EtherCAT Library from v3.3.19.0 for reading into the PLC.
- For processing the BIC/BTN data in the PLC, the following auxiliary functions are available in *Tc2_Utilities* from TwinCAT 3.1 build 4024.24 onwards
  - F_SplitBIC: The function splits the Beckhoff Identification Code (BIC) sBICValue into its components based on known identifiers and returns the recognized partial strings in a structure ST_SplitBIC as return value.
  - BIC_TO_BTN: The function extracts the BTN from the BIC and returns it as a value.
- Note: in the case of electronic further processing, the BTN is to be handled as a string(8); the identifier "SBTN" is not part of the BTN.
- Technical background
  The new BIC information is additionally written as a category in the ESI-EEPROM during the device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored with the help of a category according to ETG.2010. ID 03 indicates to all EtherCAT masters that they must not overwrite these data in case of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see there. This results in a memory requirement of approx. 50..200 bytes in the EEPROM.
- Special cases
  - If multiple, hierarchically arranged ESCs are installed in a device, only the top-level ESC carries the eBIC Information.
  - If multiple, non-hierarchically arranged ESCs are installed in a device, all ESCs carry the eBIC Information.
  - If the device consists of several sub-devices with their own identity, but only the top-level device is accessible via EtherCAT, the eBIC of the top-level device is located in the CoE object directory 0x10E2:01 and the eBICs of the sub-devices follow in 0x10E2:nn.

PROFIBUS, PROFINET, DeviceNet devices etc.

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

2 Product description

2.1 Introduction

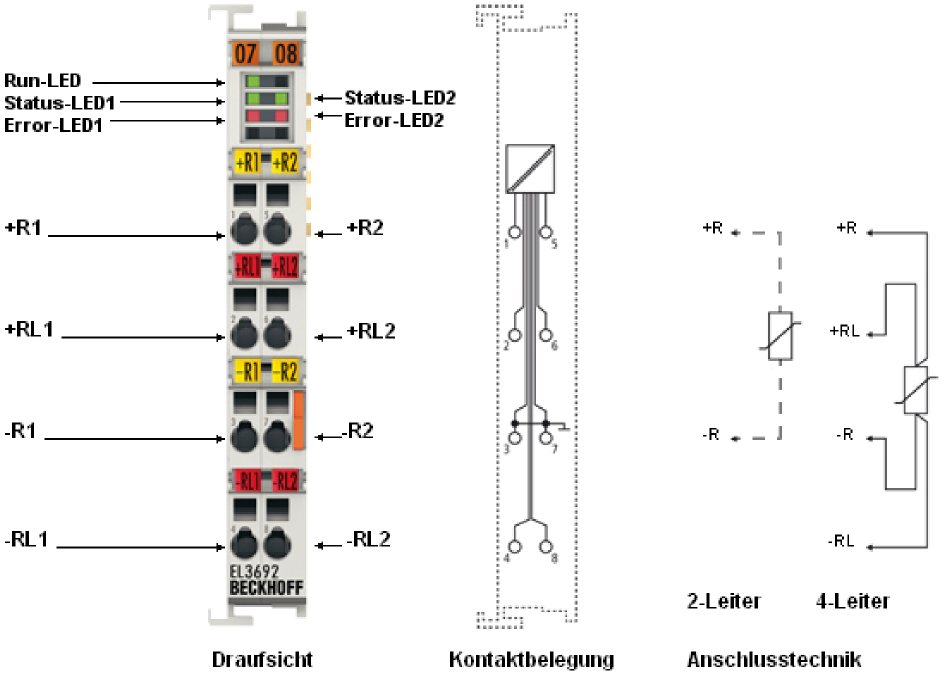


Fig. 4: EL3692

2 channel resistance measurement terminal

The EL3692 analog input terminal allows direct resistance measurement in 9 measuring ranges from 0..100 MΩ to 0..10 mΩ on 2 channels.

The circuitry of the EtherCAT Terminal enables measurement in 2- and 4-wire versions. The EL3692 offers measuring range selection, either automatic or through the controller. It is possible to determine the connected resistance with an update rate up to 100 Hz.

Due to the galvanic isolation of 1500 V between the field side and the E-bus, in single-channel mode measurements can be carried out at live points (within the permissible range). The EtherCAT Terminal indicates its state through LEDs. Malfunctions such as broken wires are indicated by error LEDs.

As further variants the EL3692-0020 with individual factory calibration certificate and the EL3692-0030 with external calibration certificate are available.

Please read the notes on the calibration certificate [▸ 20] and identification features [▸ 20] of these terminals.

Quick links

- EtherCAT basics
- Technology EL3692 [▸ 17]
- Most important CoE entries [▸ 165]
- Quick start [▸ 141]
- Process data [▸ 149]

2.2 Technology

The EL3692 resistance measuring terminal can measure resistances between milliohm (mΩ) and megohm (MΩ). 2- and 4-wire measurements are supported. The terminal is fully configurable via the Bus Coupler or the control system. Different output formats may be selected or own scaling activated.

Measuring principle of the terminal

The terminal actively applies a current to the test specimen. This current simultaneously flows through internal reference resistances. The actual resistance is determined from the proportion of the measured voltages. The terminal measures ratiometrically, i.e. fluctuations in the supply voltage and/or the measuring current have no influence on the accuracy.

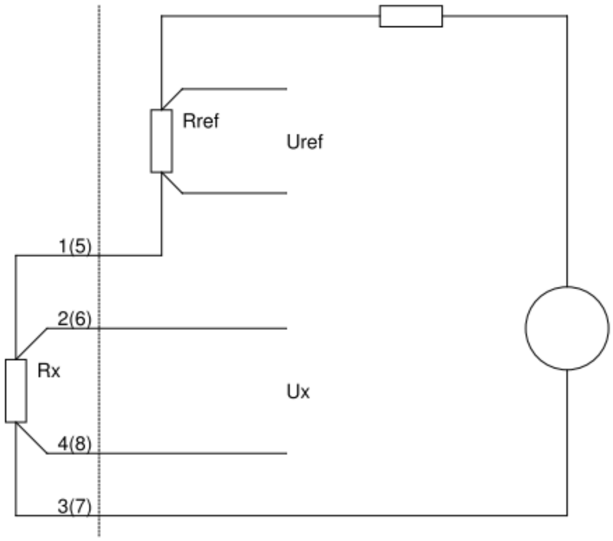


Fig. 5: Principle of operation

The figure illustrates the following relationship between the known resistance R_{ref} and the unknown resistance R_x:

$R_x / R_{ref} = U_x / U_{ref}$
$R_x = R_{ref} * U_x / U_{ref}$

The two channels of the EL3692 are measured alternately. To compensate capacitive processes the conversion time depends on the measuring range: the larger measuring range the longer the conversion. To measure a resistance at short intervals, the EL3692 should be used in single-channel mode. For measuring inductive/capacitive loads it should be noted that transition processes influence the result. A delay time can be specified in the channel settings in the CoE. The measurement and updating of the process data will only take place once this delay time (in [ms]) has elapsed (CoE 0x80n0:05 and :31 [▶ 165]). This delay time is particularly significant in Autorange mode: If it is too small, the measuring range may not change in Autorange mode because the respective measuring range limits are not reached. The automatic control in the EL3692 will try to optimize the waiting times, although manual adaptation may still be required.

The measuring frequency is determined from the filter setting for each channel. The lower the filter setting, the more measurements are carried out and averaged, and the more reliable is therefore the measurement.

The 9 measuring ranges each cover one decade. They each measure from 0 Ohm to 110% of full scale with their typical measuring current, see table. The measuring ranges of the two channels can be used independently of each other.

Although all measuring ranges theoretically measure down to 0 Ohm, the relatively large measuring error and the 2/4-wire connection technology must be taken into account with such small resistances. Apart from that, it will hardly be possible to achieve a "correct" zero measured value, because not even a connected (short) wire bridge has a resistance of 0 Ω.

In AutoRange mode, the measuring range used is determined by the terminal itself. For AutoRange operation, each measuring range can measure up to 110% above the respective measuring range end value. In this range, the measuring range is then switched to the next higher measuring range. Accordingly, below 10% of the measuring range, the changeover to the next smaller measuring range takes place. This method results in the following naming of the measuring ranges e.g. as "10..100 Ω", with 100Ω as measuring range end value and 10 Ω as lower threshold for the measuring range switchover.

In the manual mode, the measuring range is specified by the user via process data or CoE, in this case 0..110% of the respective measuring range can be measured.

The respective resistance is measured with 24 bit resolution (incl. sign). The measured value transmission takes place as 32 bit process date, in float (fixed point) or integer representation. Since no negative resistances can occur, the sign bit is used for the representation up to 110%

For more detailed information on settings and operating modes please refer to the sections on Process data [▸ 149], Operational characteristics, Diagnostics [▸ 144] and Specific data [▸ 153].

2.3 Technical data

Technical data	EL3692	EL3692-0020	EL3692-0030
Number of inputs	2		
Connection technology	2 or 4 wire		
Measuring range	9 measuring ranges with measuring range setpoint 100 mΩ ... 10 MΩ: 0..0.1 Ω, 0..1 Ω, 0..10 Ω, 0..100 Ω, 0..1 kΩ, 0..10 kΩ, 0..100 kΩ, 0..1 MΩ, 0..10 MΩ MΩ, 0..10 MΩ		
Resolution	24 bits incl. sign		
Sampling type	simultaneous		
Ground reference	single ended		
Conversion time	10 ms ..400 ms; depends on measuring range; operation mode (alternating, one channel), waiting time, filter settings		
Broken wire detection	yes		
Internal resistance	>100 MΩ		
Filter characteristics	Hardware 5 kHz, Firmware adjustable 2.5 - 100 Hz		
Measuring error	<0.5% (relative to the respective full scale value with 4-wire connection)		
Supply voltage for electronics	via the E-bus		
Current consumption via E-bus	typ. 220 mA		
Distributed Clocks	No		
Special features	automatic range selection; "auto range", pulse and continuous measurement		
Width in the process image	max. 24 bytes input, max. 4 bytes output		
Electrical isolation	1500 V (E-bus/field voltage)		
Configuration	via TwinCAT System Manager		
Weight	approx. 60 g		
permissible ambient temperature range during operation	0 °C ... + 55 °C		
permissible ambient temperature range during storage	-25 °C ... + 85 °C		
permissible relative humidity	95 %, no condensation		
Dimensions (W x H x D)	approx. 15 mm x 100 mm x 70 mm (width aligned: 12 mm)		
Mounting [▶ 38]	on 35 mm mounting rail conforms to EN 60715		
Vibration/shock resistance	conforms to EN 60068-2-6 / EN 60068-2-27 see also installation instructions for terminals with increased mechanical load capacity [▶ 41]		
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4		
Protection class	IP20		
Installation position	variable		
Calibration certificate [▶ 20]	-	yes (factory calibration certificate)	yes (DAkkS or ISO17025)
Marking / Approval*)	CE, EAC, UKCA cULus [▶ 37], ATEX [▶ 35]		CE, EAC, UKCA

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

Ex marking

Standard	Marking
ATEX	II 3 G Ex nA IIC T4 Gc

2.4 Note on Beckhoff calibration certificates

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. This documentation as a calibration certificate is only provided for devices that are expressly delivered with a certificate.

The calibration certificate (or German: “Kalibrierschein”) entitles the residual error after compensation/ adjustment to the used standard (reference device). The calibration certificate (as a PDF document) is to be assigned to the device via a unique number. It is therefore not a statement about a device class such as e.g. an approval, but always only applies to a single, named device. It is available for download.

The calibration certificate documents the measurement accuracy at the time the certificate was issued and contains, among other things, information on the ambient conditions and the reference instrument used. It does not contain statement about the behavior or the change of the measuring 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.

Performance levels of the calibration certificates

Different "qualities" of a calibration certificate are common:

- Beckhoff calibration certificates
  Such IP20 terminals can be usually identified by the product suffix -0020. The certificate is issued in Beckhoff production as PDF.
  The terminals can be obtained from Beckhoff and recalibrated by the Beckhoff service department.
- ISO17025 calibration certificates
  Such IP20 terminals can be usually identified by the product suffix -0030. The certificate is issued by a service provider on behalf of Beckhoff as part of Beckhoff production and delivered by Beckhoff as a PDF.
  The terminals can be obtained from Beckhoff and recalibrated by the Beckhoff service department.
- DAkkS calibration certificates (German: "Deutsche Akkreditierungsstelle GmbH")
  Such IP20 terminals can be usually identified by the product suffix -0030. The certificate is issued by a accredited service provider on behalf of Beckhoff as a part of Beckhoff production and delivered by Beckhoff as a PDF.
  The terminals can be obtained from Beckhoff and recalibrated by the Beckhoff service department.

Unique device number

Depending on the device, the following numbers are used for identification:

- EL/ELM terminals up to year of manufacture 2020: the ID number which is lasered on the side.

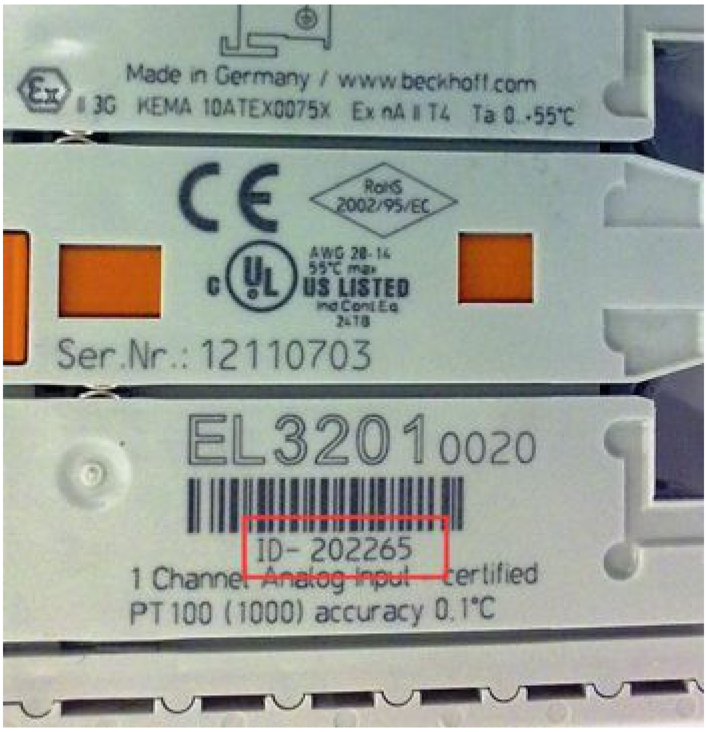

Made in Germany / www.beckhoff.com
II 3G KEMA 10ATEX0075X Ex nA II T4 Ta 0..+55°C
CE
RoHS 2002/95/EC
AWG 28-14 55°C max
c UL US LISTED
Ind.Cont.Eq. 24TB
Ser.Nr.: 12110703
EL3201 0020
ID- 202265
1 Channel Analog Input certified
PT100 (1000) accuracy 0.1°C

Fig. 6: ID number

- From year of manufacture 2021 onwards, the BTN number (Beckhoff Traceability Number) will gradually replace the ID number, this is also lasered on the side.

Beckhoff produces a wide range of analog input/output devices as IP20 terminal or IP67 box. A selection of these is also available with factory/ISO/DAkkS calibration certificates. For specific details and availability, see the technical data of the devices or contact Beckhoff Sales.

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*”.

2.5 Start

For commissioning:

- Install the EL3692 as described in chapter Mounting and wiring [▶ 34].
- configure the EL3692 in TwinCAT or another EtherCAT Master as described in the chapter Commissioning [▶ 54].

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

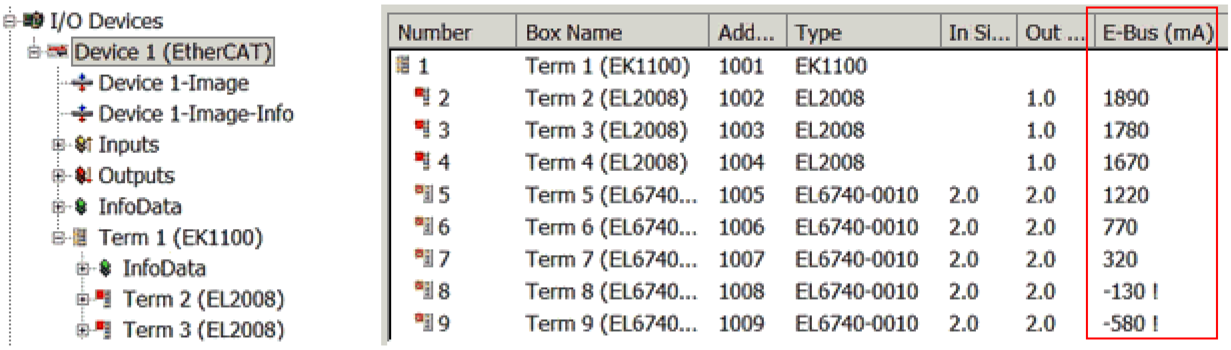
It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

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

E-Bus supply

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

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



- I/O Devices
  - Device 1 (EtherCAT)
    - Device 1-Image
    - Device 1-Image-Info
    - Inputs
    - Outputs
    - InfoData
    - Term 1 (EK1100)
      - InfoData
      - Term 2 (EL2008)
      - Term 3 (EL2008)

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 !
9	Term 9 (EL6740...	1009	EL6740-0010	2.0	2.0	-580 !

Fig. 7: System manager current calculation

NOTICE

Malfunction possible!

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

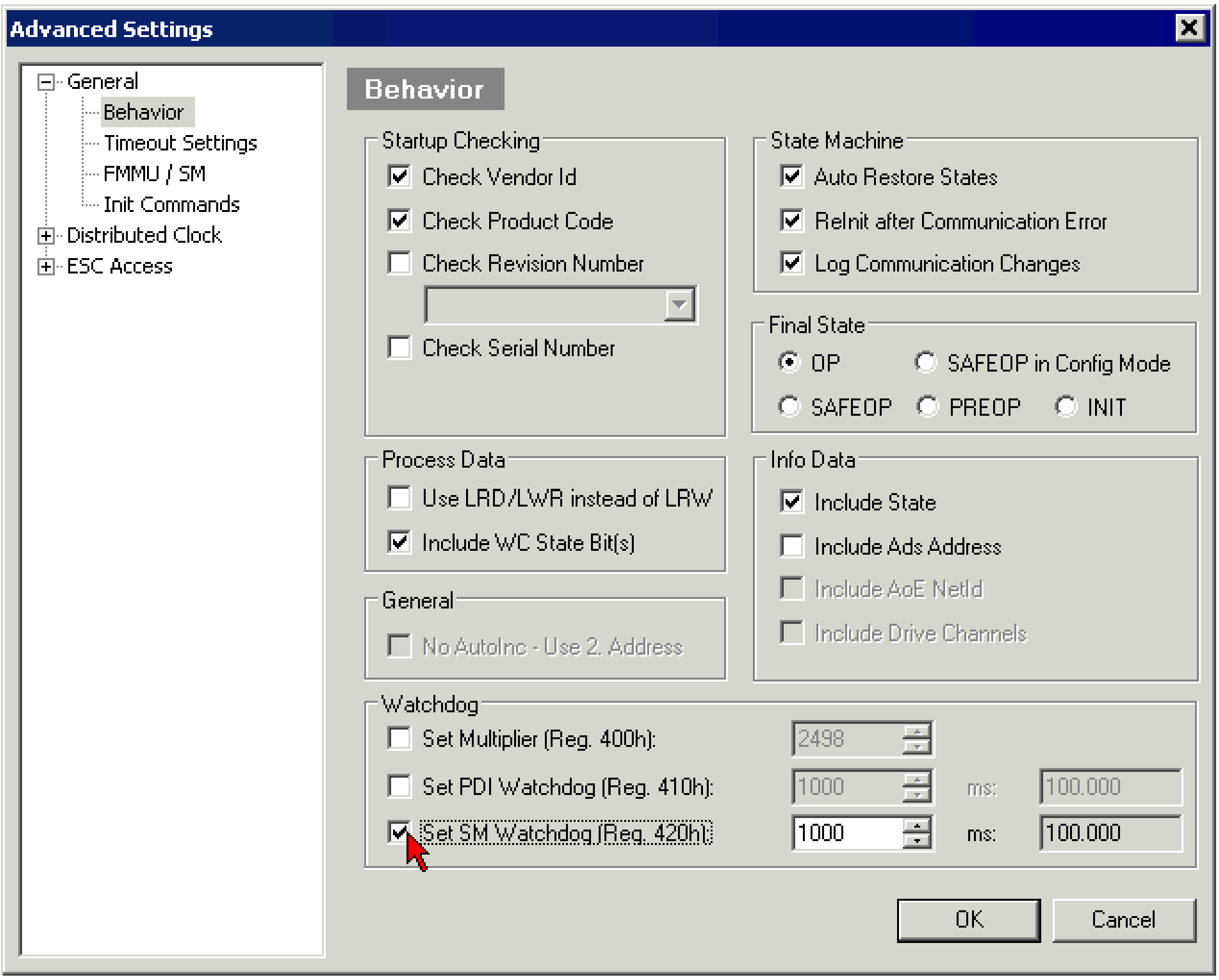
3.3 General notes for setting the watchdog

The EtherCAT terminals are equipped with a safety device (watchdog) which, e. g. in the event of interrupted process data traffic, switches the outputs (if present) to a presettable state after a presettable time, depending on the device and setting, e. g. to FALSE (off) or an output value.

The EtherCAT slave controller (ESC) features two watchdogs:

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

Their times are individually parameterized in TwinCAT as follows:



Behavior

Startup Checking
☑ Check Vendor Id
☑ Check Product Code
☐ Check Revision Number
☐ Check Serial Number

State Machine
☑ Auto Restore States
☑ ReInit after Communication Error
☑ Log Communication Changes

Final State
● OP ○ SAFEOP in Config Mode
○ SAFEOP ○ PREOP ○ INIT

Process Data
☐ Use LRD/LWR instead of LRW
☑ Include WC State Bit(s)

Info Data
☑ Include State
☐ Include Ads Address
☐ Include AoE NetId
☐ Include Drive Channels

General
☐ No AutoInc - Use 2. Address

Watchdog
☐ Set Multiplier (Reg. 400h): 2498
☐ Set PDI Watchdog (Reg. 410h): 1000 ms: 100.000
☑ Set SM Watchdog (Reg. 420h): 1000 ms: 100.000

Fig. 8: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.
- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers x0400/0410/0420: ESC Access -> Memory

SM watchdog (SyncManager Watchdog)

The SyncManager watchdog is reset with each successful EtherCAT process data communication with the terminal. If, for example, no EtherCAT process data communication with the terminal takes place for longer than the set and activated SM watchdog time due to a line interruption, the watchdog is triggered. The status of the terminal (usually OP) remains unaffected. The watchdog is only reset again by a successful EtherCAT process data access.

The SyncManager watchdog is therefore a monitoring for correct and timely process data communication with the ESC from the EtherCAT side.

The maximum possible watchdog time depends on the device. For example, for "simple" EtherCAT slaves (without firmware) with watchdog execution in the ESC it is usually up to 170 seconds. For complex EtherCAT slaves (with firmware) the SM watchdog function is usually parameterized via Reg. 400/420 but executed by the µC and can be significantly lower. In addition, the execution may then be subject to a certain time uncertainty. Since the TwinCAT dialog may allow inputs up to 65535, a test of the desired watchdog time is recommended.

PDI watchdog (Process Data Watchdog)

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

PDI (Process Data Interface) is the internal interface of the ESC, e.g. to local processors in the EtherCAT slave. With the PDI watchdog this communication can be monitored for failure.

The PDI watchdog is therefore a monitoring for correct and timely process data communication with the ESC, but viewed from the application side.

Calculation

Watchdog time = [1/25 MHz * (Watchdog multiplier + 2)] * PDI/SM watchdog

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

The value in Multiplier + 2 corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

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

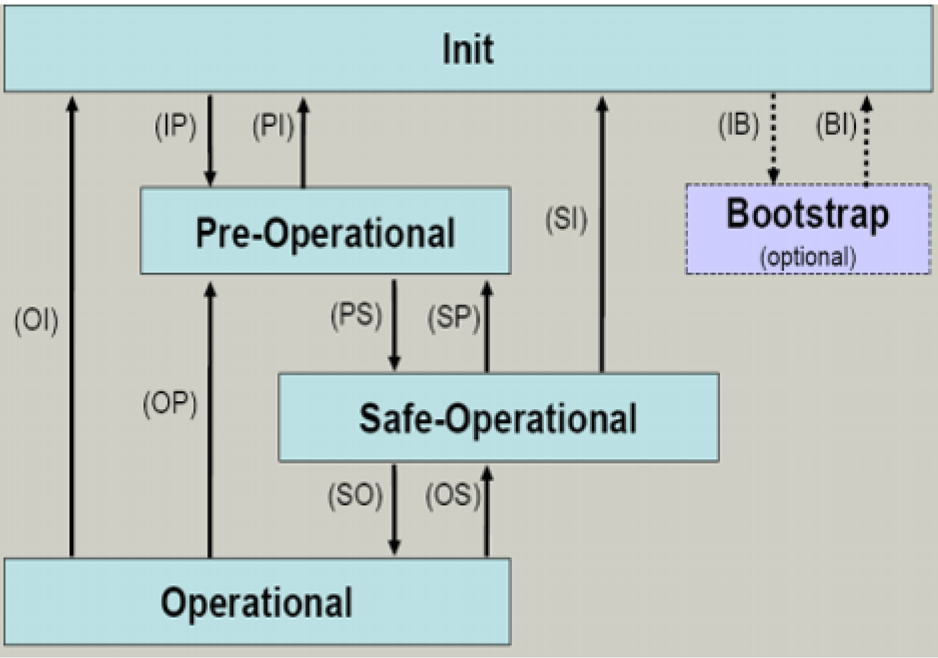


Fig. 9: States of the EtherCAT State Machine

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

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

Operational (Op)

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

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

Boot

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

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

3.5 CoE Interface

General description

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

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

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

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

- Index: 0x0000 …0xFFFF (0...65535$_{\text{dec}}$)
- SubIndex: 0x00…0xFF (0...255$_{\text{dec}}$)

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

The relevant ranges for EtherCAT fieldbus users are:

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

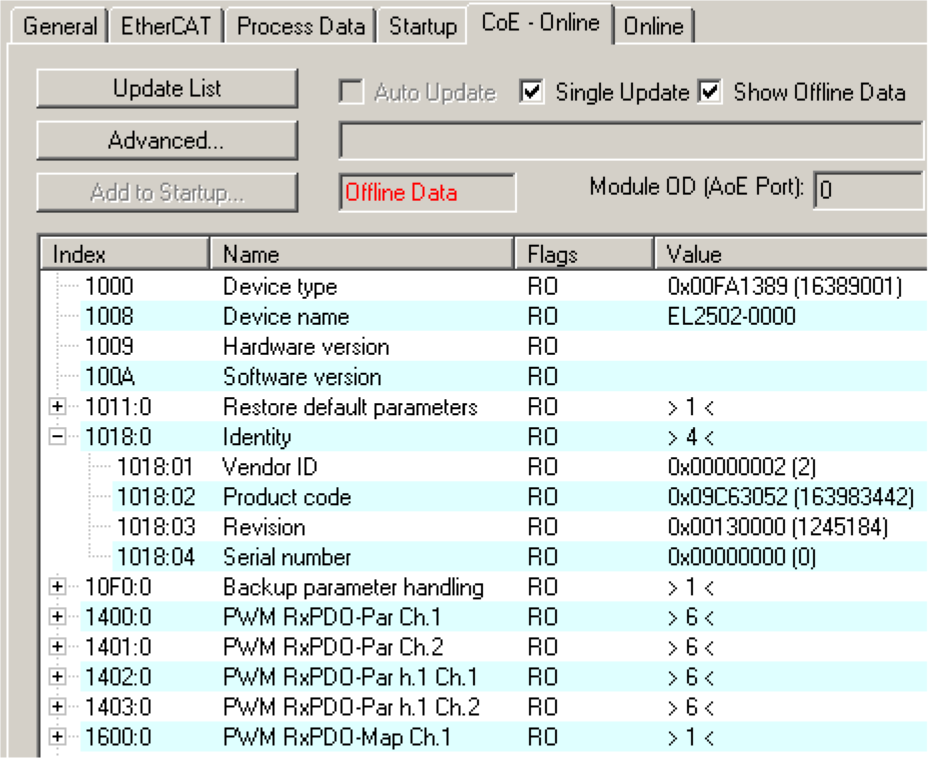
Other important ranges are:

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



□ Auto Update ☑ Single Update ☑ Show Offline Data

Offline Data

Module OD (AoE Port): 0

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

Fig. 10: “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.

NOTICE

Changes in the CoE directory (CAN over EtherCAT), program access

When using/manipulating the CoE parameters observe the general CoE notes in chapter "CoE interface" of the EtherCAT system documentation:

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the Beckhoff website),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see TwinCAT3 | PLC Library: Tc2_EtherCAT and Example program R/W CoE)

Data management and function “NoCoeStorage”

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

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

Data management

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

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

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

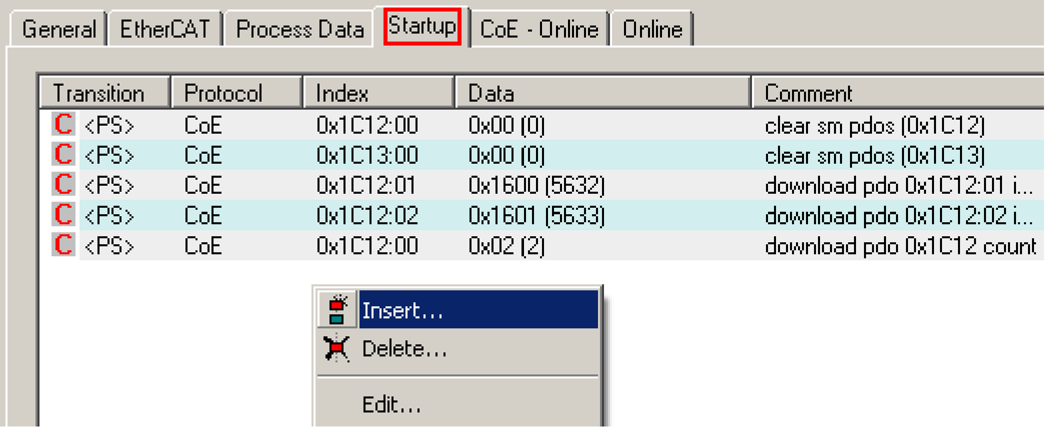
Startup list

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.



Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1C12:00	0x00 (0)	clear sm pdos (0x1C12)
C <PS>	CoE	0x1C13:00	0x00 (0)	clear sm pdos (0x1C13)
C <PS>	CoE	0x1C12:01	0x1600 (5632)	download pdo 0x1C12:01 i...
C <PS>	CoE	0x1C12:02	0x1601 (5633)	download pdo 0x1C12:02 i...
C <PS>	CoE	0x1C12:00	0x02 (2)	download pdo 0x1C12 count

Fig. 11: Startup list in the TwinCAT System Manager

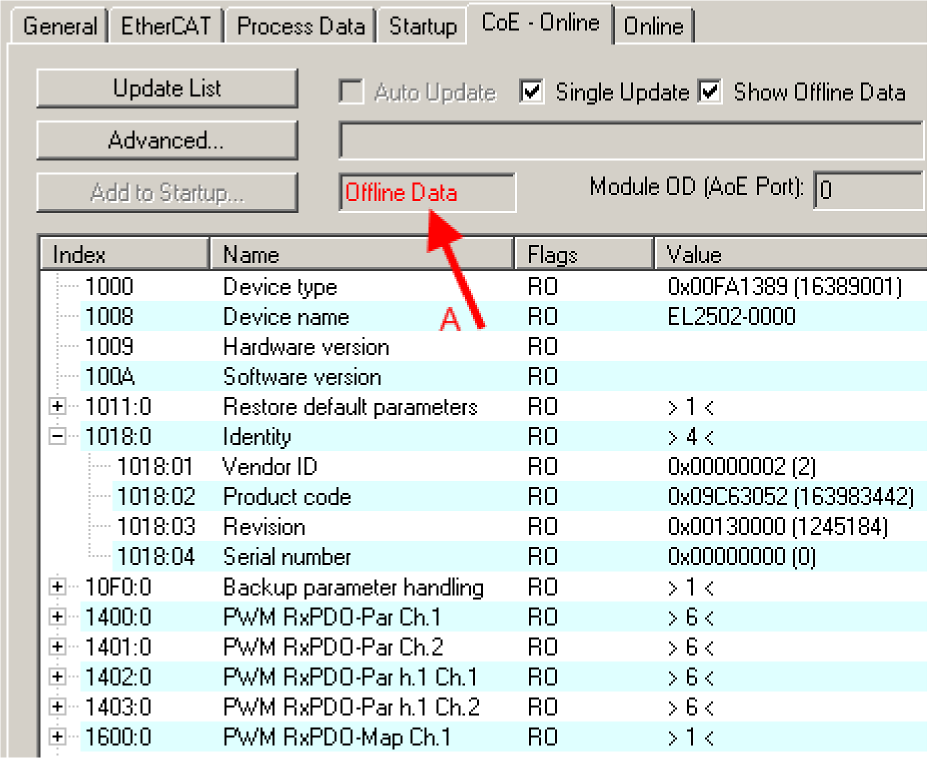
The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can be created.

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.



□ Auto Update ☑ Single Update ☑ Show Offline Data

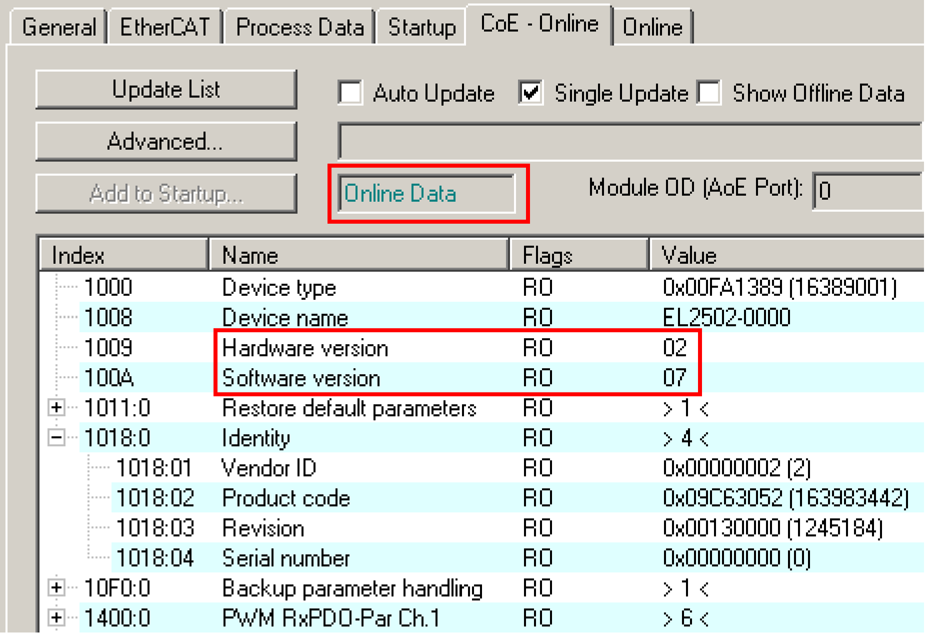
Offline Data

Module OD (AoE Port): 0

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

Fig. 12: 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.



☐ Auto Update ☑ Single Update ☐ Show Offline Data

Online Data

Module OD (AoE Port): 0

Index	Name	Flags	Value
1000	Device type	RO	0x00FA1389 (16389001)
1008	Device name	RO	EL2502-0000
1009	Hardware version	RO	02
100A	Software version	RO	07
⊞ 1011:0	Restore default parameters	RO	> 1 <
⊟ 1018:0	Identity	RO	> 4 <
1018:01	Vendor ID	RO	0x00000002 (2)
1018:02	Product code	RO	0x09C63052 (163983442)
1018:03	Revision	RO	0x00130000 (1245184)
1018:04	Serial number	RO	0x00000000 (0)
⊞ 10F0:0	Backup parameter handling	RO	> 1 <
⊞ 1400:0	PWM RxPDO-Par Ch.1	RO	> 6 <

Fig. 13: Online list

Channel-based order

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

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

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

This is generally written as 0x80n0.

Detailed information on the CoE interface can be found in the EtherCAT system documentation on the Beckhoff website.

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 Instructions for ESD protection

NOTICE

Destruction of the devices by electrostatic discharge possible!

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

- 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 by 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.

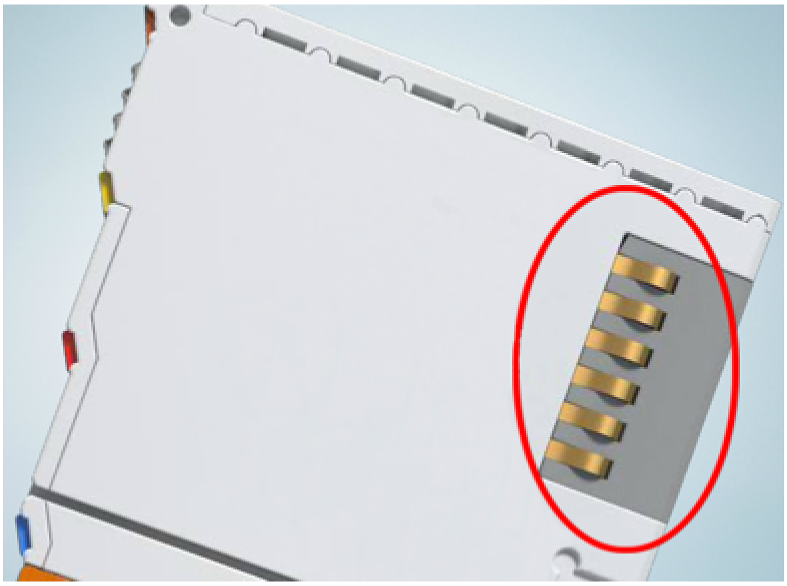


Fig. 14: Spring contacts of the Beckhoff I/O components

4.2 Explosion protection

4.2.1 ATEX - Special conditions (standard temperature range)

⚠ **WARNING**

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

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

Standards

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

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

Marking

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

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

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

or

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

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

4.2.2 Continuative documentation for ATEX and IECEx

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

4.3 UL notice

⚠ CAUTION

Application

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

⚠ CAUTION

Examination

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

⚠ CAUTION

For devices with Ethernet connectors

Not for connection to telecommunication circuits.

Basic principles

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

4.4 Installation on mounting rails

⚠ **WARNING**

Risk of electric shock and damage of device!

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

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

Assembly

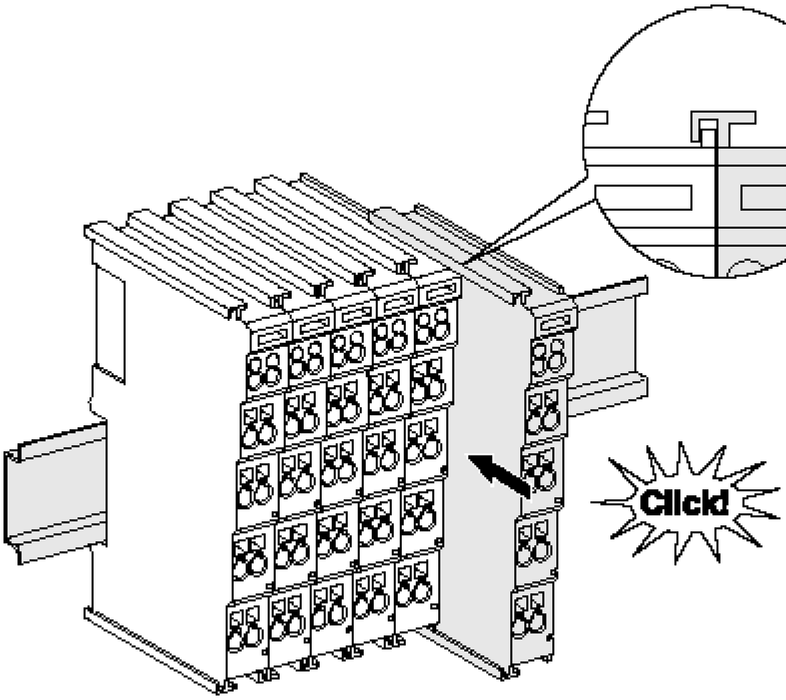


Fig. 15: Attaching on mounting rail

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

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

Fixing of mounting rails

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

Disassembly

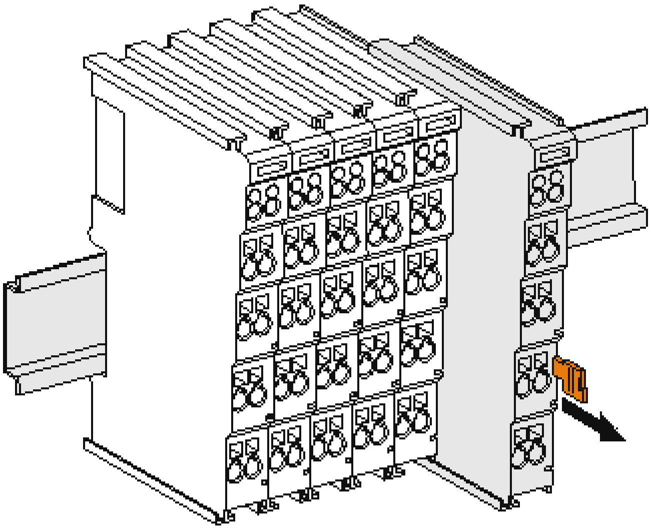


Fig. 16: Disassembling of terminal

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

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.

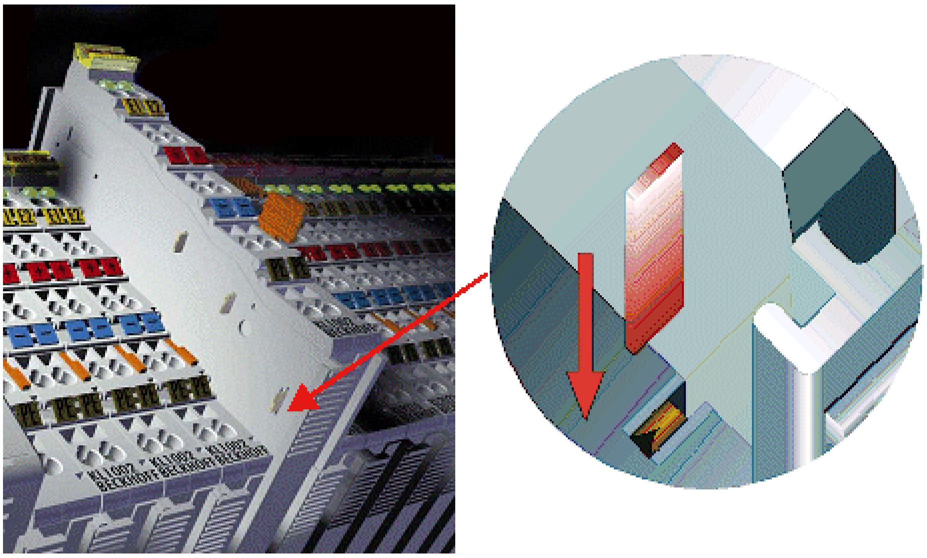


Fig. 17: Power contact on left side

NOTICE

Possible damage of the device

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

⚠ **WARNING**

Risk of electric shock!

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

4.5 Installation instructions for enhanced mechanical load capacity

⚠ **WARNING**

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

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

Additional checks

The terminals have undergone the following additional tests:

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

Additional installation instructions

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.

4.6 Connection

4.6.1 Connection system

⚠ WARNING

Risk of electric shock and damage of device!

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

Overview

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

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

Standard wiring (ELxxxx / KLxxxx)

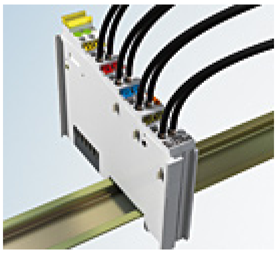


Fig. 18: Standard wiring

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)

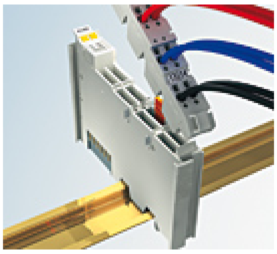


Fig. 19: Pluggable wiring

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

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

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

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

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

High Density Terminals (HD Terminals)

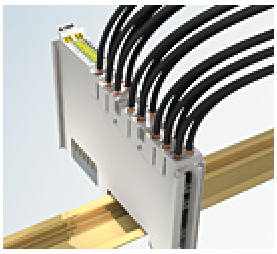


Fig. 20: High Density Terminals

The 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

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

Ultrasonically “bonded” (ultrasonically welded) conductors

Ultrasonically “bonded” conductors

It is also possible to connect the Standard and High Density Terminals with ultrasonically “bonded” (ultrasonically welded) conductors. In this case, please note the tables concerning the wire-size width [▶ 44]!

4.6.2 Wiring

⚠ WARNING

Risk of electric shock and damage of device!

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

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

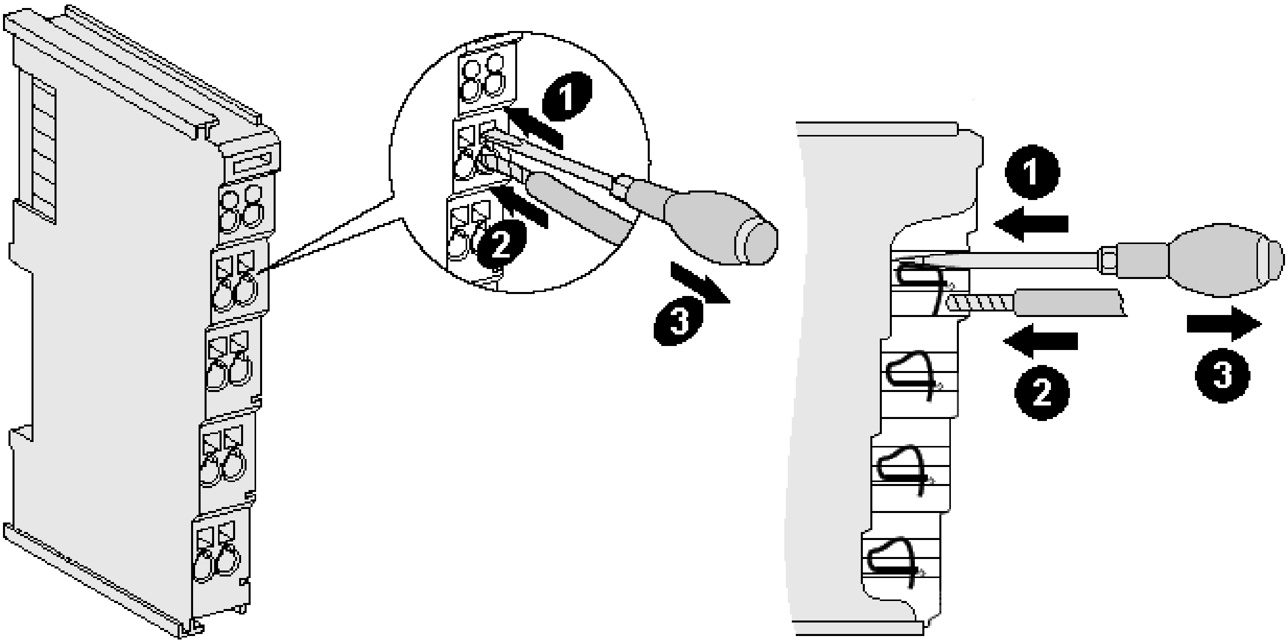


Fig. 21: Connecting a cable on a terminal point

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 [▶ 43]) 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 (see notice [▶ 43])
Wire stripping length	8 ... 9 mm

4.6.3 Shielding

Shielding

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

4.7 Installation positions

NOTICE

Constraints regarding installation position and operating temperature range

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

Optimum installation position (standard)

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

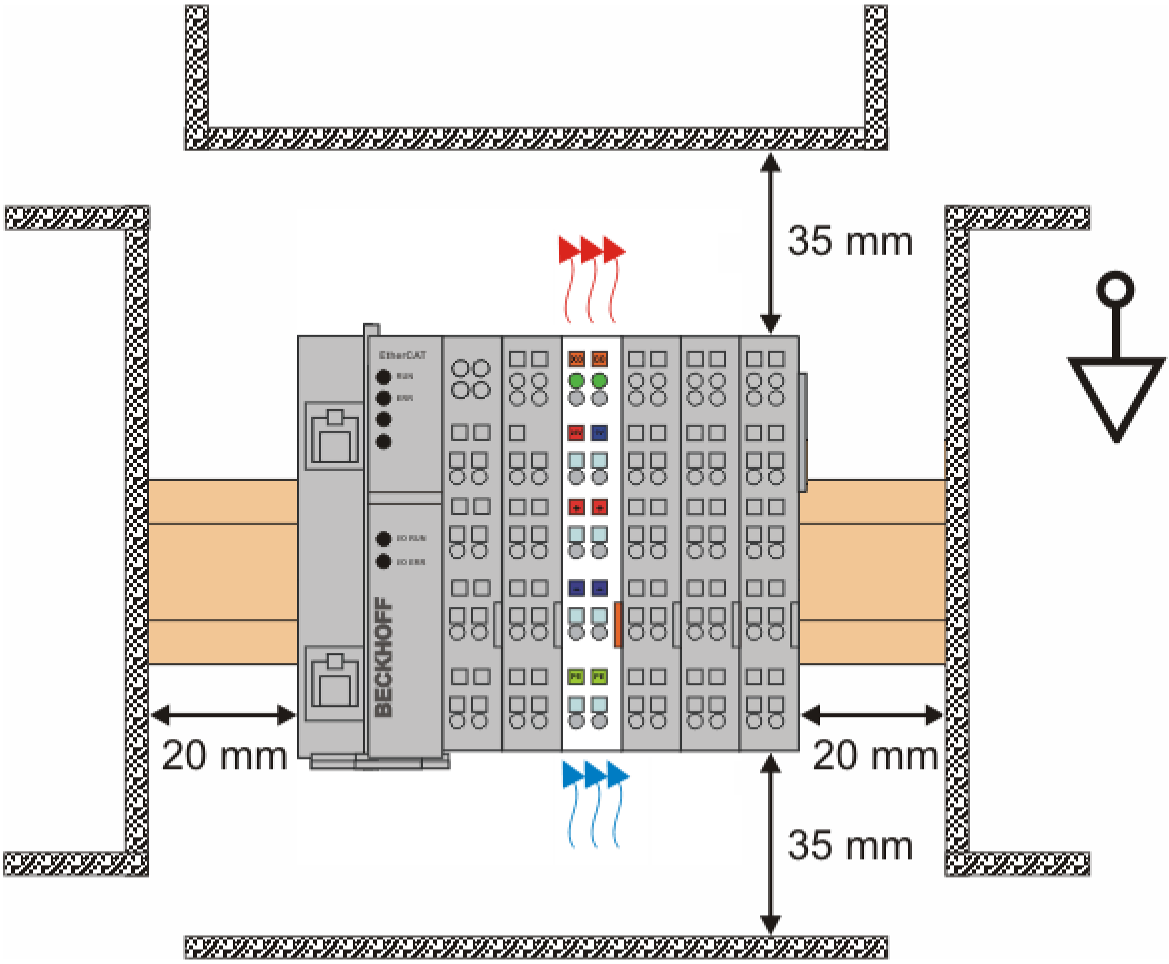


Fig. 22: Recommended distances for standard installation position

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.

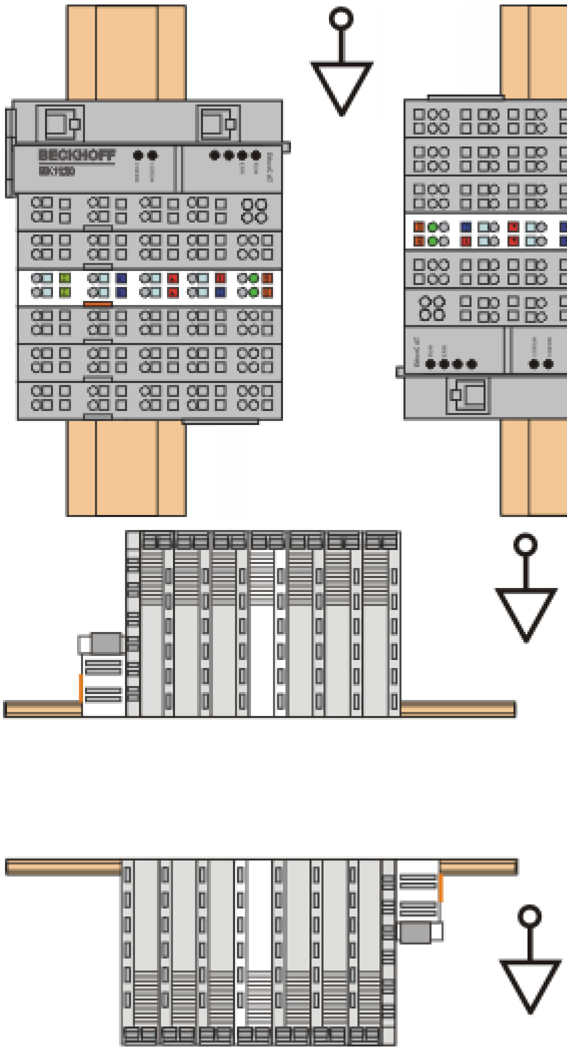


Fig. 23: Other installation positions

4.8 LEDs and connection

This section contains an overview of the terminal, followed by a table with a description of the connection at the terminal.
This is followed by a table with possible LED states.

The following figure shows the terminal overview including the connection options:

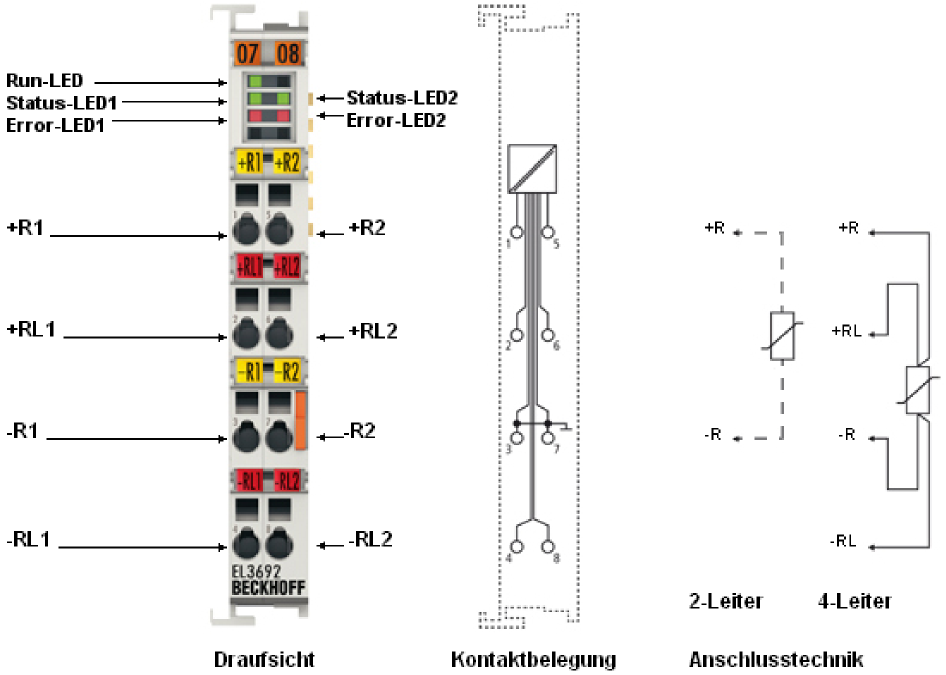


Fig. 24: Connection diagram

EL3692 pin assignment

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

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal
		flashing uniformly	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		flashing slowly	State of the EtherCAT State Machine: **SAFEOP** = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
		flashing rapidly	State of the EtherCAT State Machine: **BOOTSTRAP** = function for terminal firmware updates
STATUS1, STATUS2	green	The respective channel is carrying out a measurement; the test specimen is live.	
ERROR1, ERROR2	red	Wire breakage. The resistance is outside the valid range.	

4.9 Electrical connection

Recommended cables

In order to ensure trouble-free and low-noise measurement we recommend using shielded cables. The cable screen should be connected thoroughly with the DIN rail near the terminal.

2/4-wire connection

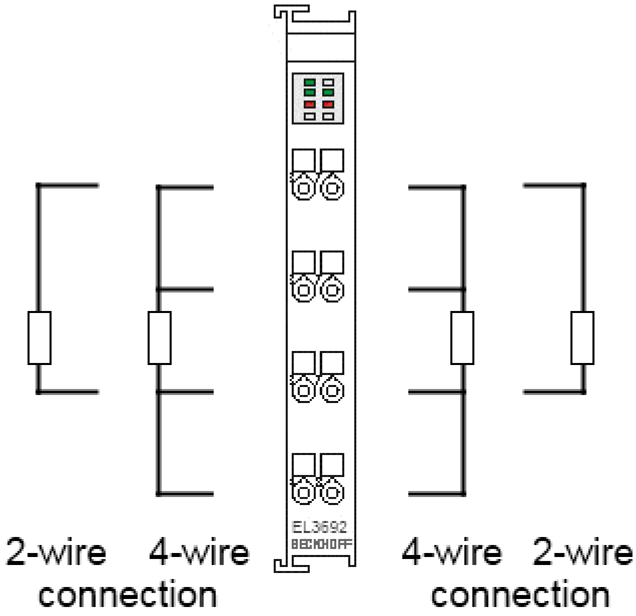


Fig. 25: Connection

Both channels of the EL3692 are in 4-wire mode as standard.

Measurements on live objects

Measurement of live test specimens (DuT) is only permitted if no current flows through the test specimen.

NOTICE

Otherwise the device may be damaged.

The two channels of the EL3692 are not electrically isolated. Measurements on live objects should therefore be carried out in single-channel mode.

The following figure shows three potential applications; only the first two cases are possible. In the last case either no measurement takes place since the terminal detects overvoltages at its inputs, or the measurement is erroneous if the voltage at the DuT is still within the measurement limits of the terminal.

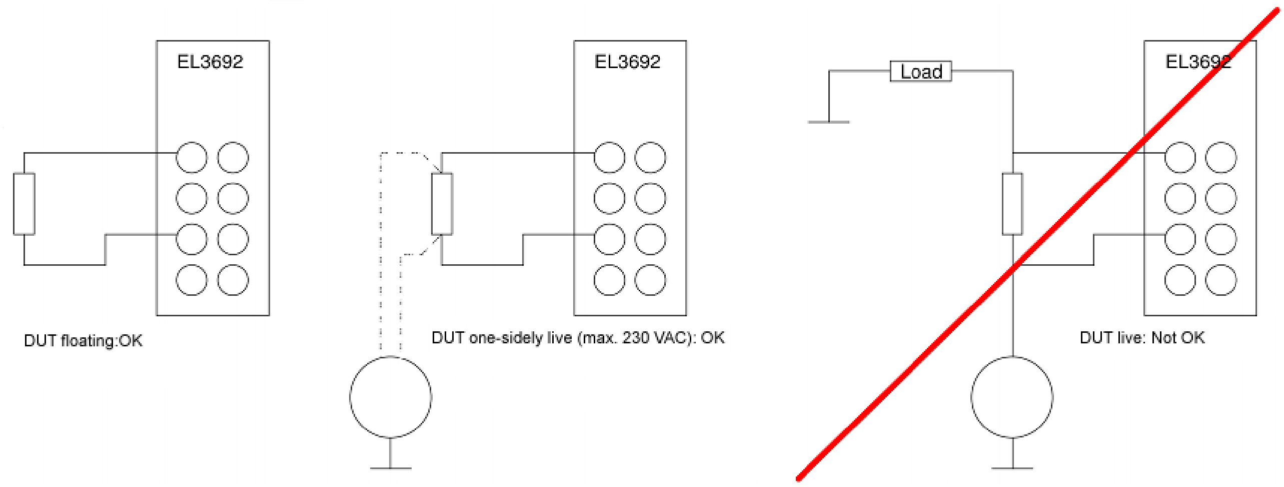


Fig. 26: Connection options

4.10 Positioning of passive Terminals

Hint for positioning of passive terminals in the bus terminal block

EtherCAT Terminals (ELxxxx / ESxxxx), which do not take an active part in data transfer within the bus terminal block are so called passive terminals. The passive terminals have no current consumption out of the E-Bus.
To ensure an optimal data transfer, you must not directly string together more than two passive terminals!

Examples for positioning of passive terminals (highlighted)

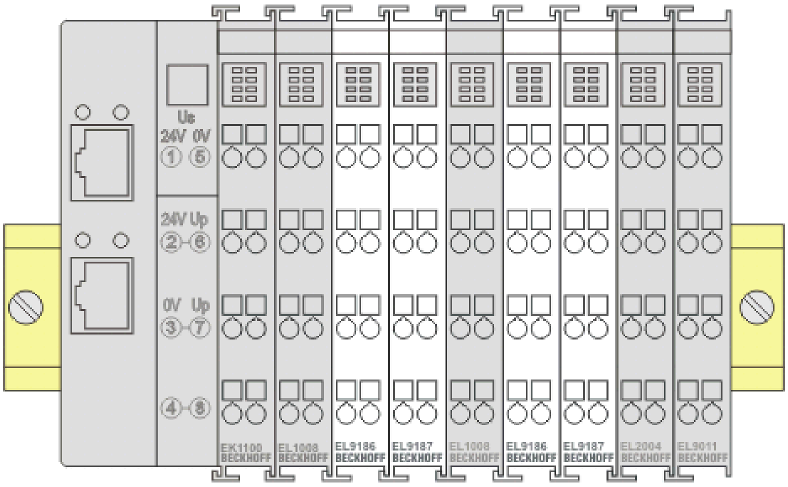


Fig. 27: Correct positioning

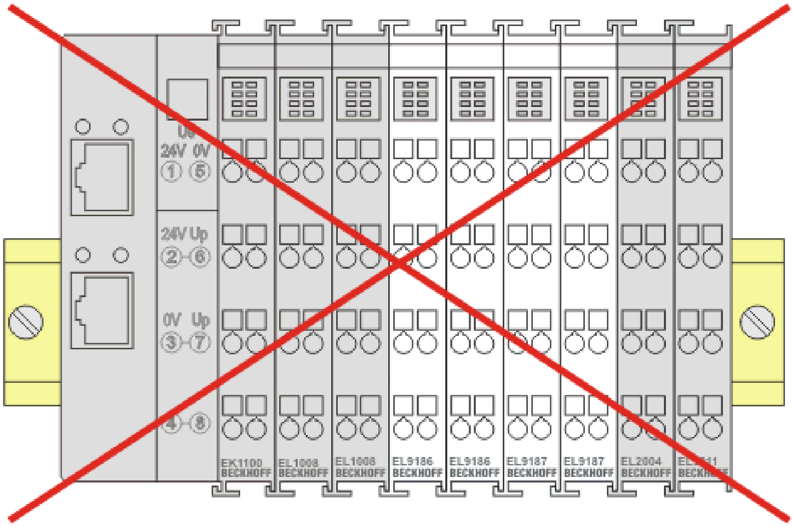


Fig. 28: Incorrect positioning

4.11 Disposal

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

5 Commissioning

5.1 TwinCAT Quick Start

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

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

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

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

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

The following relationship is envisaged between the user PC and individual control elements:

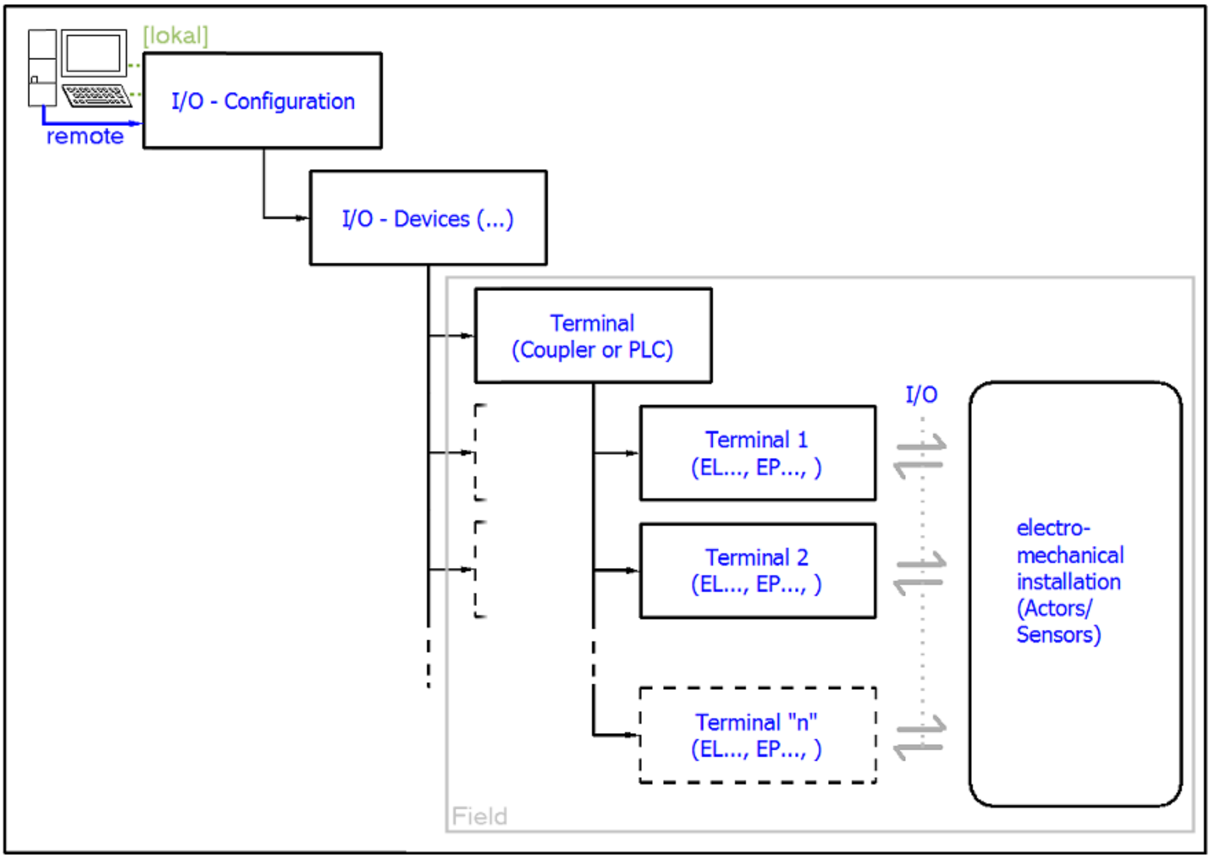


Fig. 29: Relationship between user side (commissioning) and installation

Insertion of certain components (I/O device, terminal, box...) by users functions the same way as in TwinCAT 2 and TwinCAT 3. The descriptions below relate solely to the online procedure.

Example configuration (actual configuration)

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

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

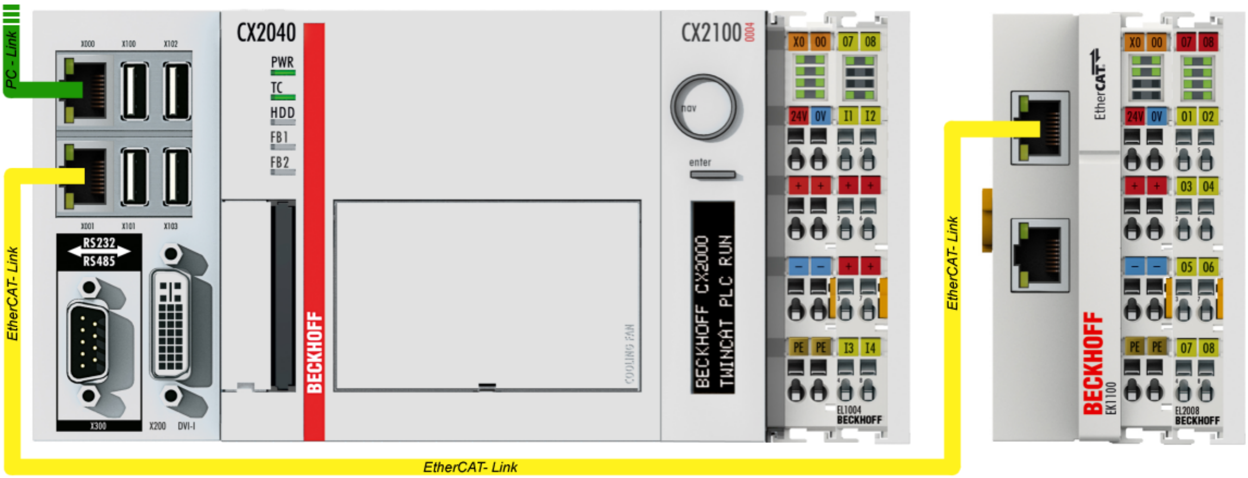


Fig. 30: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

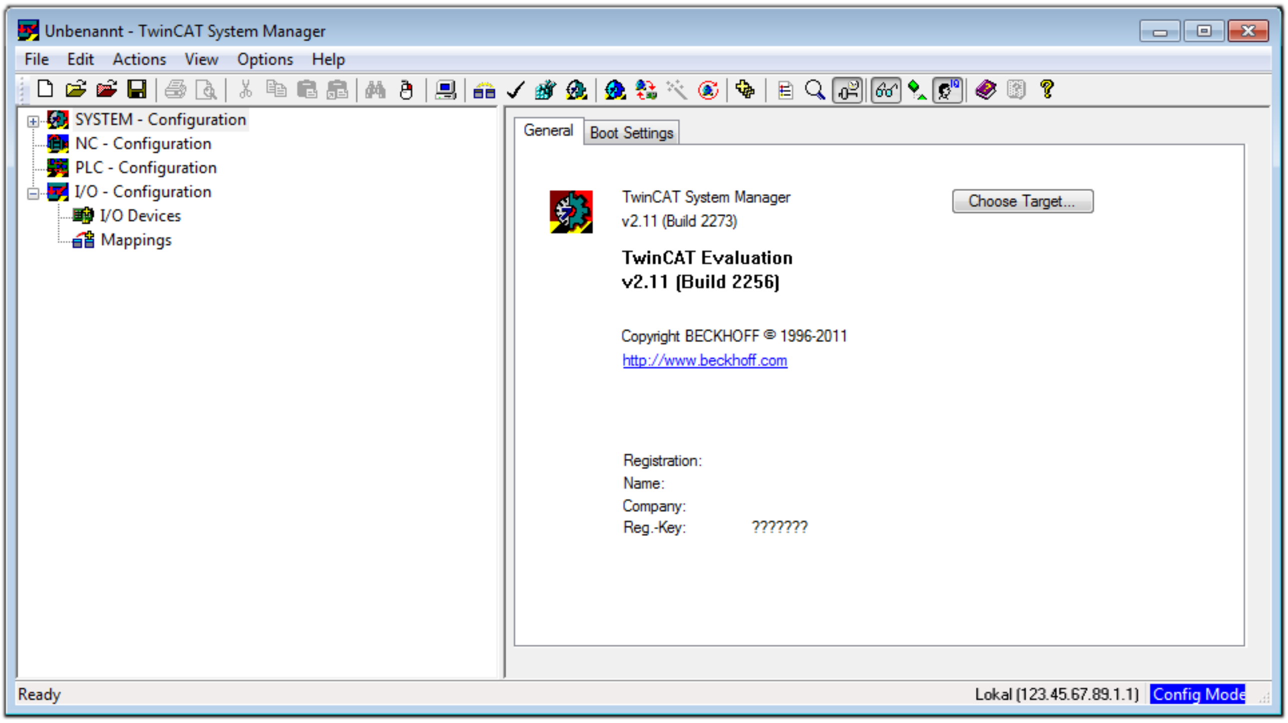
Note that all combinations of a configuration are possible; for example, the EL1004 terminal could also be connected after the coupler, or the EL2008 terminal could additionally be connected to the CX2040 on the right, in which case the EK1100 coupler wouldn't be necessary.

5.1.1 TwinCAT 2

Startup

TwinCAT 2 basically uses two user interfaces: the TwinCAT System Manager for communication with the electromechanical components and TwinCAT PLC Control for the development and compilation of a controller. The starting point is the TwinCAT System Manager.

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



TwinCAT System Manager
v2.11 (Build 2273)

TwinCAT Evaluation
v2.11 (Build 2256)

Copyright BECKHOFF © 1996-2011
http://www.beckhoff.com

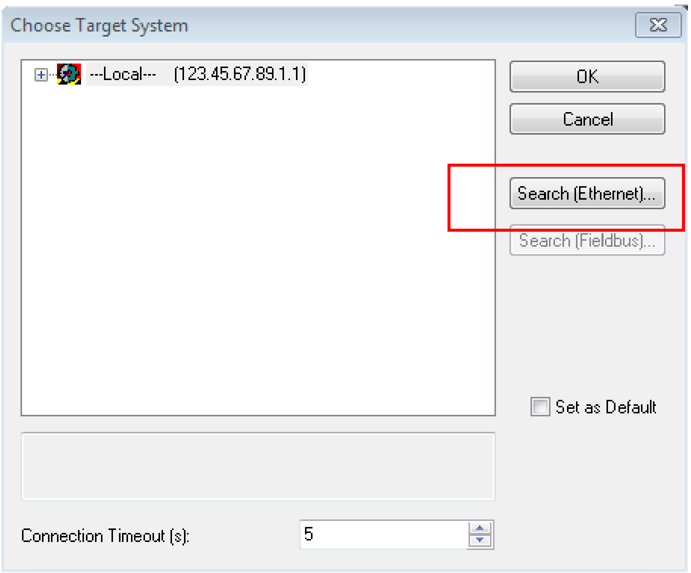
Registration:
Name:
Company:
Reg.-Key: ???????

Lokal (123.45.67.89.1.1)

Fig. 31: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system, including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thus the next step is “Insert Device [▶ 59]”.

If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. In the menu under “Actions” → “Choose Target System...”, the following window is opened for this via the symbol “ 🖥 ” or the “F8” key:



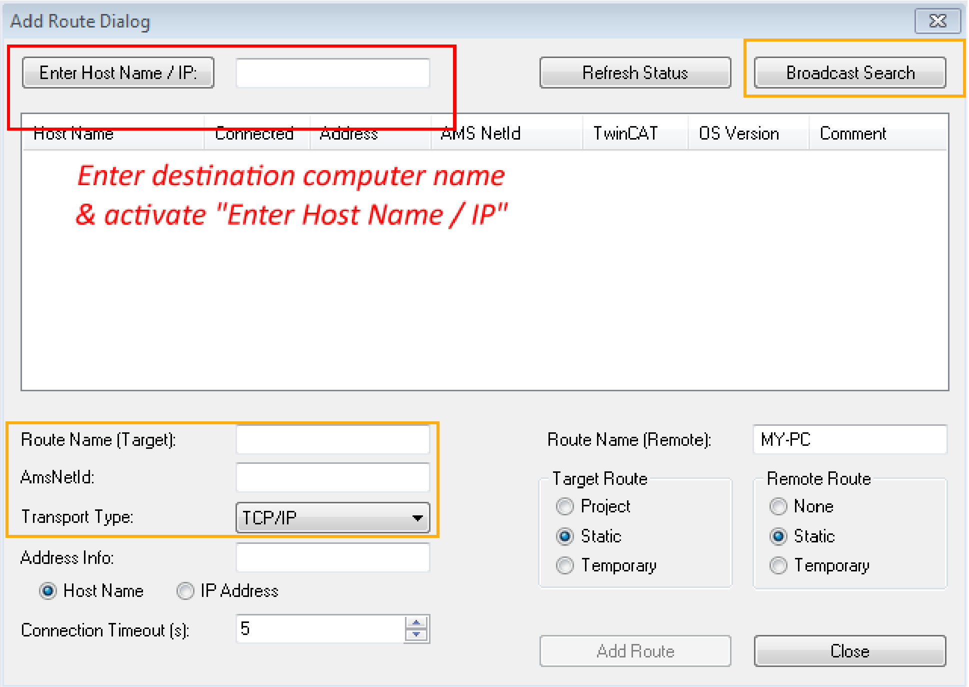
---Local--- (123.45.67.89.1.1)

Connection Timeout (s): 5

Fig. 32: Selection of the target system

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

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



Enter destination computer name
& activate "Enter Host Name / IP"

Route Name (Target):

AmsNetId:

Transport Type: TCP/IP

Address Info:

● Host Name ○ IP Address

Connection Timeout (s): 5

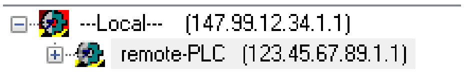
Route Name (Remote): MY-PC

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

Fig. 33: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



---Local--- (147.99.12.34.1.1)
remote-PLC (123.45.67.89.1.1)

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

Adding devices

In the configuration tree of the TwinCAT 2 System Manager user interface on the left, select “I/O Devices” and then right-click to open a context menu and select “Scan Devices…”, or start the action in the menu bar via . The TwinCAT System Manager may first have to be set to “Config Mode” via or via the menu
“Actions” → “Set/Reset TwinCAT to Config Mode…” (Shift + F4).

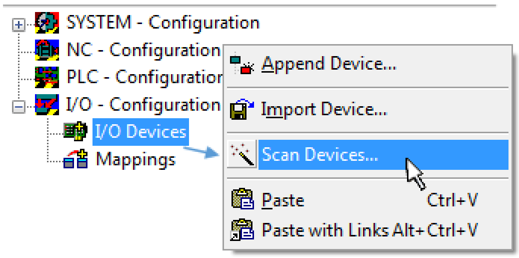
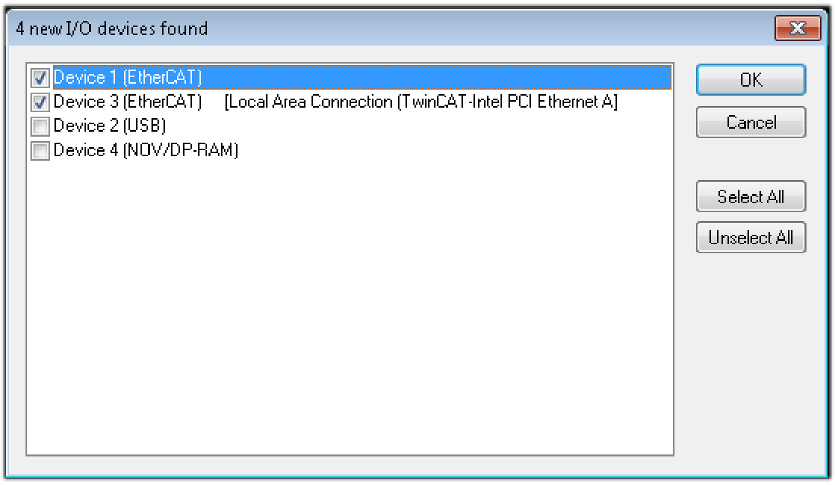


Fig. 34: Select “Scan Devices...”

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


4 new I/O devices found

☑ Device 1 (EtherCAT)
☑ Device 3 (EtherCAT) [Local Area Connection (TwinCAT-Intel PCI Ethernet A]
☐ Device 2 (USB)
☐ Device 4 (NOV/DP-RAM)

Fig. 35: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the example configuration [▸ 55] described at the beginning of this section, the result is as follows:

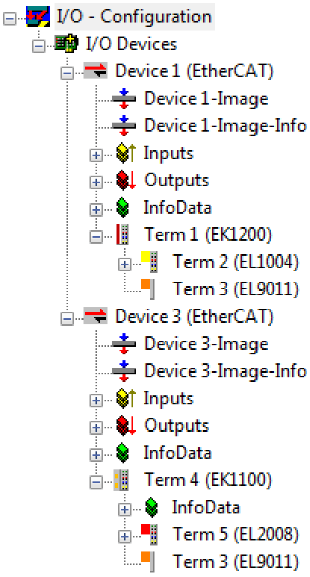


Fig. 36: Mapping of the configuration in the TwinCAT 2 System Manager

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

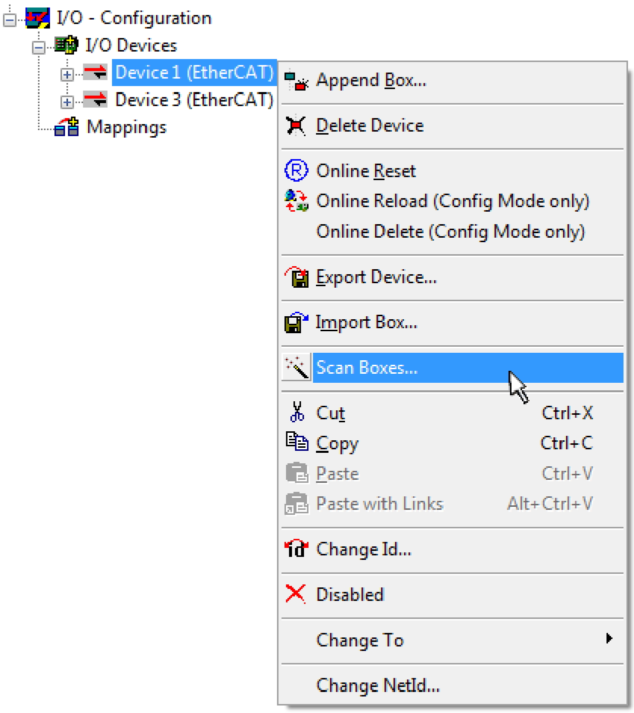


Fig. 37: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

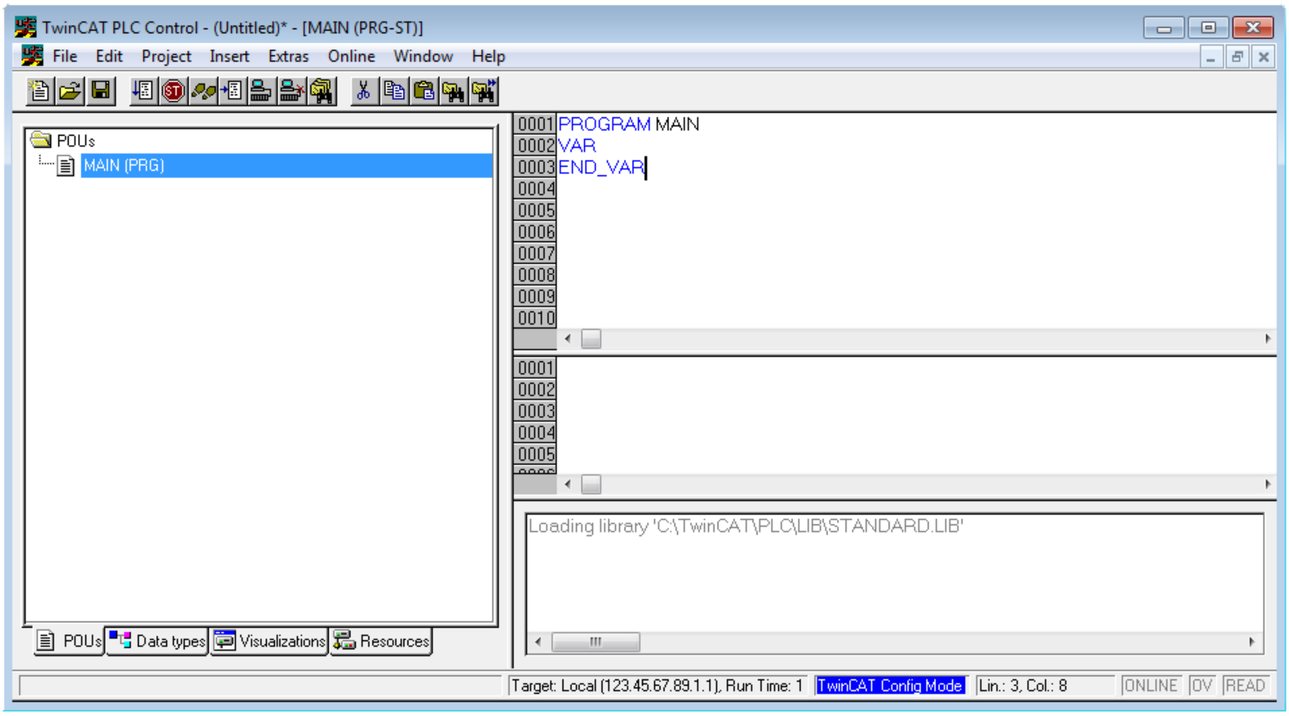
TwinCAT PLC Control is the development environment for generating the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

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

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

The following section refers solely to Structured Text (ST).

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



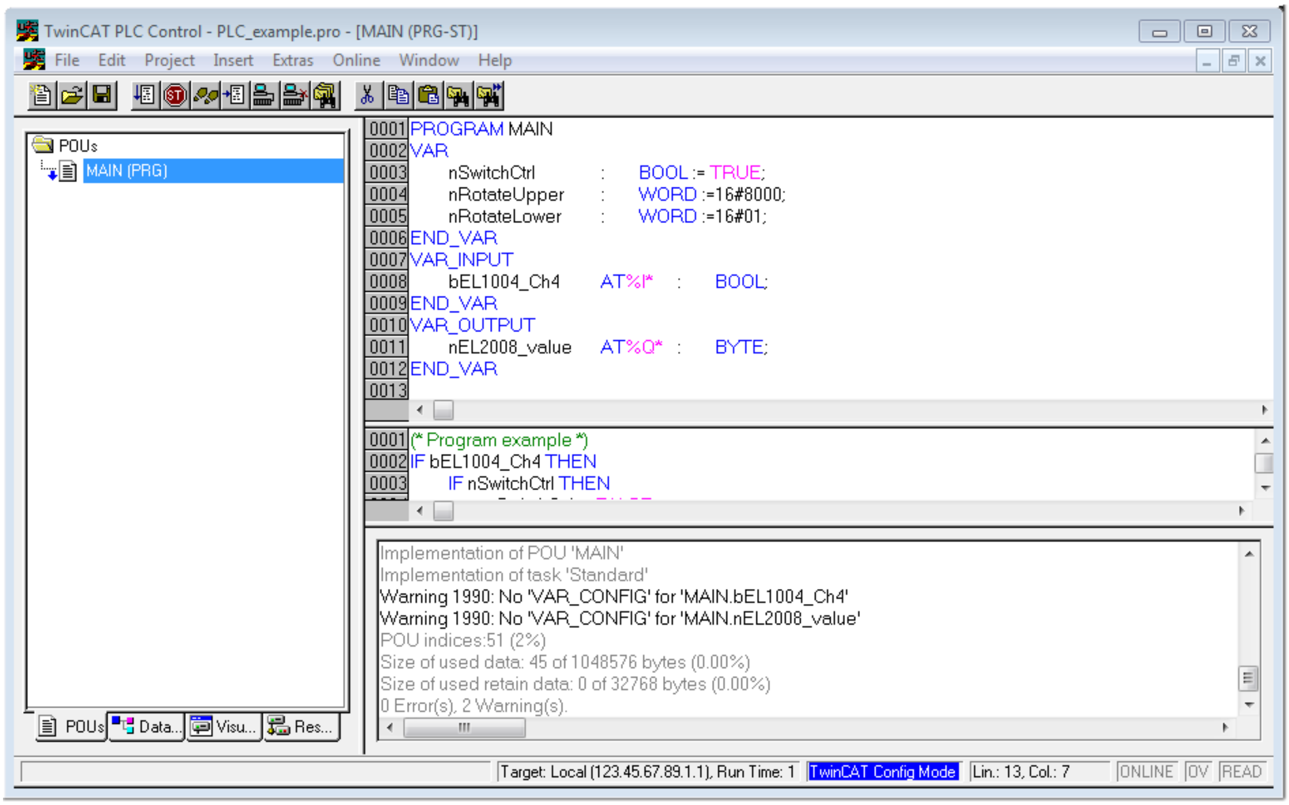
```
PROGRAM MAIN
VAR
END_VAR
```

Loading library 'C:\TwinCAT\PLC\LIB\STANDARD.LIB'

Target: Local (123.45.67.89.1.1), Run Time: 1 | TwinCAT Config Mode | Lin.: 3, Col.: 8 | ONLINE | OV | READ

Fig. 38: TwinCAT PLC Control after startup

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


TwinCAT PLC Control - PLC_example.pro - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs
Data...
Visu...
Res...

```
PROGRAM MAIN
VAR
        nSwitchCtrl         :     BOOL := TRUE;
        nRotateUpper        :     WORD :=16#8000;
        nRotateLower        :     WORD :=16#01;
END_VAR
VAR_INPUT
        bEL1004_Ch4       AT%I*    :     BOOL;
END_VAR
VAR_OUTPUT
        nEL2008_value     AT%Q*    :     BYTE;
END_VAR
```

```
(* Program example *)
IF bEL1004_Ch4 THEN
        IF nSwitchCtrl THEN
```

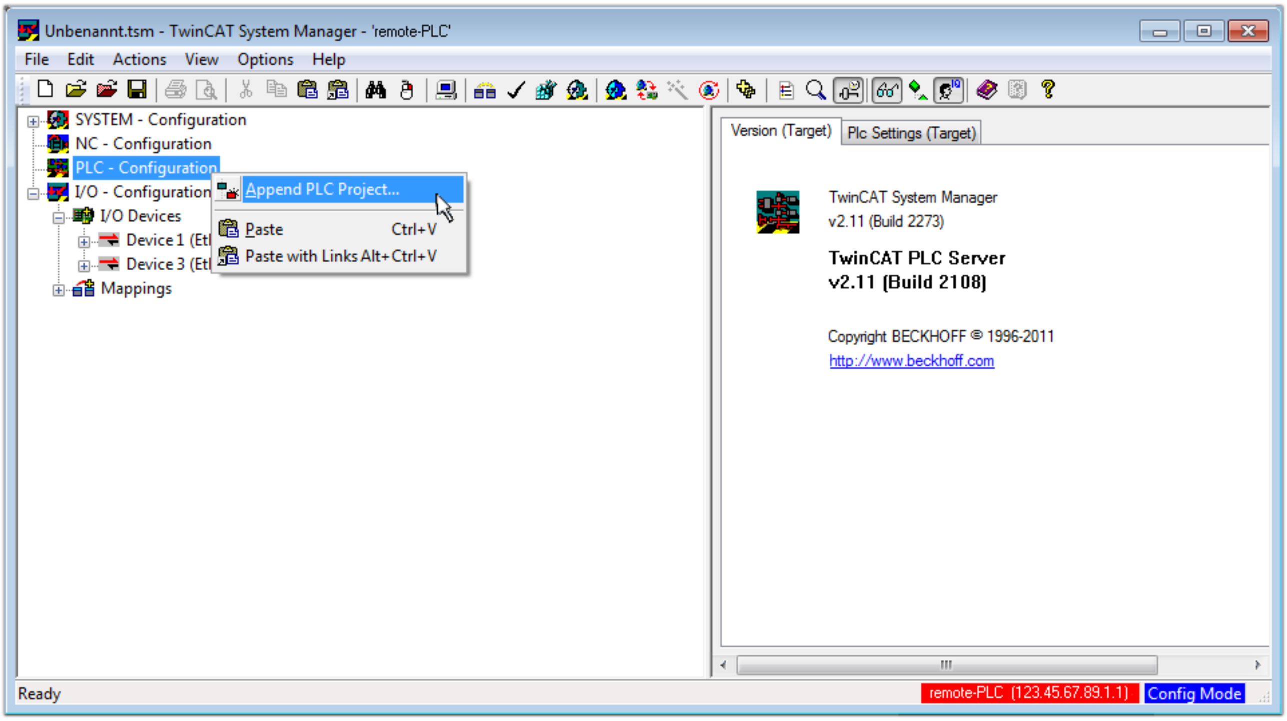
```
Implementation of POU 'MAIN'
Implementation of task 'Standard'
Warning 1990: No 'VAR_CONFIG' for 'MAIN.bEL1004_Ch4'
Warning 1990: No 'VAR_CONFIG' for 'MAIN.nEL2008_value'
POU indices:51 (2%)
Size of used data: 45 of 1048576 bytes (0.00%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 2 Warning(s).
```

Target: Local (123.45.67.89.1.1), Run Time: 1 | TwinCAT Config Mode | Lin.: 13, Col.: 7 | ONLINE | OV | READ

Fig. 39: Example program with variables after a compile process (without variable integration)

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

First, integrate the TwinCAT PLC Control project in the **System Manager**. This is performed via the context menu of the PLC configuration (right-click) and selecting “Append PLC Project…”:



TwinCAT System Manager
v2.11 (Build 2273)

TwinCAT PLC Server
v2.11 (Build 2108)

Copyright BECKHOFF © 1996-2011
http://www.beckhoff.com

Ready | remote-PLC (123.45.67.89.1.1) | Config Mode

Fig. 40: 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:

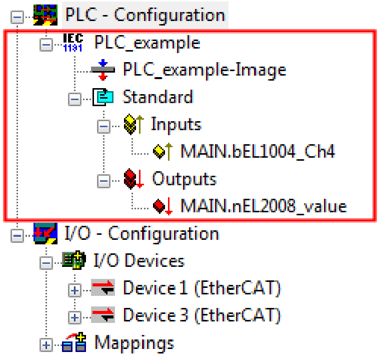
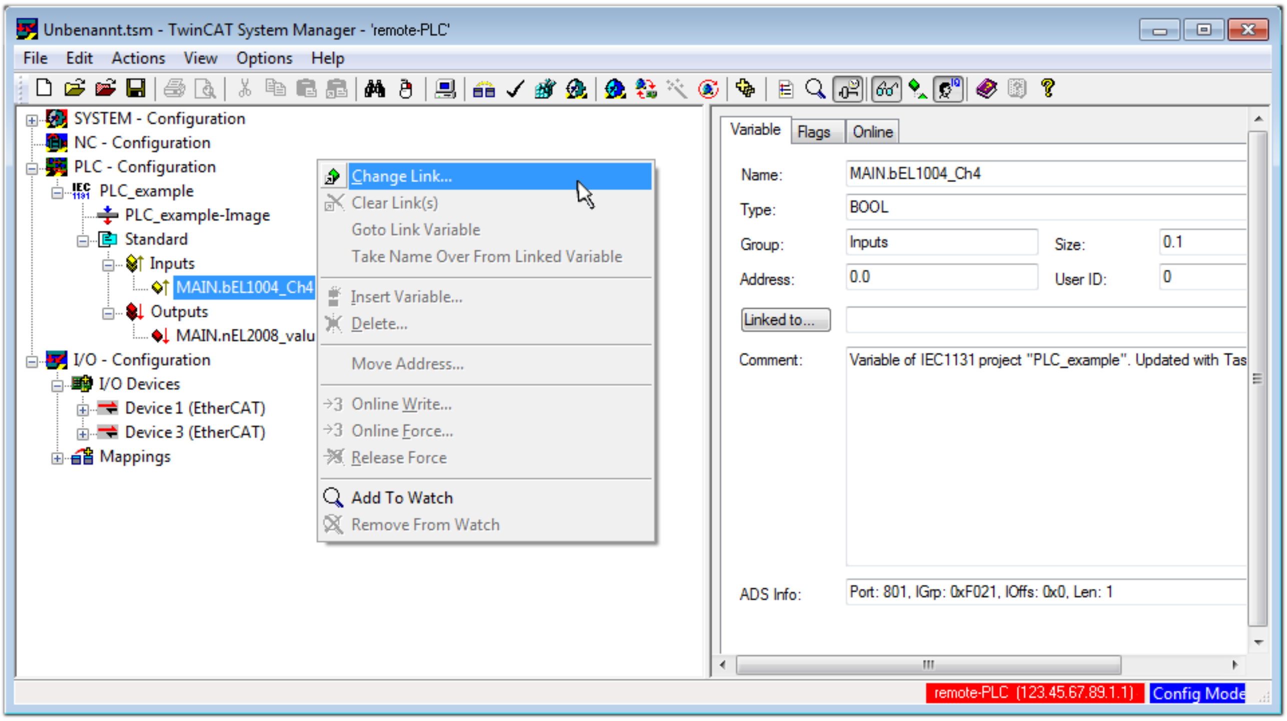


Fig. 41: PLC project integrated in the PLC configuration of the System Manager

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”:



Name: MAIN.bEL1004_Ch4

Type: BOOL

Group: Inputs Size: 0.1

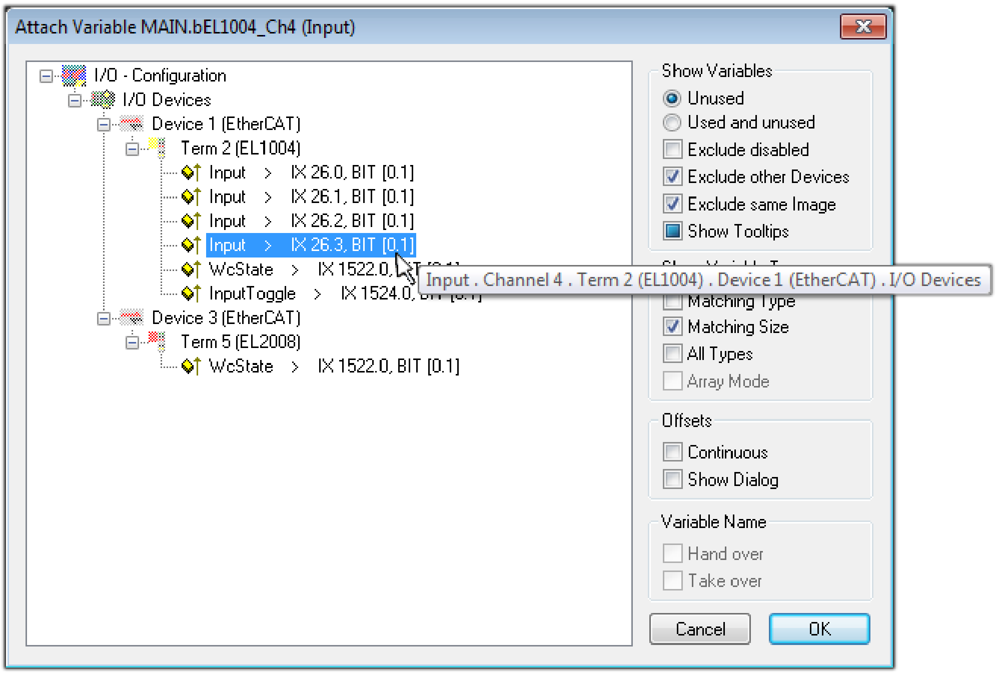
Address: 0.0 User ID: 0

Comment: Variable of IEC1131 project "PLC_example". Updated with Tas

ADS Info: Port: 801, IGrp: 0xF021, IOffs: 0x0, Len: 1

Fig. 42: Creating the links between PLC variables and process objects

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



Attach Variable MAIN.bEL1004_Ch4 (Input)

```
I/O - Configuration
  I/O Devices
    Device 1 (EtherCAT)
      Term 2 (EL1004)
        Input  >  IX 26.0, BIT [0.1]
        Input  >  IX 26.1, BIT [0.1]
        Input  >  IX 26.2, BIT [0.1]
        Input  >  IX 26.3, BIT [0.1]
        WcState  >  IX 1522.0, BIT [0.1]
        InputToggle  >  IX 1524.0, BIT [0.1]
    Device 3 (EtherCAT)
      Term 5 (EL2008)
        WcState  >  IX 1522.0, BIT [0.1]
```

Input . Channel 4 . Term 2 (EL1004) . Device 1 (EtherCAT) . I/O Devices

Show Variables: ● Unused, ○ Used and unused, □ Exclude disabled, ☑ Exclude other Devices, ☑ Exclude same Image, ■ Show Tooltips

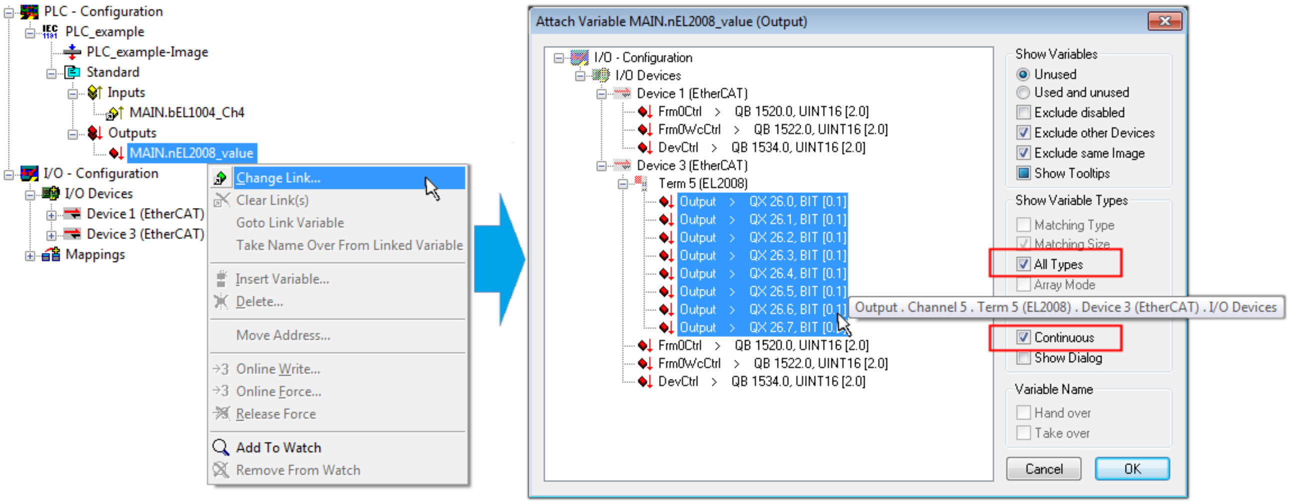
Show Variable Types: □ Matching Type, ☑ Matching Size, □ All Types, □ Array Mode

Offsets: □ Continuous, □ Show Dialog

Variable Name: □ Hand over, □ Take over

Fig. 43: Selecting BOOL-type PDO

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



```
PLC - Configuration
  PLC_example
    PLC_example-Image
    Standard
      Inputs
        MAIN.bEL1004_Ch4
      Outputs
        MAIN.nEL2008_value
I/O - Configuration
  I/O Devices
    Device 1 (EtherCAT)
    Device 3 (EtherCAT)
  Mappings
```

Attach Variable MAIN.nEL2008_value (Output)

```
I/O - Configuration
  I/O Devices
    Device 1 (EtherCAT)
      Fm0Ctrl  >  QB 1520.0, UINT16 [2.0]
      Fm0WcCtrl  >  QB 1522.0, UINT16 [2.0]
      DevCtrl  >  QB 1534.0, UINT16 [2.0]
    Device 3 (EtherCAT)
      Term 5 (EL2008)
        Output  >  QX 26.0, BIT [0.1]
        Output  >  QX 26.1, BIT [0.1]
        Output  >  QX 26.2, BIT [0.1]
        Output  >  QX 26.3, BIT [0.1]
        Output  >  QX 26.4, BIT [0.1]
        Output  >  QX 26.5, BIT [0.1]
        Output  >  QX 26.6, BIT [0.1]
        Output  >  QX 26.7, BIT [0.1]
      Fm0Ctrl  >  QB 1520.0, UINT16 [2.0]
      Fm0WcCtrl  >  QB 1522.0, UINT16 [2.0]
      DevCtrl  >  QB 1534.0, UINT16 [2.0]
```

Output . Channel 5 . Term 5 (EL2008) . Device 3 (EtherCAT) . I/O Devices

Show Variables: ● Unused, ○ Used and unused, □ Exclude disabled, ☑ Exclude other Devices, ☑ Exclude same Image, ■ Show Tooltips

Show Variable Types: □ Matching Type, ☑ Matching Size, ☑ All Types, □ Array Mode

☑ Continuous, □ Show Dialog

Variable Name: □ Hand over, □ Take over

Fig. 44: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte

corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol (⧉) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

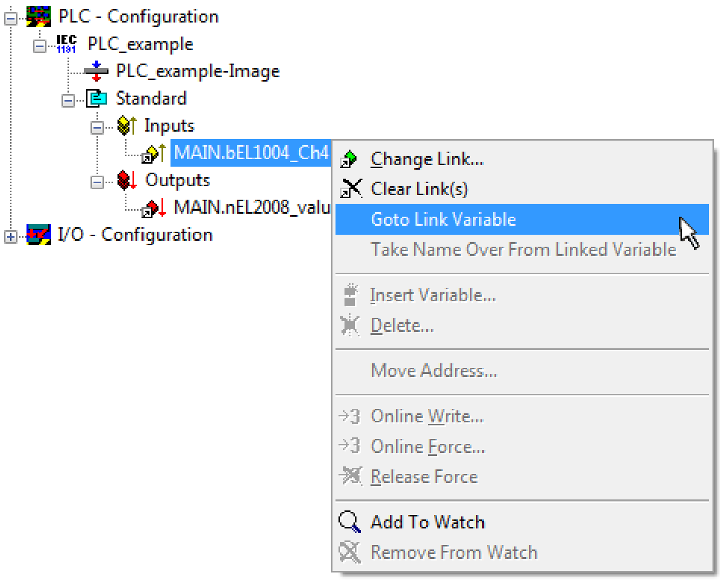
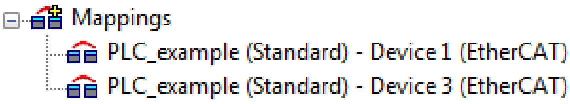


Fig. 45: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of assigning variables to the PDO is completed via the menu option “Actions” → “Create assignment”, or via .

This can be visualized in the configuration:


Mappings
PLC_example (Standard) - Device 1 (EtherCAT)
PLC_example (Standard) - Device 3 (EtherCAT)

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

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated. First, the configuration can be verified via ✓ (or via “Actions” → “Check Configuration”). If no error is present, the configuration can be activated via (or via “Actions” → “Activate Configuration…”) to transfer the System Manager settings to the runtime system. Confirm the messages “Old configurations will be overwritten!” and “Restart TwinCAT system in Run mode” with “OK”.

A few seconds later, the real-time status 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 the Ethernet via “Online” → “Choose Runtime System…”:

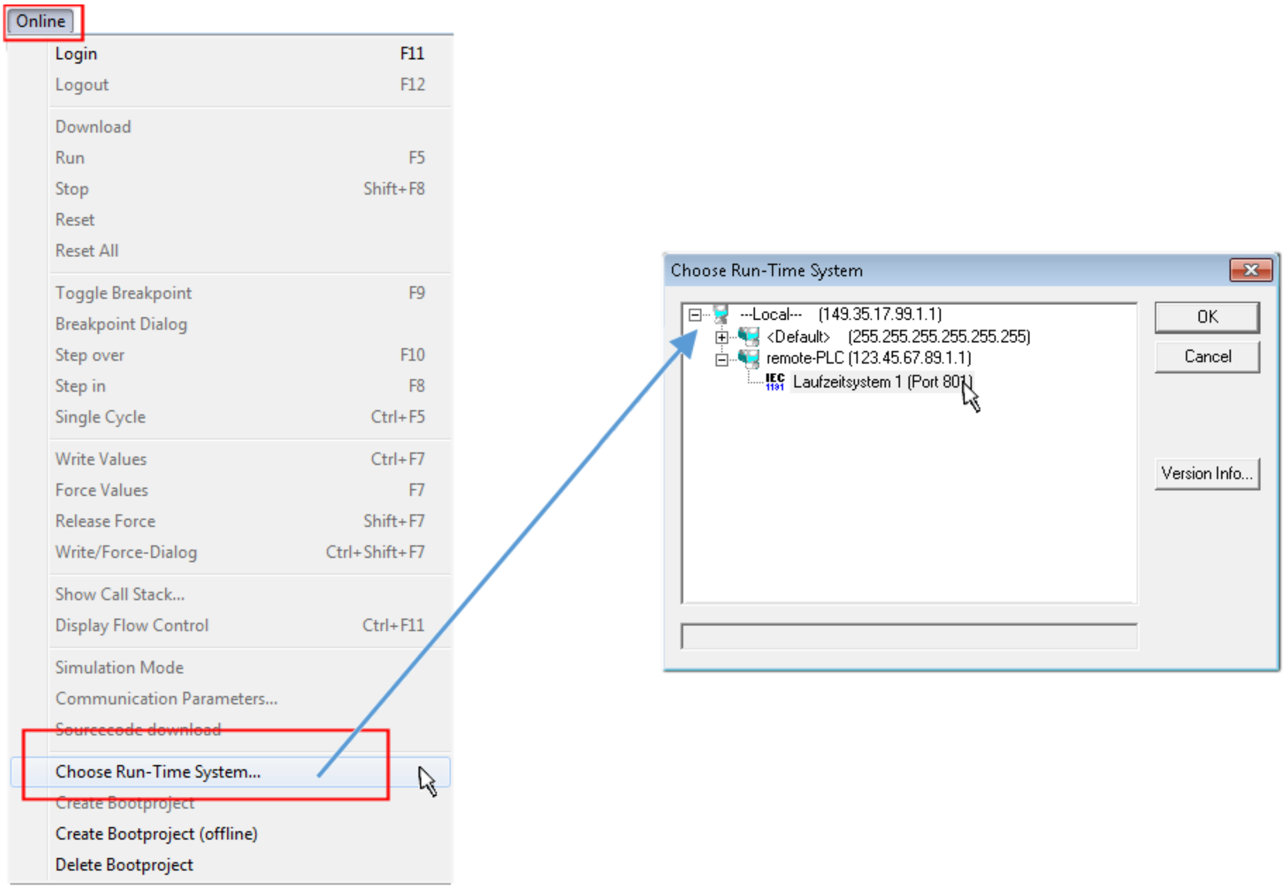
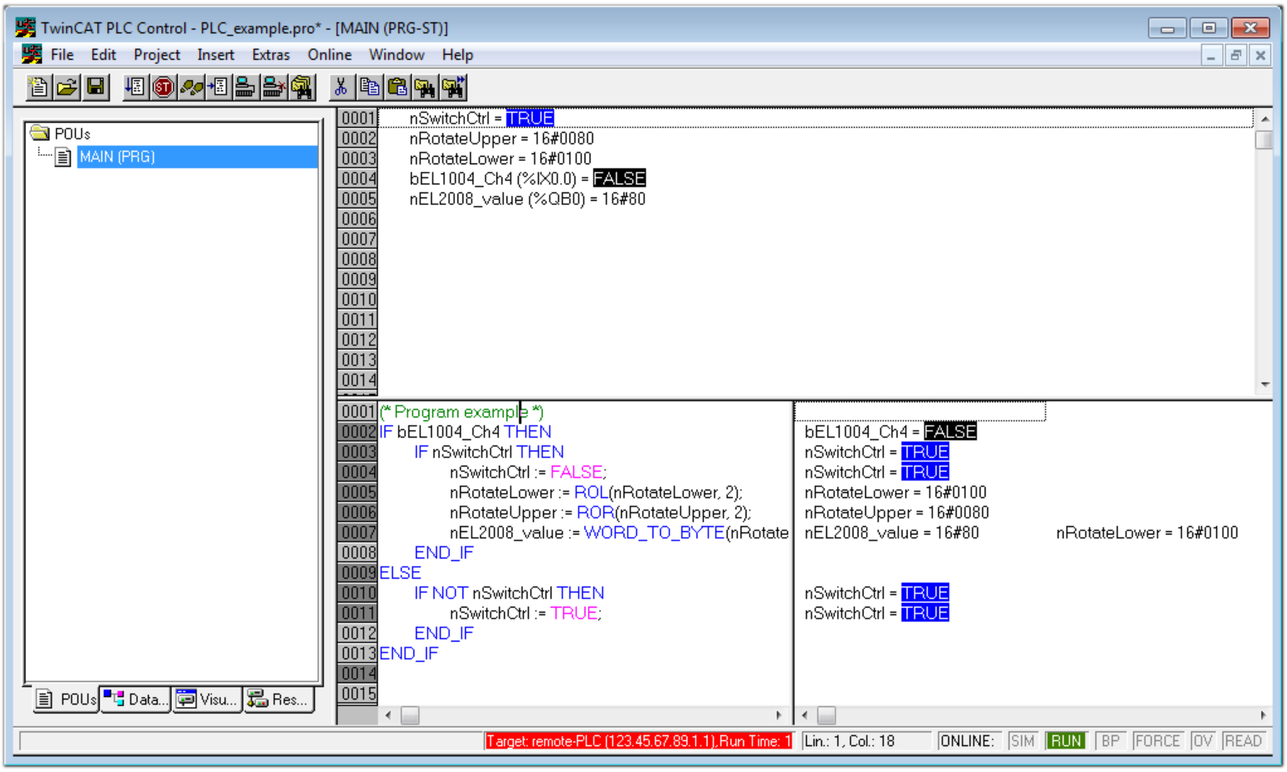


Fig. 46: Choose target system (remote)

In this example, “Runtime system 1 (port 801)” is selected and confirmed. Link the PLC with the real-time system via the menu option “Online” → “Login”, the F11 key or by clicking on the symbol . The control program can then be loaded for execution. This results in the message “No program on the controller! Should the new program be loaded?”, which should be confirmed with “Yes”. The runtime environment is ready for the program start:



TwinCAT PLC Control - PLC_example.pro* - [MAIN (PRG-ST)]
File Edit Project Insert Extras Online Window Help
POUs
MAIN (PRG)
POUs
Data...
Visu...
Res...

```
nSwitchCtrl = TRUE
nRotateUpper = 16#0080
nRotateLower = 16#0100
bEL1004_Ch4 (%IX0.0) = FALSE
nEL2008_value (%QB0) = 16#80
```

```
(* Program example *)
IF bEL1004_Ch4 THEN
    IF nSwitchCtrl THEN
        nSwitchCtrl := FALSE;
        nRotateLower := ROL(nRotateLower, 2);
        nRotateUpper := ROR(nRotateUpper, 2);
        nEL2008_value := WORD_TO_BYTE(nRotate
    END_IF
ELSE
    IF NOT nSwitchCtrl THEN
        nSwitchCtrl := TRUE;
    END_IF
END_IF
```

```
bEL1004_Ch4 = FALSE
nSwitchCtrl = TRUE
nSwitchCtrl = TRUE
nRotateLower = 16#0100
nRotateUpper = 16#0080
nEL2008_value = 16#80          nRotateLower = 16#0100


nSwitchCtrl = TRUE
nSwitchCtrl = TRUE
```

```
Target: remote-PLC (123.45.67.89.1.1),Run Time: 1   Lin.: 1, Col.: 18   ONLINE:  SIM  RUN  BP  FORCE  OV  READ
```

Fig. 47: 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 3 makes the development environment areas available all together, with Microsoft Visual Studio: after startup, the project folder explorer appears on the left in the general window area (see “TwinCAT System Manager” of TwinCAT 2) for communication with the electromechanical components.

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

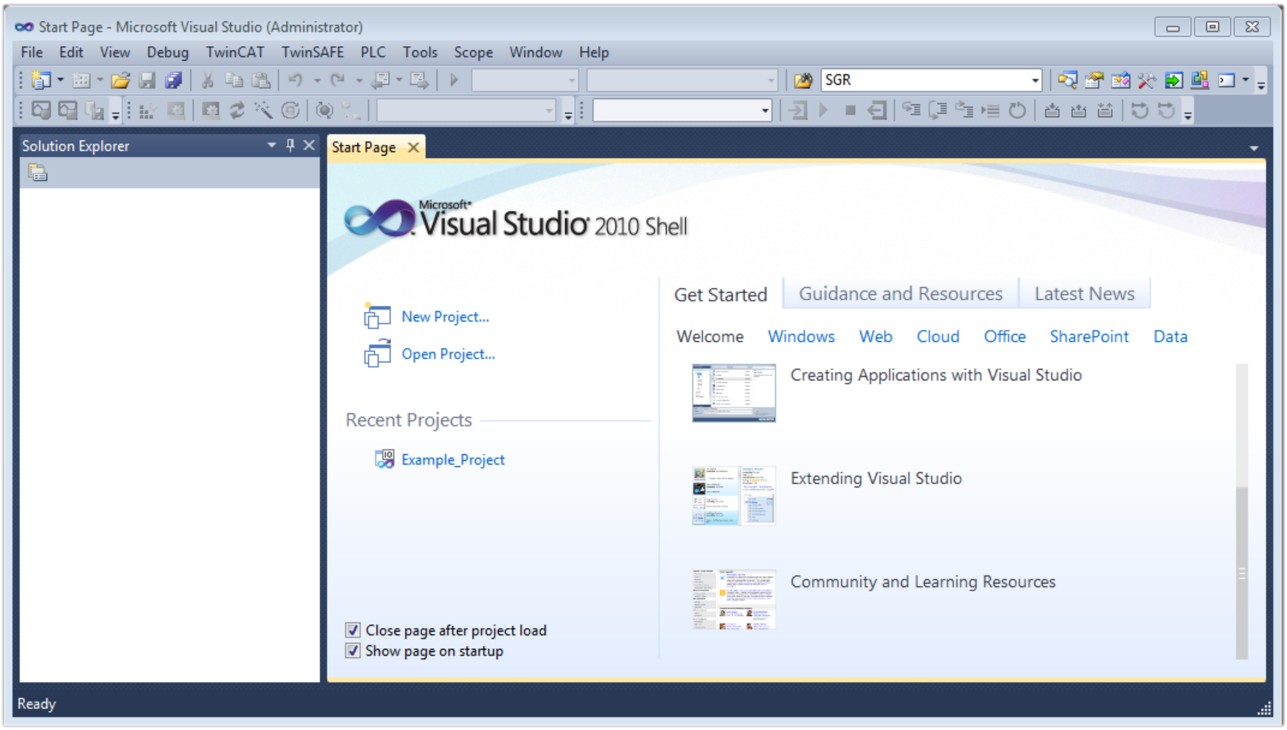
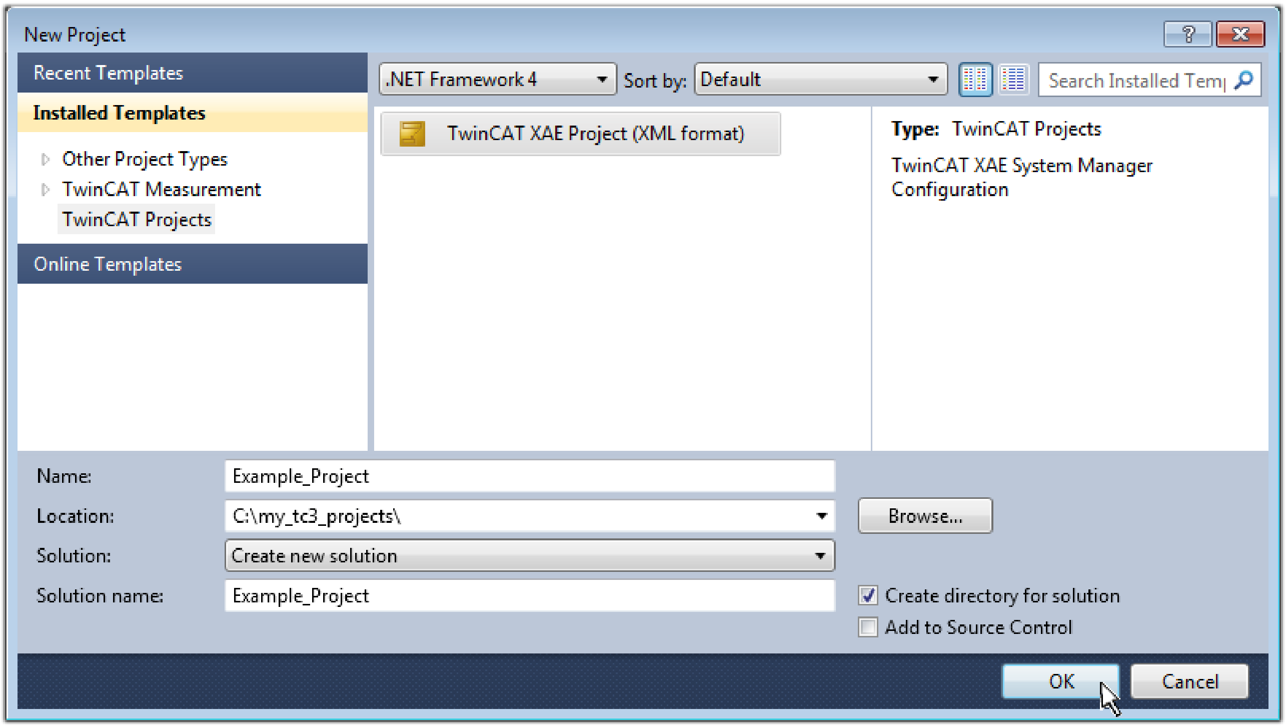

Start Page - Microsoft Visual Studio (Administrator)
File Edit View Debug TwinCAT TwinSAFE PLC Tools Scope Window Help
SGR
Solution Explorer
Start Page
Microsoft Visual Studio 2010 Shell
New Project...
Open Project...
Recent Projects
Example_Project
Get Started
Guidance and Resources
Latest News
Welcome Windows Web Cloud Office SharePoint Data
Creating Applications with Visual Studio
Extending Visual Studio
Community and Learning Resources
☑ Close page after project load
☑ Show page on startup
Ready

Fig. 48: 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):



.NET Framework 4

Sort by: Default

TwinCAT XAE Project (XML format)

Type: TwinCAT Projects

TwinCAT XAE System Manager Configuration

Name: Example_Project

Location: C:\my_tc3_projects\

Solution: Create new solution

Solution name: Example_Project

☑ Create directory for solution

☐ Add to Source Control

Fig. 49: Create new TwinCAT 3 project

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

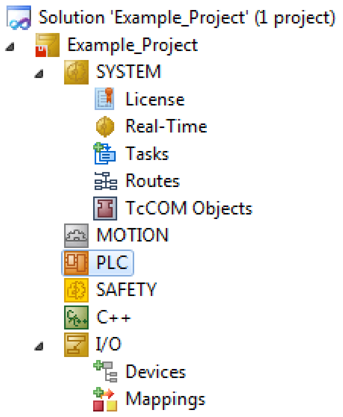
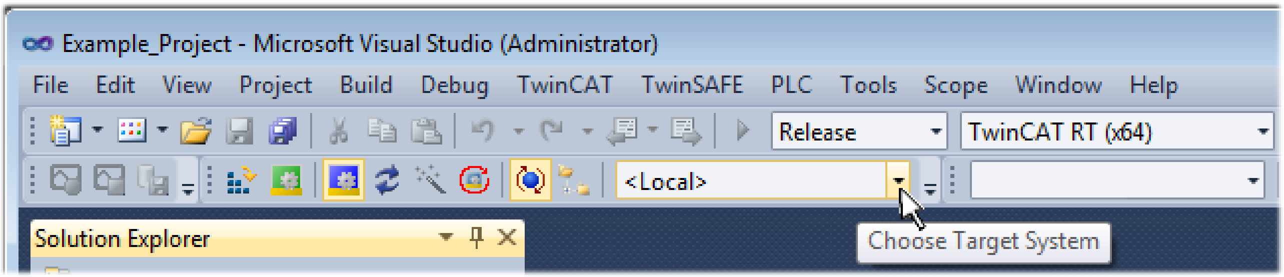

Solution 'Example_Project' (1 project)
Example_Project
SYSTEM
License
Real-Time
Tasks
Routes
TcCOM Objects
MOTION
PLC
SAFETY
C++
I/O
Devices
Mappings

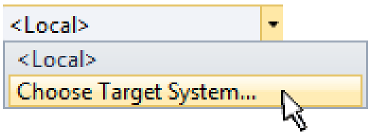
Fig. 50: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, "Insert Device [▸ 70]".

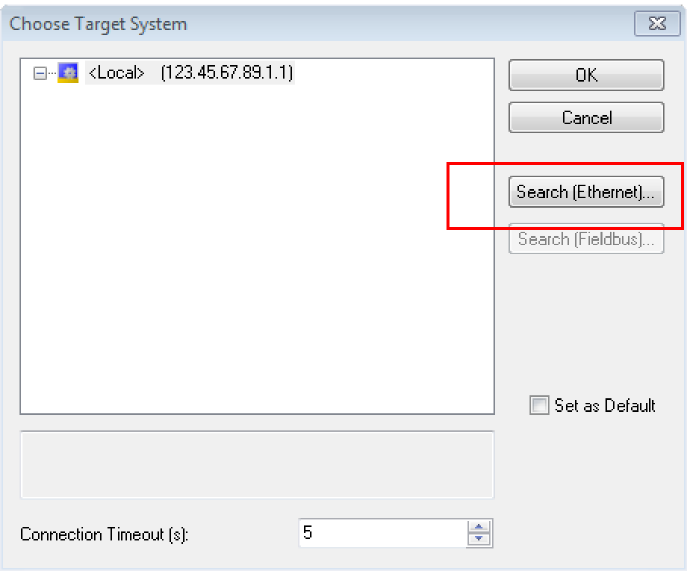
If the intention is to address the TwinCAT runtime environment installed on a PLC remotely from another system used as a development environment, the target system must be made known first. Via the symbol in the menu bar:


Example_Project - Microsoft Visual Studio (Administrator)

expand the pull-down menu:



and open the following window:


Choose Target System

<Local> (123.45.67.89.1.1)

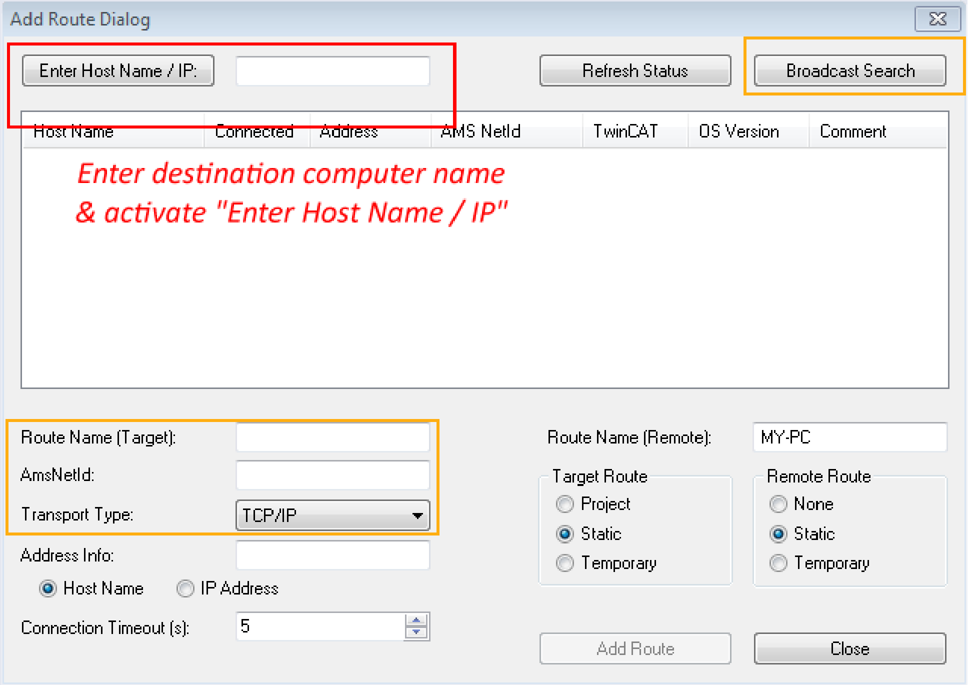
☐ Set as Default

Connection Timeout (s): 5

Fig. 51: Selection dialog: Choose the target system

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

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



Transport Type: TCP/IP

Address Info: ● Host Name ○ IP Address

Connection Timeout (s): 5

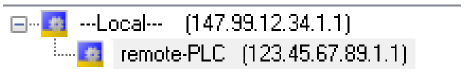
Route Name (Remote): MY-PC

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

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

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



---Local--- (147.99.12.34.1.1)
remote-PLC (123.45.67.89.1.1)

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

Adding devices

In the project folder explorer on the left of the Visual Studio shell user interface, select “Devices” within the element “I/O”, then right-click to open a context menu and select “Scan” or start the action via in the menu bar. The TwinCAT System Manager may first have to be set to “Config mode” via or via the menu “TwinCAT” → “Restart TwinCAT (Config Mode)”.

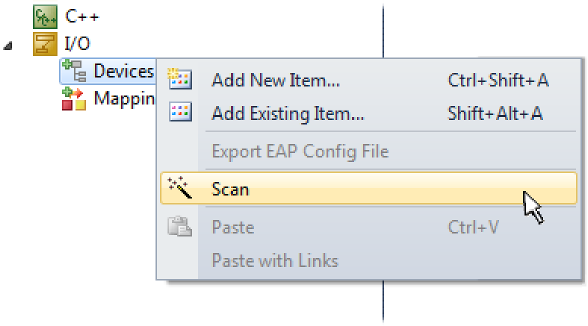
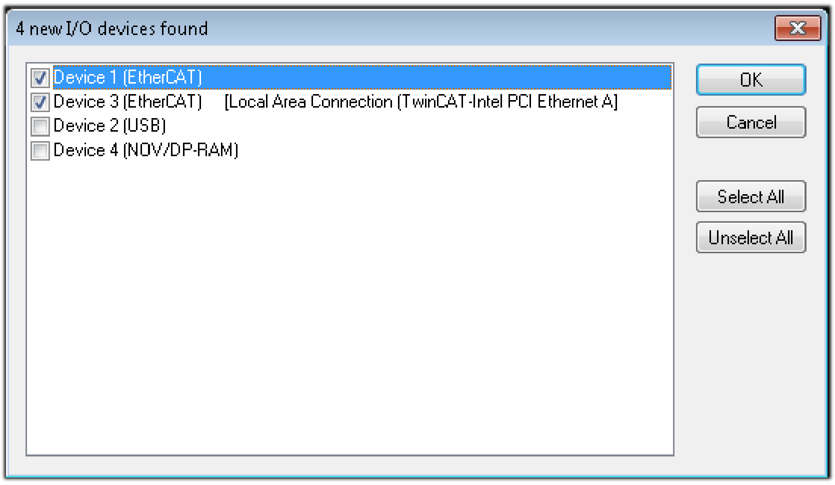


Fig. 53: Select “Scan”

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



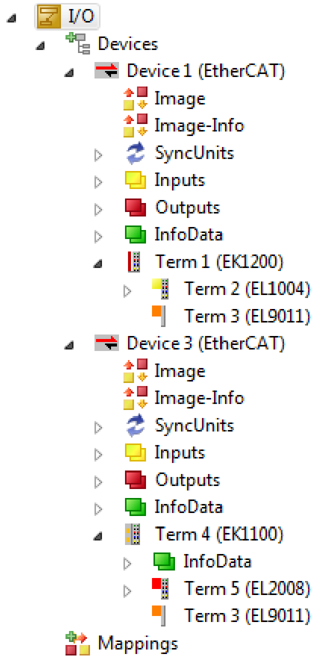
4 new I/O devices found

- ☑ Device 1 (EtherCAT)
- ☑ Device 3 (EtherCAT) [Local Area Connection (TwinCAT-Intel PCI Ethernet A)
- □ Device 2 (USB)
- □ Device 4 (NOV/DP-RAM)

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

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

Based on the example configuration [▸ 55] described at the beginning of this section, the result is as follows:



```
I/O
  Devices
    Device 1 (EtherCAT)
      Image
      Image-Info
      SyncUnits
      Inputs
      Outputs
      InfoData
      Term 1 (EK1200)
        Term 2 (EL1004)
        Term 3 (EL9011)
    Device 3 (EtherCAT)
      Image
      Image-Info
      SyncUnits
      Inputs
      Outputs
      InfoData
      Term 4 (EK1100)
        InfoData
        Term 5 (EL2008)
        Term 3 (EL9011)
  Mappings
```

Fig. 55: Mapping of the configuration in VS shell of the TwinCAT 3 environment

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

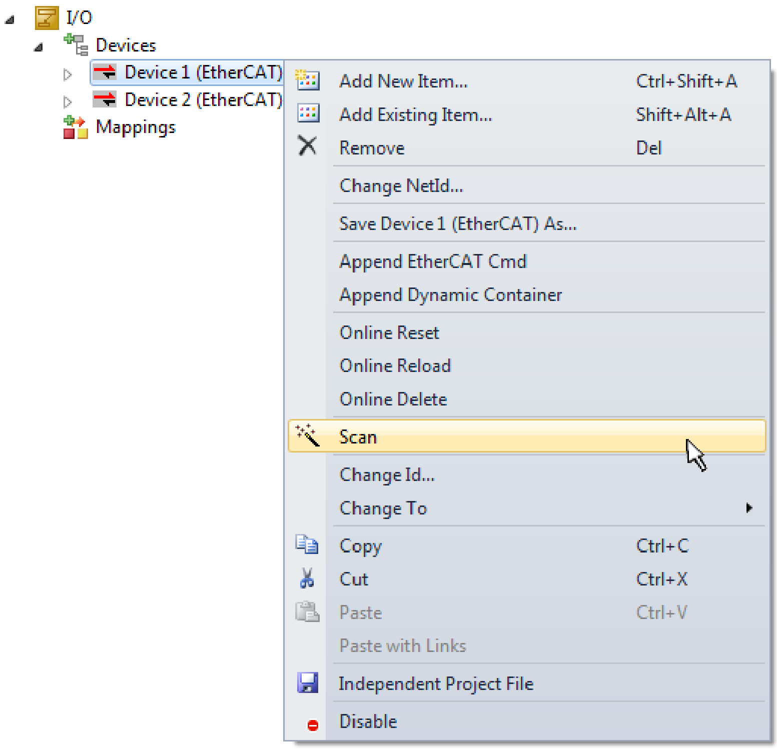


Fig. 56: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

The following section refers solely to Structured Text (ST).

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

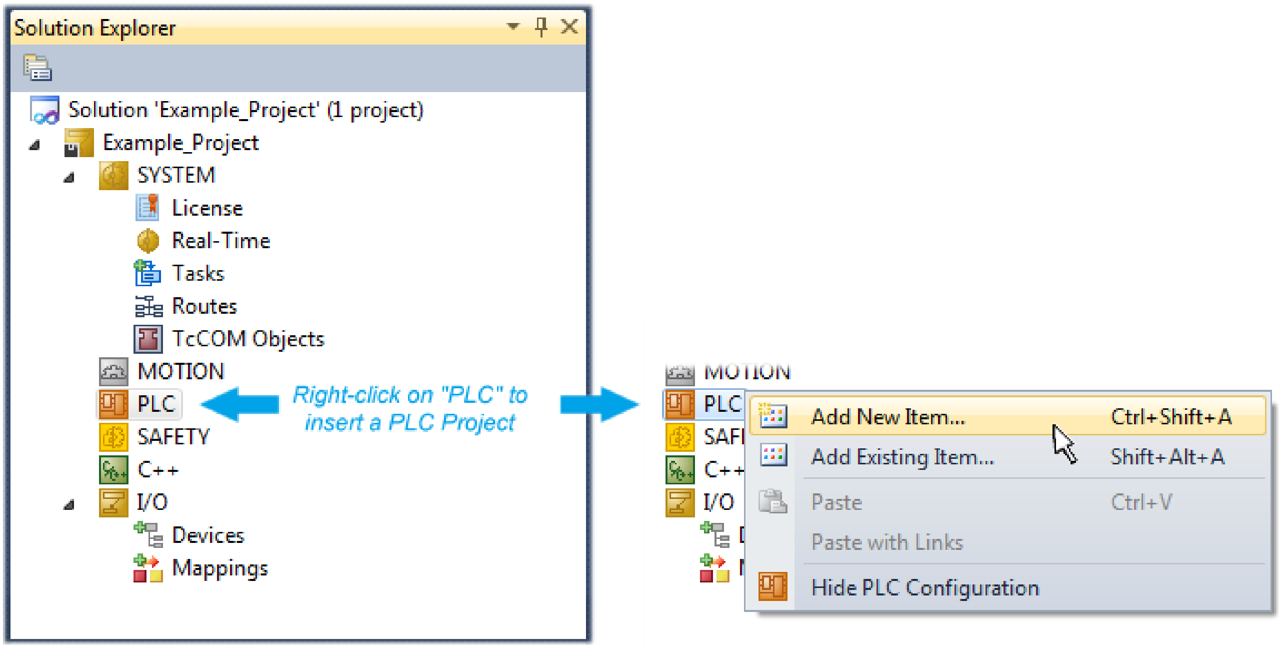
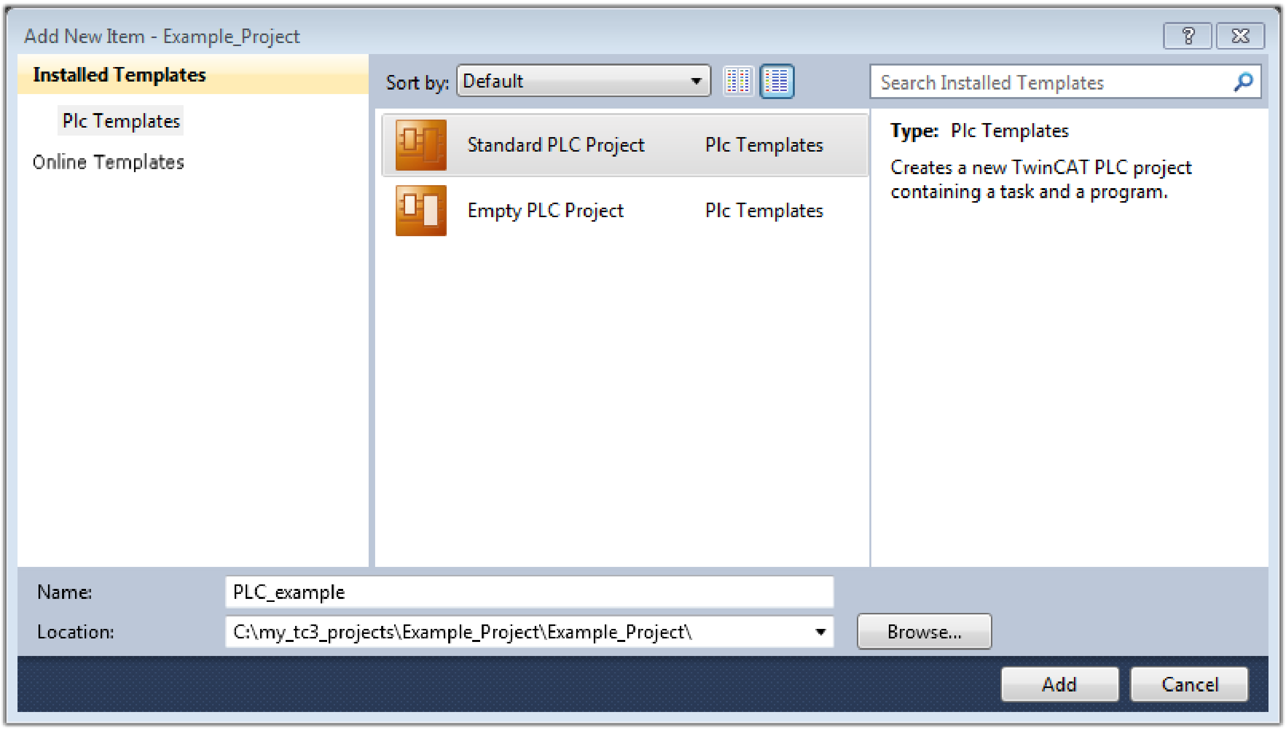


Fig. 57: Adding the programming environment in “PLC”

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


Add New Item - Example_Project

Installed Templates

Plc Templates

Online Templates

Sort by: Default

Search Installed Templates

Standard PLC Project	Plc Templates
Empty PLC Project	Plc Templates

Type: Plc Templates

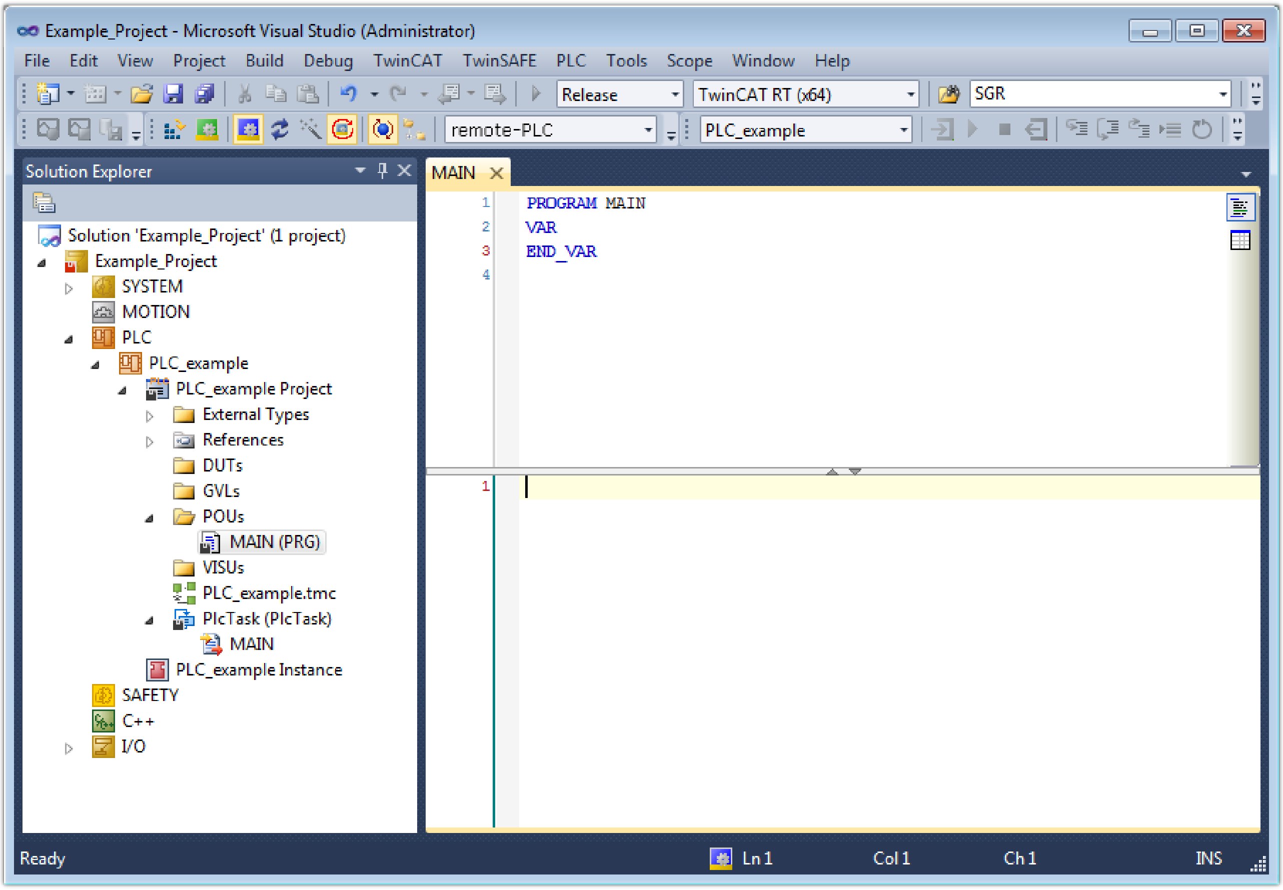
Creates a new TwinCAT PLC project containing a task and a program.

Name: PLC_example

Location: C:\my_tc3_projects\Example_Project\Example_Project\

Fig. 58: Specifying the name and directory for the PLC programming environment

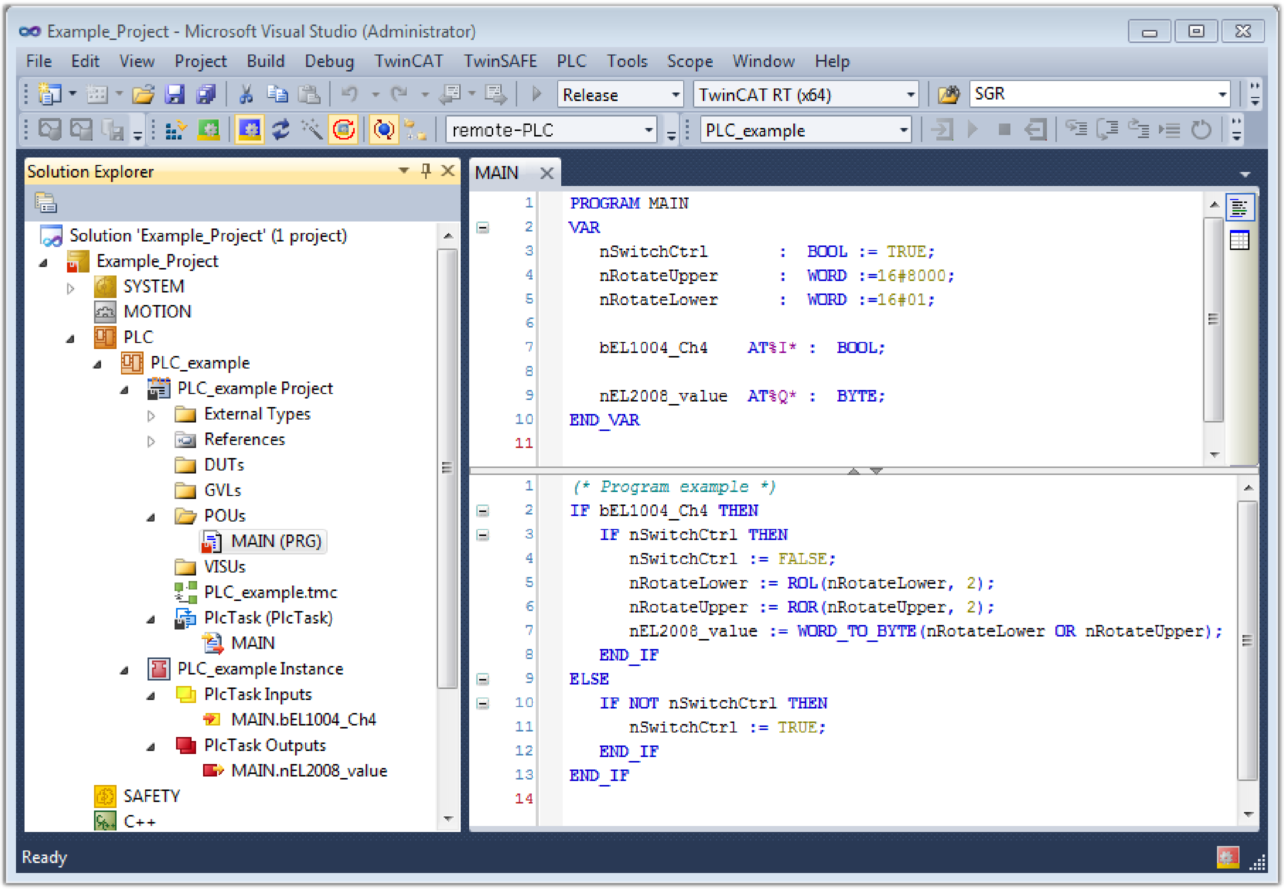
The “Main” program, which already exists due to selecting “Standard PLC project”, can be opened by double-clicking on “PLC_example_project” in “POUs”. The following user interface is shown for an initial project:



```
PROGRAM MAIN
VAR
END_VAR
```

Fig. 59: Initial “Main” program for the standard PLC project

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



```
PROGRAM MAIN
VAR
   nSwitchCtrl       :  BOOL := TRUE;
   nRotateUpper      :  WORD :=16#8000;
   nRotateLower      :  WORD :=16#01;

   bEL1004_Ch4    AT%I* :  BOOL;

   nEL2008_value  AT%Q* :  BYTE;
END_VAR
```

```
(* Program example *)
IF bEL1004_Ch4 THEN
   IF nSwitchCtrl THEN
      nSwitchCtrl  := FALSE;
      nRotateLower := ROL(nRotateLower, 2);
      nRotateUpper := ROR(nRotateUpper, 2);
      nEL2008_value := WORD_TO_BYTE(nRotateLower OR nRotateUpper);
   END_IF
ELSE
   IF NOT nSwitchCtrl THEN
      nSwitchCtrl  := TRUE;
   END_IF
END_IF
```

Fig. 60: Example program with variables after a compile process (without variable integration)

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

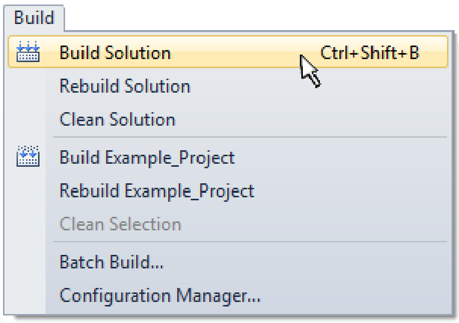
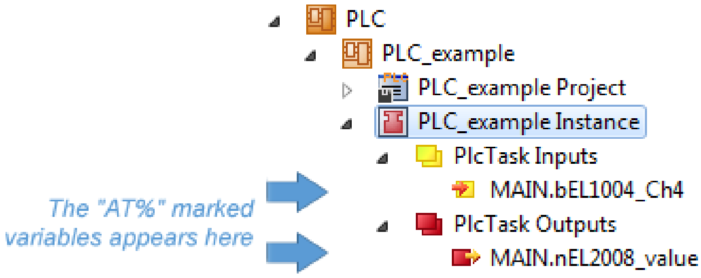


Fig. 61: Start program compilation

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



Assigning variables

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

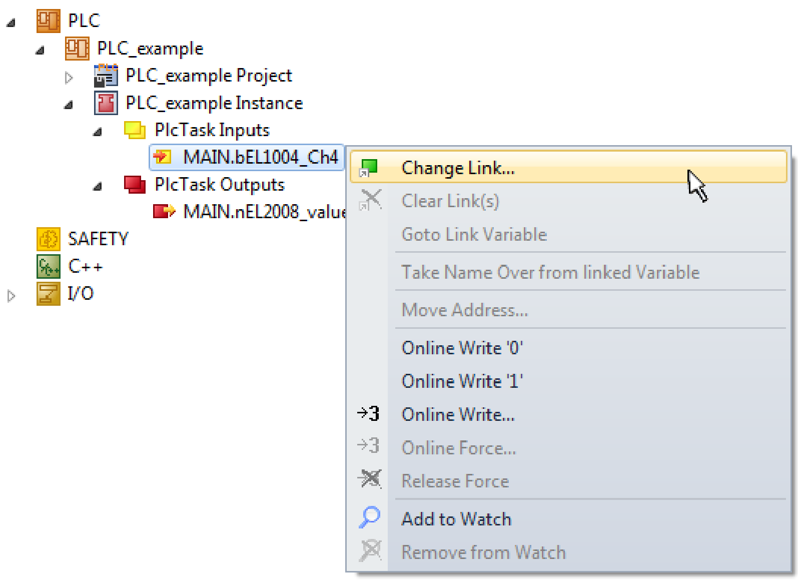


Fig. 62: Creating the links between PLC variables and process objects

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

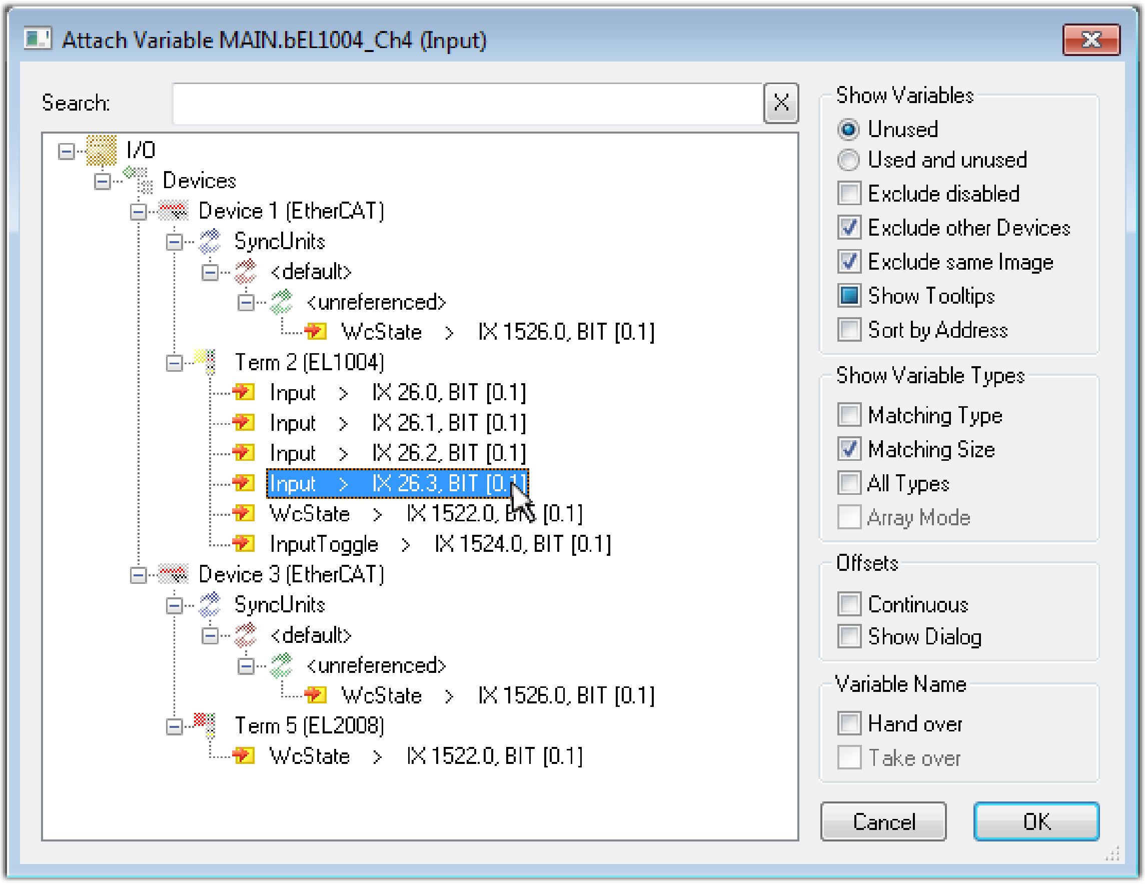
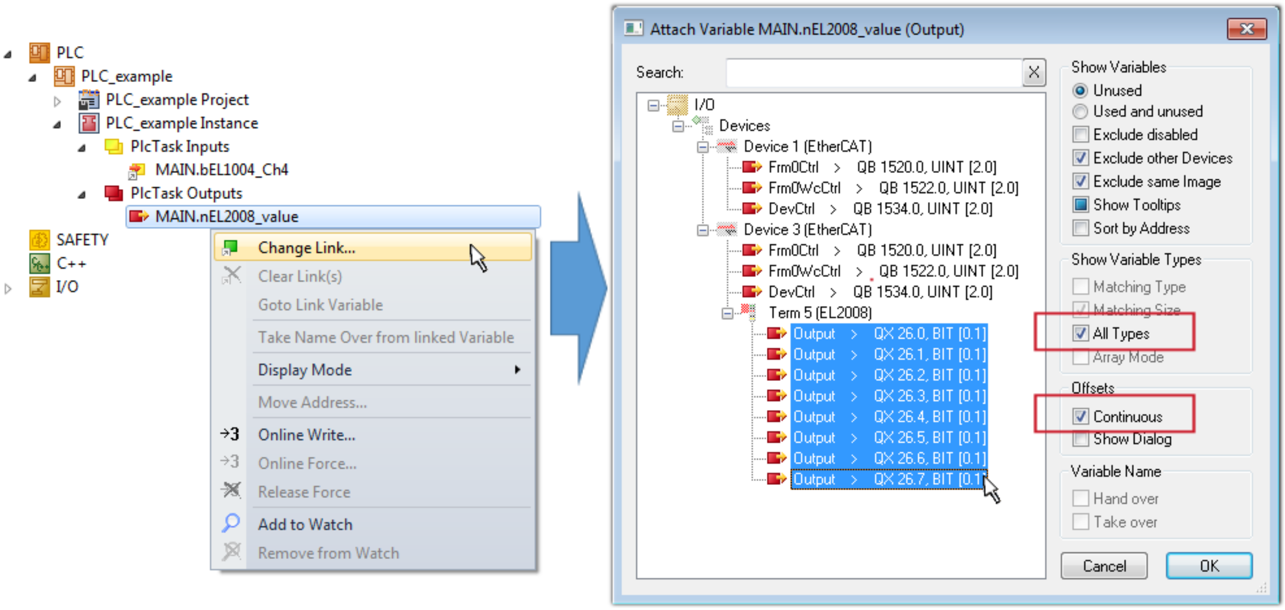


Fig. 63: Selecting BOOL-type PDO

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


PLC
PLC_example
PLC_example Project
PLC_example Instance
PlcTask Inputs
MAIN.bEL1004_Ch4
PlcTask Outputs
MAIN.nEL2008_value
SAFETY
C++
I/O
Change Link...
Clear Link(s)
Goto Link Variable
Take Name Over from linked Variable
Display Mode
Move Address...
Online Write...
Online Force...
Release Force
Add to Watch
Remove from Watch
Attach Variable MAIN.nEL2008_value (Output)
Search:
I/O
Devices
Device 1 (EtherCAT)
Frm0Ctrl > QB 1520.0, UINT [2.0]
Frm0WcCtrl > QB 1522.0, UINT [2.0]
DevCtrl > QB 1534.0, UINT [2.0]
Device 3 (EtherCAT)
Frm0Ctrl > QB 1520.0, UINT [2.0]
Frm0WcCtrl > QB 1522.0, UINT [2.0]
DevCtrl > QB 1534.0, UINT [2.0]
Term 5 (EL2008)
Output > QX 26.0, BIT [0.1]
Output > QX 26.1, BIT [0.1]
Output > QX 26.2, BIT [0.1]
Output > QX 26.3, BIT [0.1]
Output > QX 26.4, BIT [0.1]
Output > QX 26.5, BIT [0.1]
Output > QX 26.6, BIT [0.1]
Output > QX 26.7, BIT [0.1]
Cancel
OK

Show Variables
● Unused
○ Used and unused
□ Exclude disabled
☑ Exclude other Devices
☑ Exclude same Image
■ Show Tooltips
□ Sort by Address

Show Variable Types
□ Matching Type
☑ Matching Size
☑ All Types
□ Array Mode

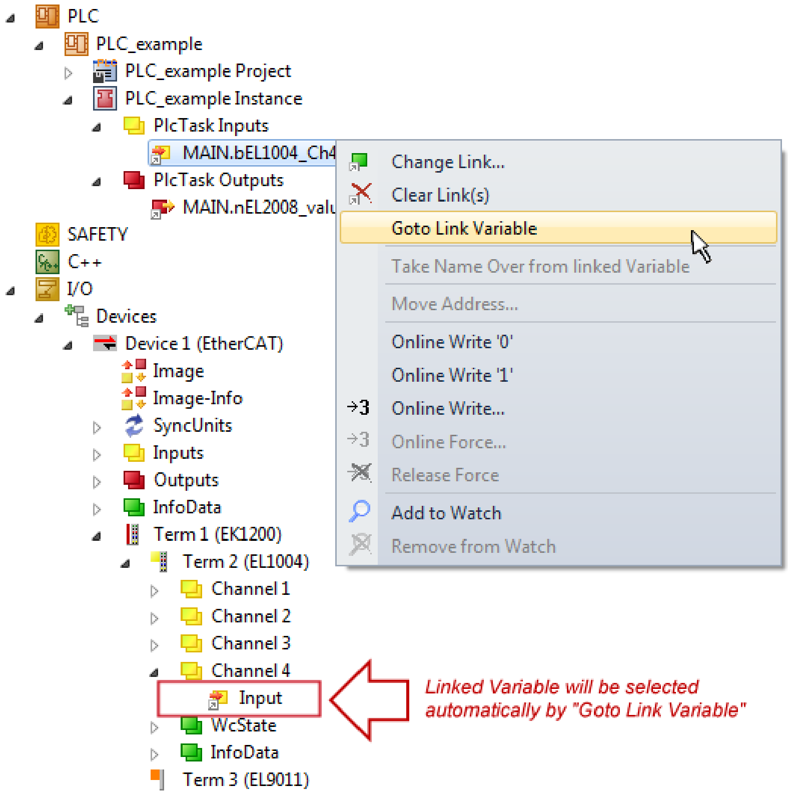
Offsets
☑ Continuous
□ Show Dialog

Variable Name
□ Hand over
□ Take over

Fig. 64: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte

corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol (⧉) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:



Linked Variable will be selected automatically by "Goto Link Variable"

Fig. 65: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

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

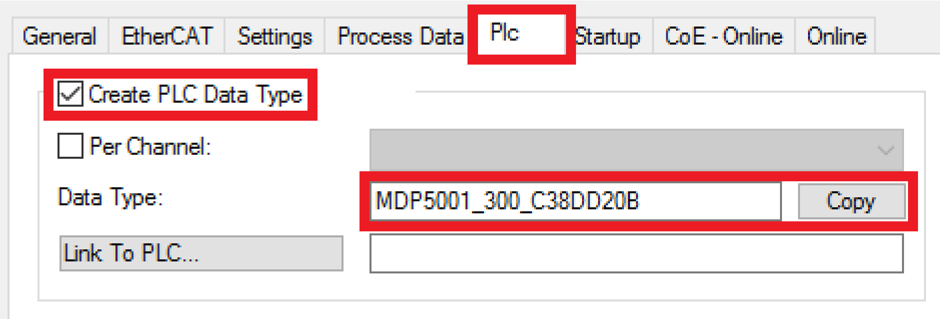
Note on type of variable assignment

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

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

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

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



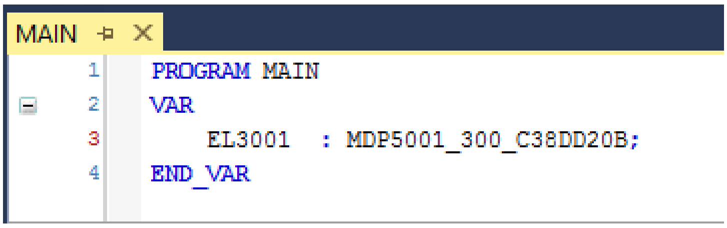
☑ Create PLC Data Type

□ Per Channel:

Data Type: MDP5001_300_C38DD20B

Fig. 66: Creating a PLC data type

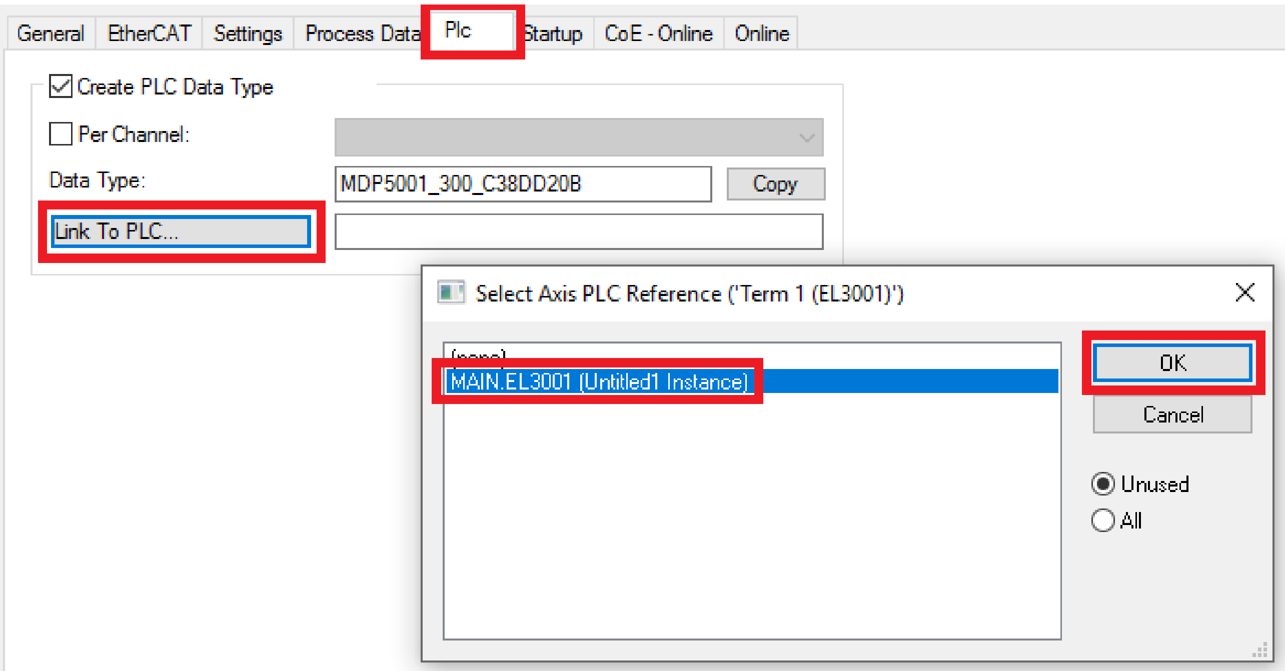
4. An instance of the data structure of the copied data type must then be created in the PLC.



```
PROGRAM MAIN
VAR
    EL3001  : MDP5001_300_C38DD20B;
END_VAR
```

Fig. 67: Instance_of_struct

5. Then the project folder must be created. This can be done either via the key combination “CTRL + Shift + B” or via the “Build” tab in TwinCAT.
6. The structure in the “PLC” tab of the terminal must then be linked to the created instance.



☑ Create PLC Data Type

□ Per Channel:

Data Type: MDP5001_300_C38DD20B

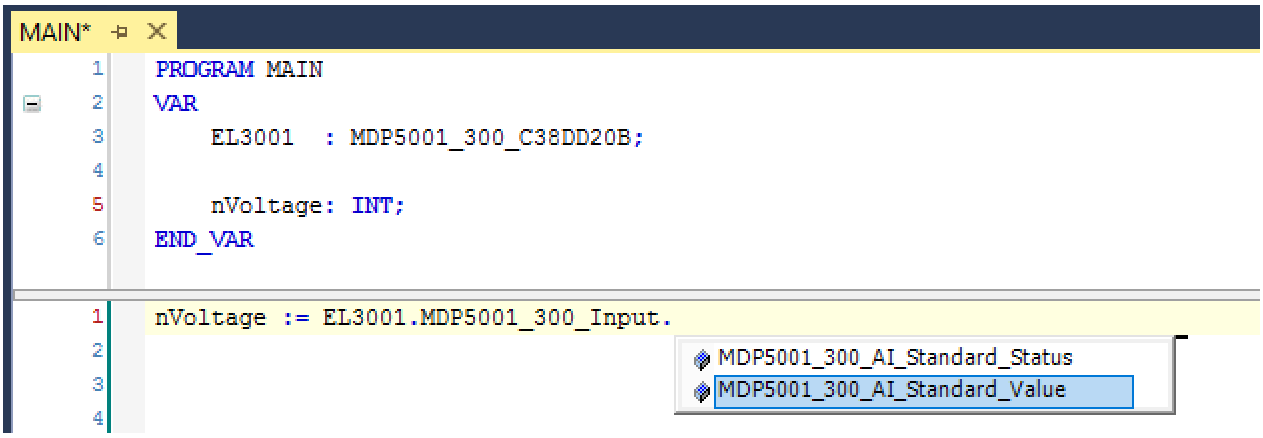
Select Axis PLC Reference ('Term 1 (EL3001)')

(none)
MAIN.EL3001 (Untitled1 Instance)

● Unused
○ All

Fig. 68: Linking the structure

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



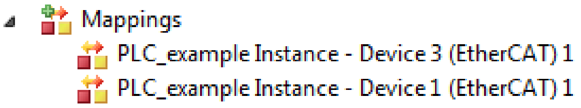
```
PROGRAM MAIN
VAR
    EL3001  : MDP5001_300_C38DD20B;

    nVoltage: INT;
END_VAR
```

```
nVoltage := EL3001.MDP5001_300_Input.
```

Fig. 69: Reading a variable from the structure of the process data

Activation of the configuration

The allocation of PDO to PLC variables has now established the connection from the controller to the inputs and outputs of the terminals. The configuration can now be activated with or via the menu under “TwinCAT” in order to transfer the settings of the development environment to the runtime system. Confirm the messages “Old configurations will be overwritten!” and “Restart TwinCAT system in Run mode” with “OK”. The corresponding assignments can be seen in the project folder explorer:

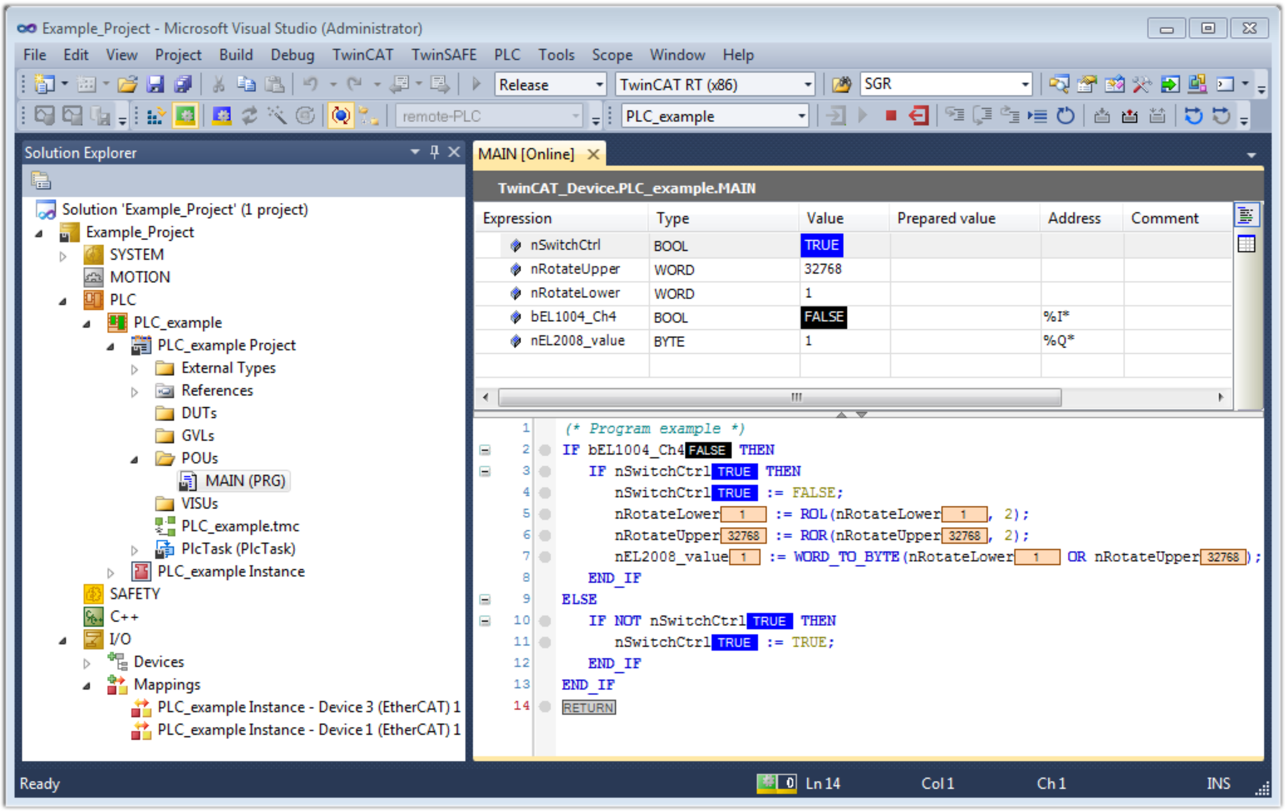


Mappings
PLC_example Instance - Device 3 (EtherCAT) 1
PLC_example Instance - Device 1 (EtherCAT) 1

A few seconds later, the corresponding status of the Run mode is displayed in the form of a rotating symbol at the bottom right of the VS shell development environment. The PLC system can then be started as described below.

Starting the controller

Select the menu option “PLC” → “Login” or click on to link the PLC with the real-time system and load the control program for execution. This results in the message ”*No program on the controller! Should the new program be loaded?*”, which should be acknowledged with “Yes”. The runtime environment is ready for the program to be started by clicking on symbol ▶ , the “F5” key or via “PLC” in the menu, by selecting “Start”. The started programming environment shows the runtime values of individual variables:



TwinCAT_Device.PLC_example.MAIN

Expression	Type	Value	Prepared value	Address	Comment
nSwitchCtrl	BOOL	TRUE			
nRotateUpper	WORD	32768			
nRotateLower	WORD	1			
bEL1004_Ch4	BOOL	FALSE		%I*	
nEL2008_value	BYTE	1		%Q*	

```
(* Program example *)
IF bEL1004_Ch4 FALSE THEN
   IF nSwitchCtrl TRUE THEN
      nSwitchCtrl TRUE := FALSE;
      nRotateLower 1 := ROL(nRotateLower 1, 2);
      nRotateUpper 32768 := ROR(nRotateUpper 32768, 2);
      nEL2008_value 1 := WORD_TO_BYTE(nRotateLower 1 OR nRotateUpper 32768);
   END_IF
ELSE
   IF NOT nSwitchCtrl TRUE THEN
      nSwitchCtrl TRUE := TRUE;
   END_IF
END_IF
RETURN
```

Fig. 70: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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

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- and 64 bit operating system
  - Automatic code generation and project creation with the TwinCAT Automation Interface
  - More...

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

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

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.

A: Via the TwinCAT Adapter dialog

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

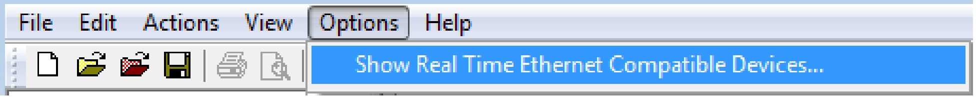


Fig. 71: System Manager “Options” (TwinCAT 2)

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

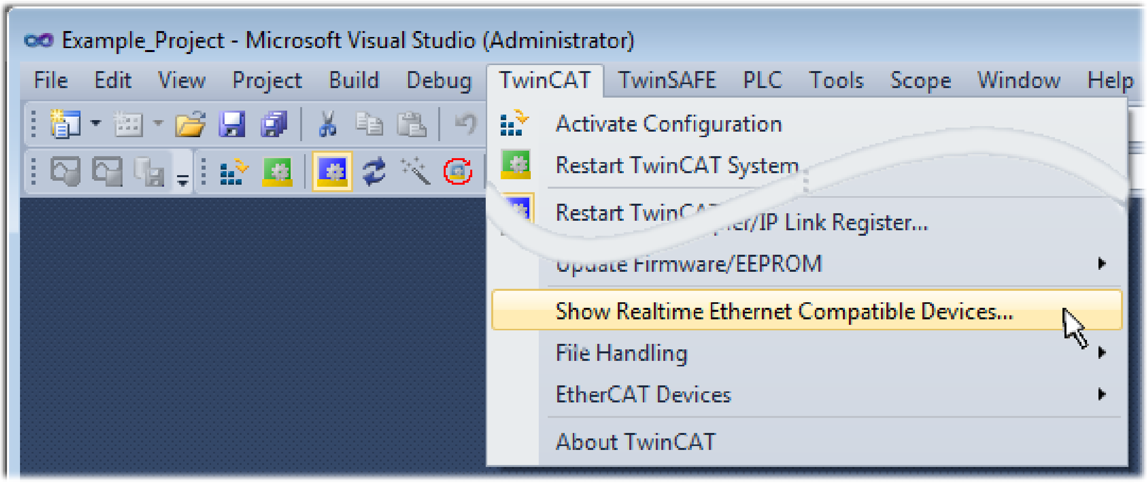
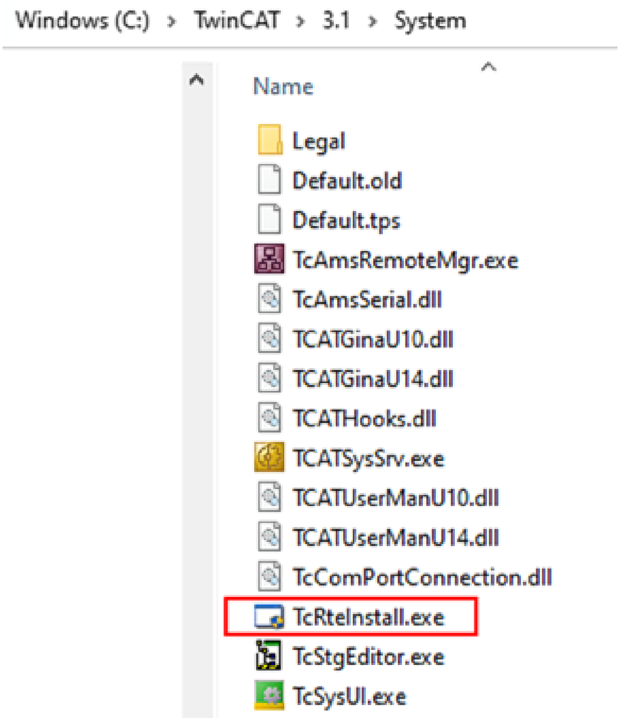


Fig. 72: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory


Windows (C:) › TwinCAT › 3.1 › System

Name

Legal
Default.old
Default.tps
TcAmsRemoteMgr.exe
TcAmsSerial.dll
TCATGinaU10.dll
TCATGinaU14.dll
TCATHooks.dll
TCATSysSrv.exe
TCATUserManU10.dll
TCATUserManU14.dll
TcComPortConnection.dll
TcRteInstall.exe
TcStgEditor.exe
TcSysUI.exe

Fig. 73: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

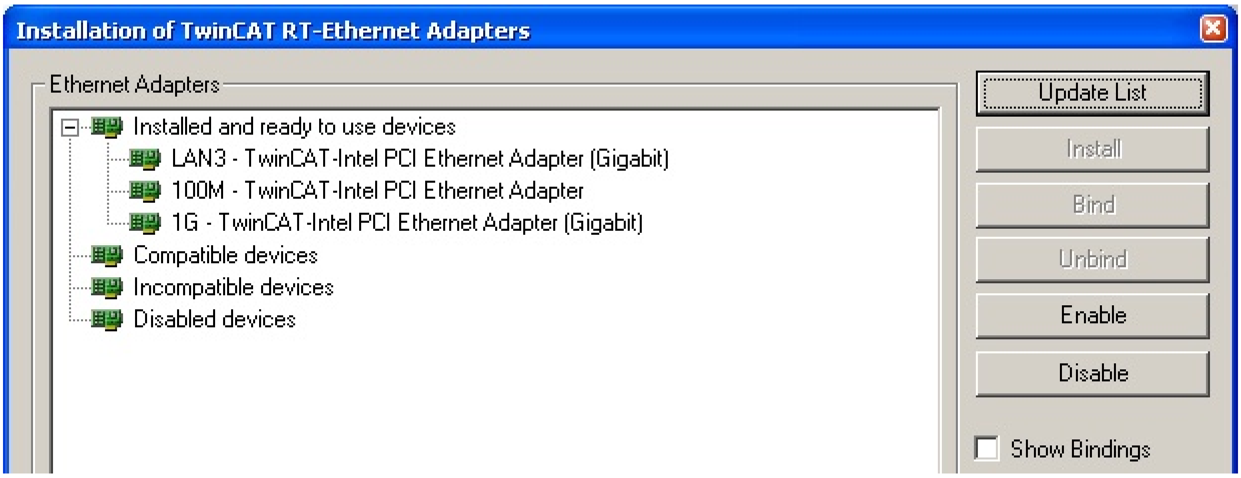
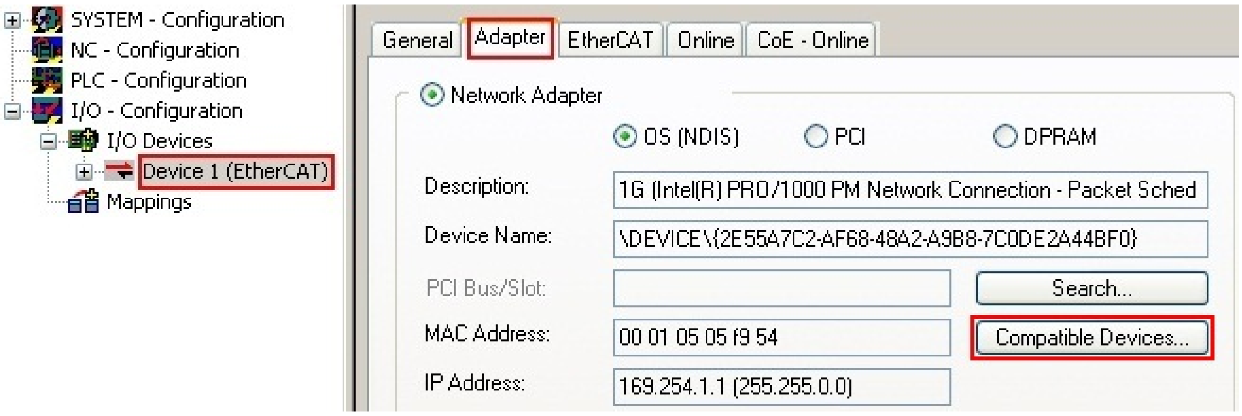

Installation of TwinCAT RT-Ethernet Adapters
Ethernet Adapters
Installed and ready to use devices
LAN3 - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
100M - TwinCAT-Intel PCI Ethernet Adapter
1G - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
Compatible devices
Incompatible devices
Disabled devices
Update List
Install
Bind
Unbind
Enable
Disable
Show Bindings

Fig. 74: 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” [▶ 92] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):



● Network Adapter

● OS (NDIS) ○ PCI ○ DPRAM

Description: 1G (Intel(R) PRO/1000 PM Network Connection - Packet Sched

Device Name: \DEVICE\{2E55A7C2-AF68-48A2-A9B8-7C0DE2A44BF0}

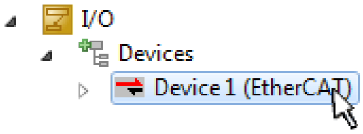
PCI Bus/Slot:

MAC Address: 00 01 05 05 f9 54

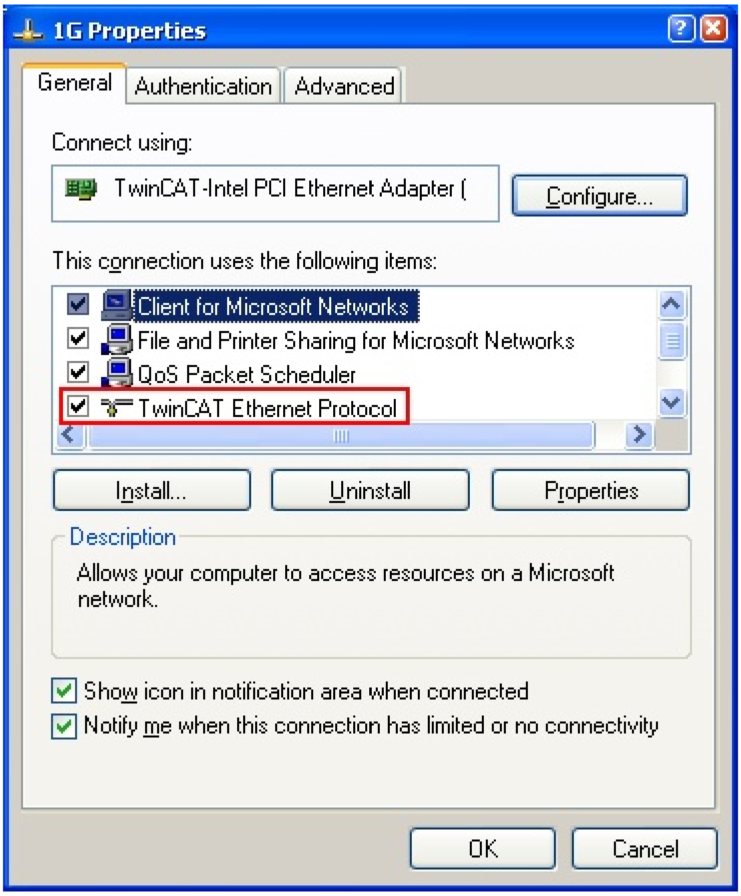
IP Address: 169.254.1.1 (255.255.0.0)

Fig. 75: 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)



Connect using:

TwinCAT-Intel PCI Ethernet Adapter (

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ TwinCAT Ethernet Protocol

Description

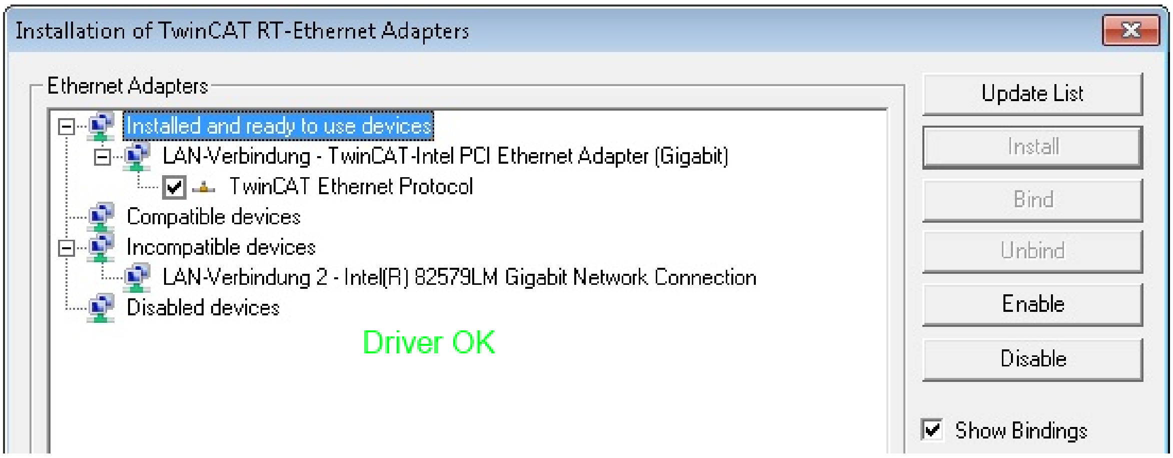
Allows your computer to access resources on a Microsoft network.

☑ Show icon in notification area when connected

☑ Notify me when this connection has limited or no connectivity

Fig. 76: Windows properties of the network interface

A correct setting of the driver could be:

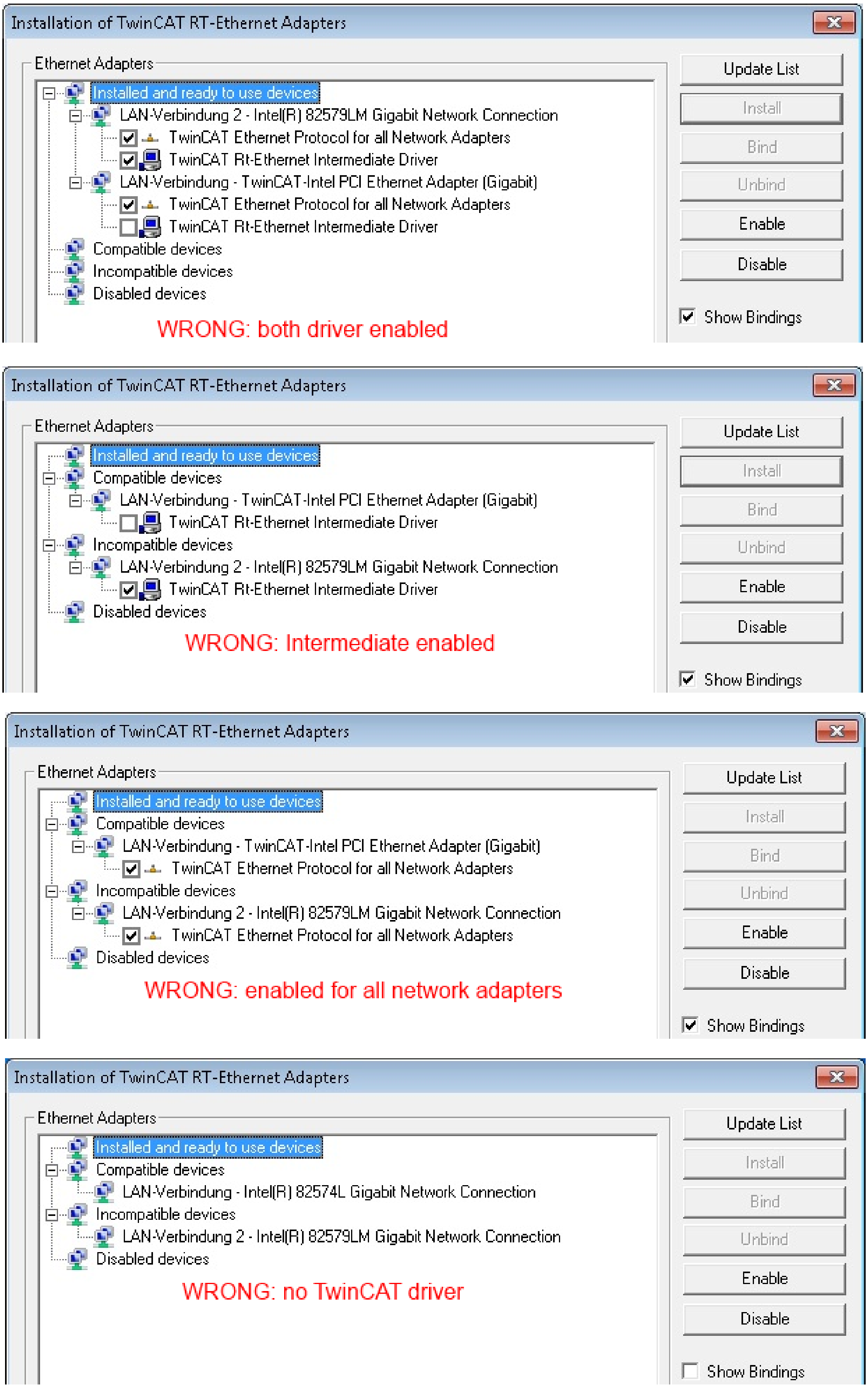


- Installed and ready to use devices
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☑ TwinCAT Ethernet Protocol
- Compatible devices
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
- Disabled devices

☑ Show Bindings

Fig. 77: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:



Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters

- Installed and ready to use devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
    - ☑ TwinCAT Ethernet Protocol for all Network Adapters
    - ☑ TwinCAT Rt-Ethernet Intermediate Driver
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☑ TwinCAT Ethernet Protocol for all Network Adapters
    - ☐ TwinCAT Rt-Ethernet Intermediate Driver
- Compatible devices
- Incompatible devices
- Disabled devices

WRONG: both driver enabled

☑ Show Bindings

Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters

- Installed and ready to use devices
- Compatible devices
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☐ TwinCAT Rt-Ethernet Intermediate Driver
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
    - ☑ TwinCAT Rt-Ethernet Intermediate Driver
- Disabled devices

WRONG: Intermediate enabled

☑ Show Bindings

Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters

- Installed and ready to use devices
- Compatible devices
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☑ TwinCAT Ethernet Protocol for all Network Adapters
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
    - ☑ TwinCAT Ethernet Protocol for all Network Adapters
- Disabled devices

WRONG: enabled for all network adapters

☑ Show Bindings

Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters

- Installed and ready to use devices
- Compatible devices
  - LAN-Verbindung - Intel(R) 82574L Gigabit Network Connection
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
- Disabled devices

WRONG: no TwinCAT driver

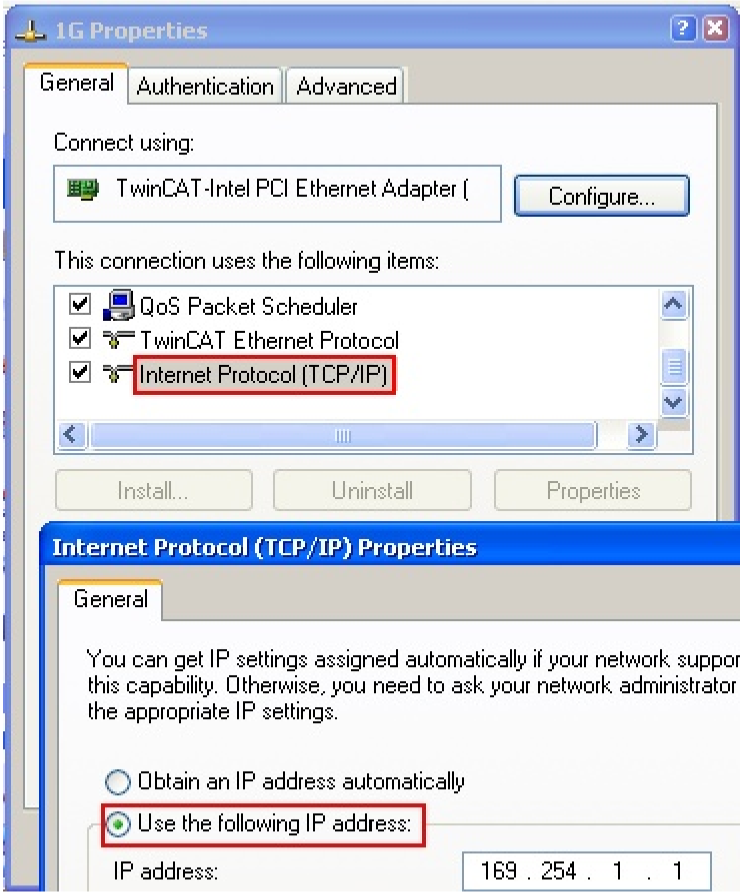
☐ Show Bindings

Fig. 78: Incorrect driver settings for the Ethernet port

IP address of the port used

IP address/DHCP

In most cases an Ethernet port that is configured as an EtherCAT device will not transport general IP packets. For this reason and in cases where an EL6601 or similar devices are used it is useful to specify a fixed IP address for this port via the “Internet Protocol TCP/IP” driver setting and to disable DHCP. In this way the delay associated with the DHCP client for the Ethernet port assigning itself a default IP address in the absence of a DHCP server is avoided. A suitable address space is 192.168.x.x, for example.



1G Properties

General | Authentication | Advanced

Connect using:

TwinCAT-Intel PCI Ethernet Adapter (

This connection uses the following items:

☑ QoS Packet Scheduler
☑ TwinCAT Ethernet Protocol
☑ Internet Protocol (TCP/IP)

Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network suppor this capability. Otherwise, you need to ask your network administrator the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 169 . 254 . 1 . 1

Fig. 79: 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 [▶ 91] 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”

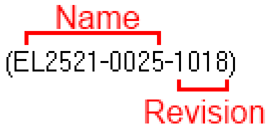
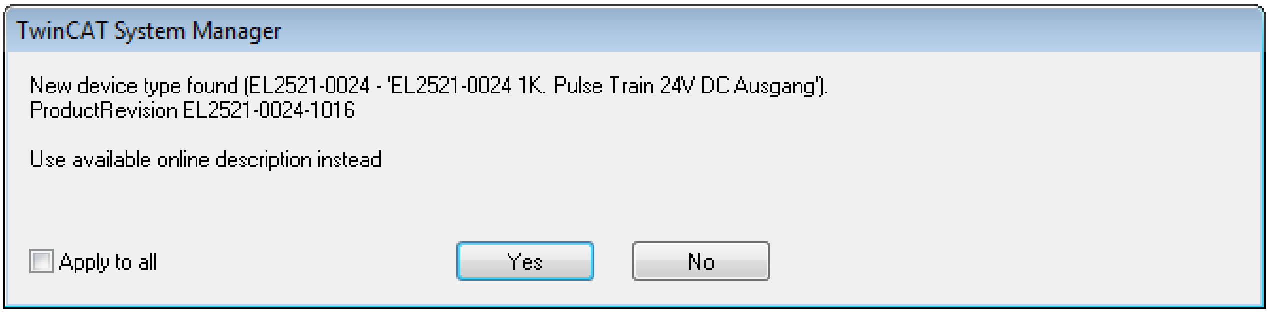


Fig. 80: Identifier structure

The order identifier consisting of name + type (here: EL2521-0025) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes [▶ 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.

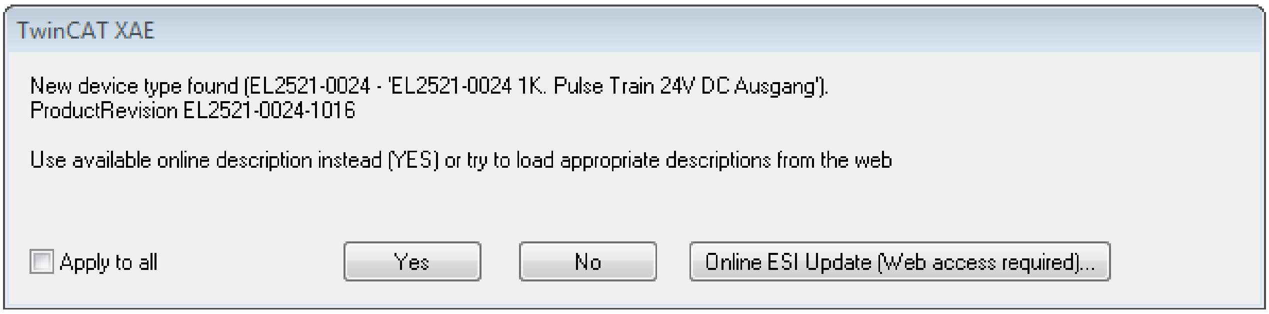

TwinCAT System Manager

New device type found [EL2521-0024 - 'EL2521-0024 1K. Pulse Train 24V DC Ausgang'].
ProductRevision EL2521-0024-1016

Use available online description instead

Fig. 81: OnlineDescription information window (TwinCAT 2)

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


TwinCAT XAE

New device type found [EL2521-0024 - 'EL2521-0024 1K. Pulse Train 24V DC Ausgang'].
ProductRevision EL2521-0024-1016

Use available online description instead (YES) or try to load appropriate descriptions from the web

Fig. 82: Information window OnlineDescription (TwinCAT 3)

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

NOTICE

Changing the “usual” configuration through a scan

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

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

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

Refer in particular to the chapter “General notes on the use of Beckhoff EtherCAT IO components” and for manual configuration to the chapter “Offline configuration creation [▸ 92]”.

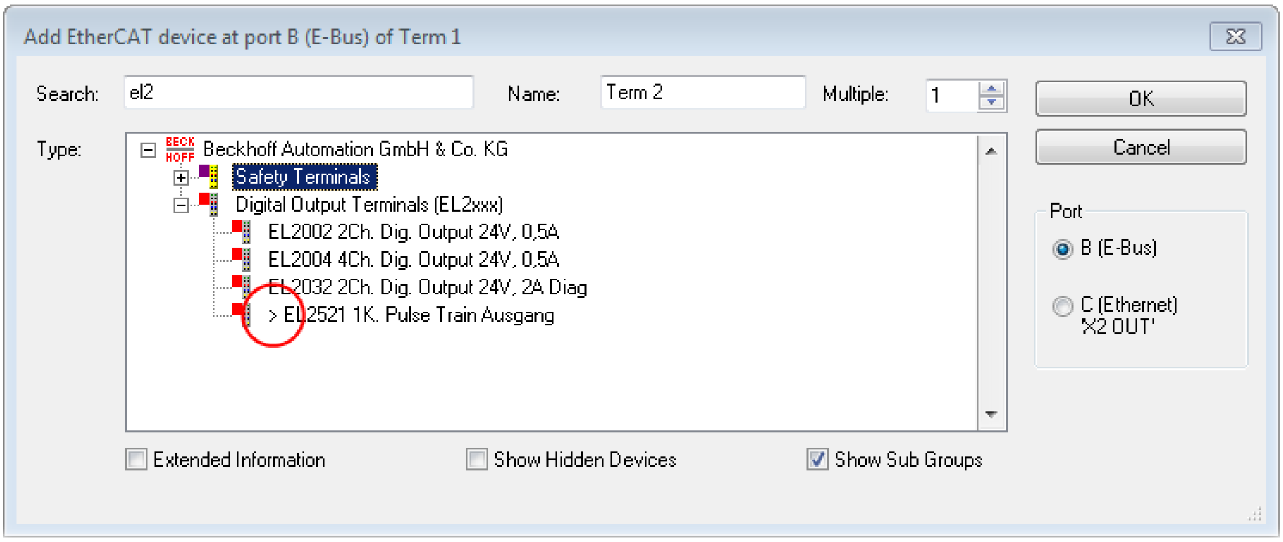
If the OnlineDescription is used regardless, the System Manager reads a copy of the device description from the EEPROM in the EtherCAT slave. In complex slaves the size of the EEPROM may not be sufficient for the complete ESI, in which case the ESI would be *incomplete* in the configurator. Therefore 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. 83: File OnlineDescription.xml created by the System Manager

Is a slave desired to be added manually to the configuration at a later stage, online created slaves are indicated by a prepended symbol “>” in the selection list (see Figure *Indication of an online recorded ESI of EL2521 as an example)*.



Add EtherCAT device at port B (E-Bus) of Term 1

Search: el2

Name: Term 2

Multiple: 1

Type:

- Beckhoff Automation GmbH & Co. KG
  - Safety Terminals
  - Digital Output Terminals (EL2xxx)
    - EL2002 2Ch. Dig. Output 24V, 0,5A
    - EL2004 4Ch. Dig. Output 24V, 0,5A
    - EL2032 2Ch. Dig. Output 24V, 2A Diag
    - > EL2521 1K. Pulse Train Ausgang

Port: ● B (E-Bus) ○ C (Ethernet) 'X2 OUT'

□ Extended Information □ Show Hidden Devices ☑ Show Sub Groups

Fig. 84: Indication of an online recorded ESI of EL2521 as an example

If such ESI files are used and the manufacturer's files become available later, the file OnlineDescription.xml should be deleted as follows:

- close all System Manager windows
- restart TwinCAT in Config mode
- delete “OnlineDescription0000...xml”
- restart TwinCAT System Manager

This file should not be visible after this procedure, if necessary press <F5> to update

OnlineDescription for TwinCAT 3.x

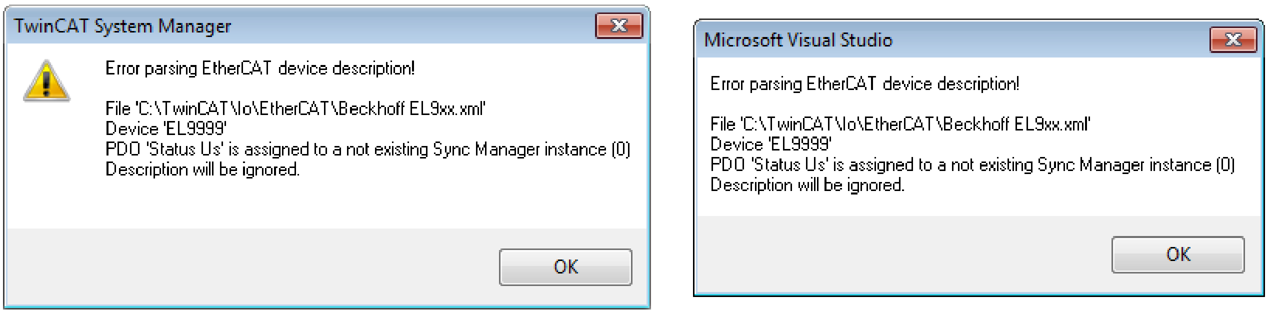
In addition to the file described above “OnlineDescription0000...xml”, a so called EtherCAT cache with new discovered devices is created by TwinCAT 3.x, e.g. under Windows 7:

C:\User\[USERNAME]\AppData\Roaming\Beckhoff\TwinCAT3\Components\Base\EtherCATCache.xml

(Please note the language settings of the OS!)
You have to delete this file, too.

Faulty ESI file

If an ESI file is faulty and the System Manager is unable to read it, the System Manager brings up an information window.



TwinCAT System Manager

Error parsing EtherCAT device description!

File 'C:\TwinCAT\Io\EtherCAT\Beckhoff EL9xx.xml'
Device 'EL9999'
PDO 'Status Us' is assigned to a not existing Sync Manager instance (0)
Description will be ignored.

Microsoft Visual Studio

Error parsing EtherCAT device description!

File 'C:\TwinCAT\Io\EtherCAT\Beckhoff EL9xx.xml'
Device 'EL9999'
PDO 'Status Us' is assigned to a not existing Sync Manager instance (0)
Description will be ignored.

Fig. 85: 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:

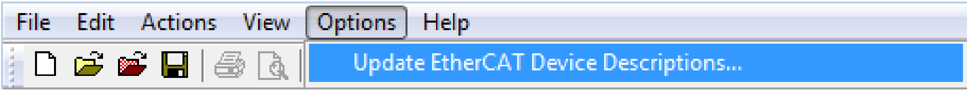
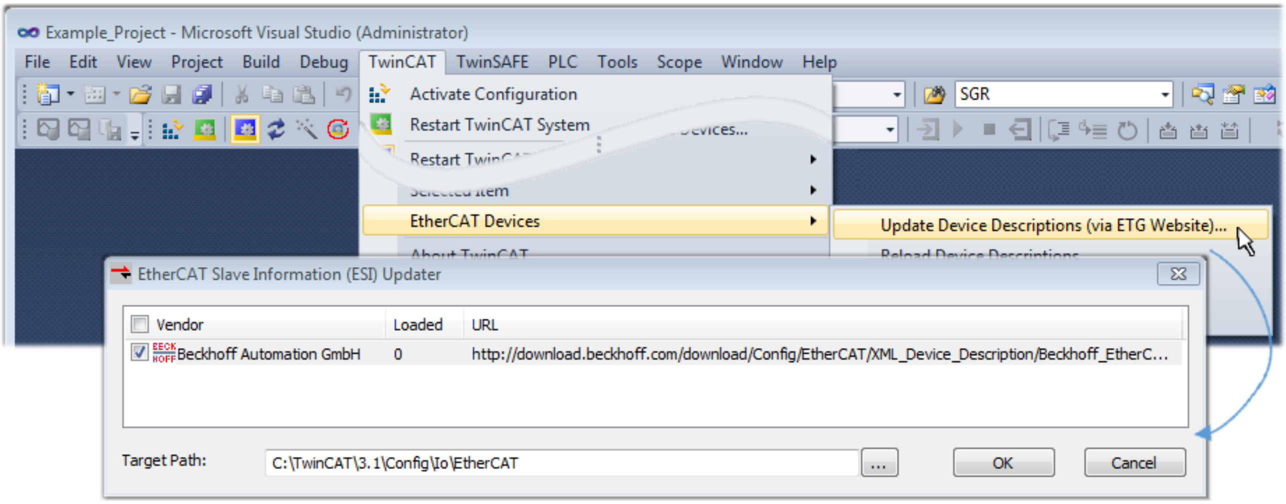


Fig. 86: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:



Vendor	Loaded	URL
☑ Beckhoff Automation GmbH	0	http://download.beckhoff.com/download/Config/EtherCAT/XML_Device_Description/Beckhoff_EtherC...

Target Path: C:\TwinCAT\3.1\Config\Io\EtherCAT

Fig. 87: 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” [▶ 87].

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 [▶ 97] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 98]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 101]

The scan with existing configuration [▶ 102] 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.

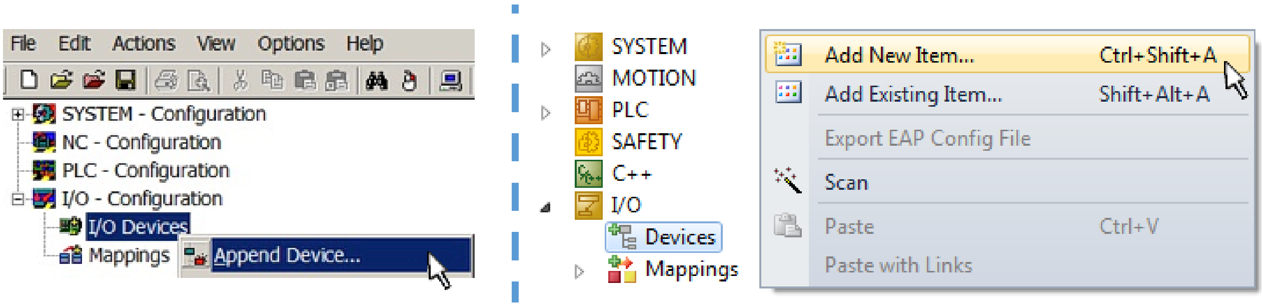


Fig. 88: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

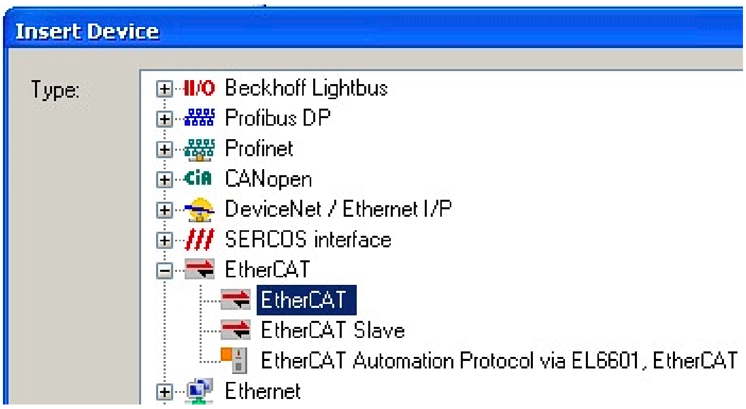


Fig. 89: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

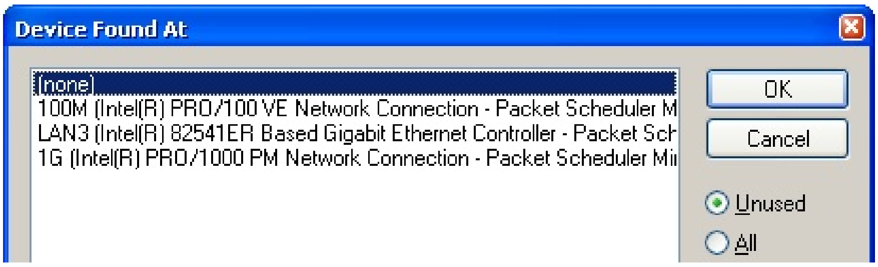
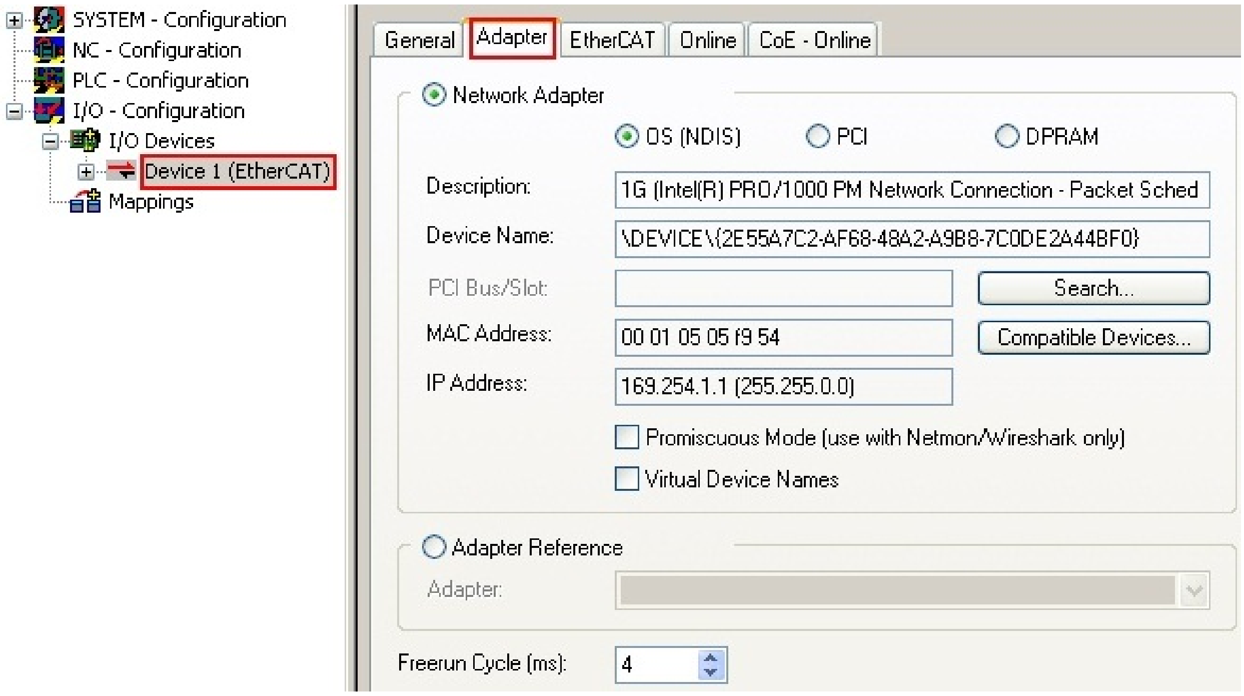

Device Found At
(none)
100M (Intel(R) PRO/100 VE Network Connection - Packet Scheduler M
LAN3 (Intel(R) 82541ER Based Gigabit Ethernet Controller - Packet Sch
1G (Intel(R) PRO/1000 PM Network Connection - Packet Scheduler Mi
OK
Cancel
● Unused
○ All

Fig. 90: 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)”.



● Network Adapter

● OS (NDIS) ○ PCI ○ DPRAM

Description: 1G (Intel(R) PRO/1000 PM Network Connection - Packet Sched

Device Name: \DEVICE\{2E55A7C2-AF68-48A2-A9B8-7C0DE2A44BF0}

PCI Bus/Slot:

MAC Address: 00 01 05 05 f9 54

IP Address: 169.254.1.1 (255.255.0.0)

□ Promiscuous Mode (use with Netmon/Wireshark only)

□ Virtual Device Names

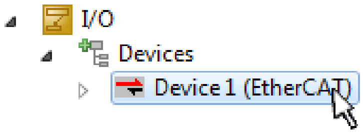
○ Adapter Reference

Adapter:

Freerun Cycle (ms): 4

Fig. 91: EtherCAT device properties (TwinCAT 2)

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



Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 81].

Defining EtherCAT slaves

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

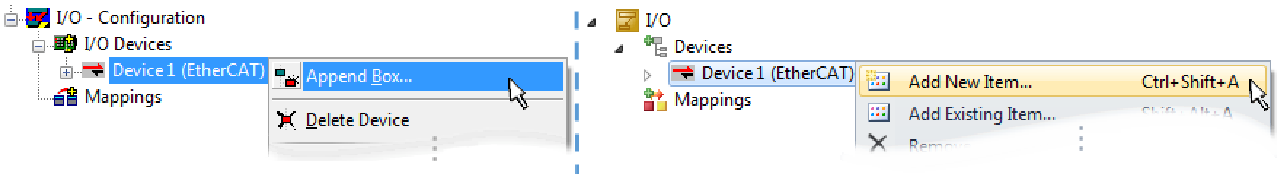


Fig. 92: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

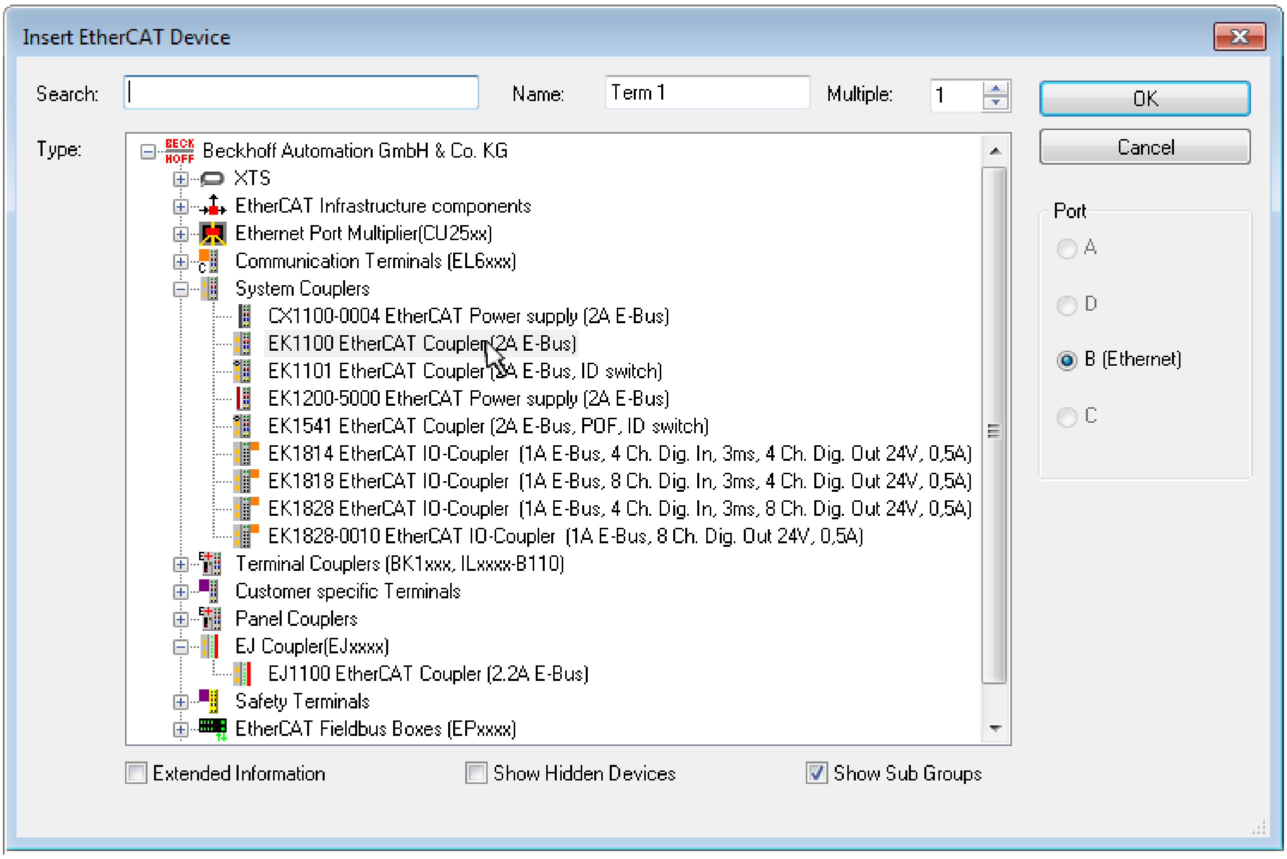
Only devices are offered for selection that can be appended to the previously selected device. Therefore, the physical layer available for this port is also displayed (Fig. “Selection dialog for new EtherCAT device”, A). In the case of cable-based Fast-Ethernet physical layer with PHY transfer, then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

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector

- “E-Bus”: LVDS “terminal bus”, EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

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


Insert EtherCAT Device

Search:
Name: Term 1
Multiple: 1

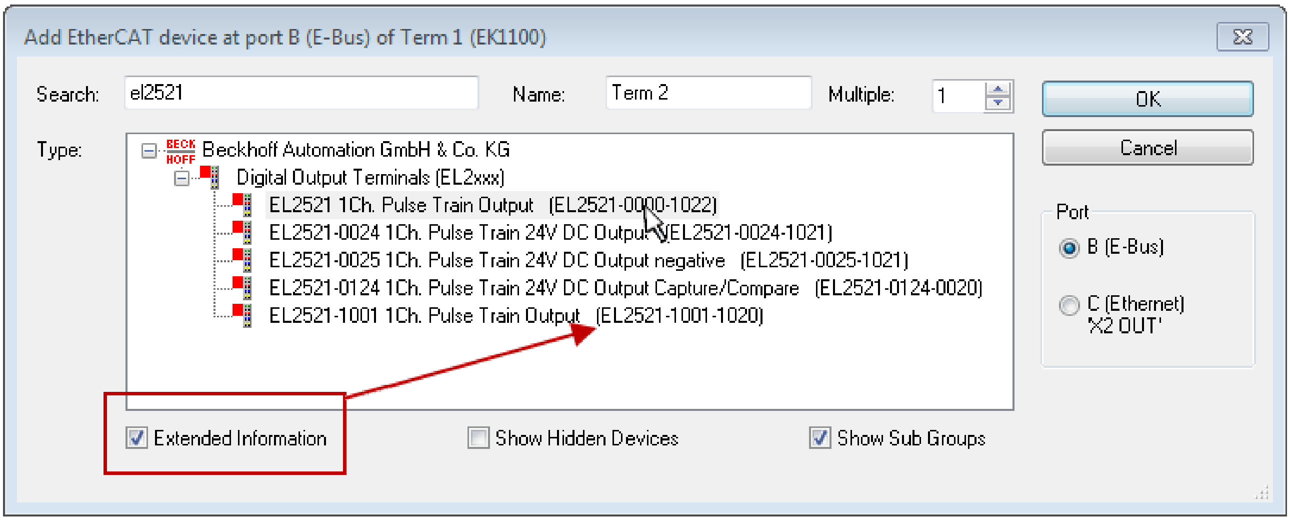
Type:
- Beckhoff Automation GmbH & Co. KG
  - XTS
  - EtherCAT Infrastructure components
  - Ethernet Port Multiplier(CU25xx)
  - Communication Terminals (EL6xxx)
  - System Couplers
    - CX1100-0004 EtherCAT Power supply (2A E-Bus)
    - EK1100 EtherCAT Coupler (2A E-Bus)
    - EK1101 EtherCAT Coupler (2A E-Bus, ID switch)
    - EK1200-5000 EtherCAT Power supply (2A E-Bus)
    - EK1541 EtherCAT Coupler (2A E-Bus, POF, ID switch)
    - EK1814 EtherCAT IO-Coupler (1A E-Bus, 4 Ch. Dig. In, 3ms, 4 Ch. Dig. Out 24V, 0,5A)
    - EK1818 EtherCAT IO-Coupler (1A E-Bus, 8 Ch. Dig. In, 3ms, 4 Ch. Dig. Out 24V, 0,5A)
    - EK1828 EtherCAT IO-Coupler (1A E-Bus, 4 Ch. Dig. In, 3ms, 8 Ch. Dig. Out 24V, 0,5A)
    - EK1828-0010 EtherCAT IO-Coupler (1A E-Bus, 8 Ch. Dig. Out 24V, 0,5A)
  - Terminal Couplers (BK1xxx, ILxxxx-B110)
  - Customer specific Terminals
  - Panel Couplers
  - EJ Coupler(EJxxxx)
    - EJ1100 EtherCAT Coupler (2.2A E-Bus)
  - Safety Terminals
  - EtherCAT Fieldbus Boxes (EPxxxx)

Port: ○ A ○ D ● B (Ethernet) ○ C

□ Extended Information □ Show Hidden Devices ☑ Show Sub Groups

Fig. 93: 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”.


Add EtherCAT device at port B (E-Bus) of Term 1 (EK1100)

Search: el2521
Name: Term 2
Multiple: 1

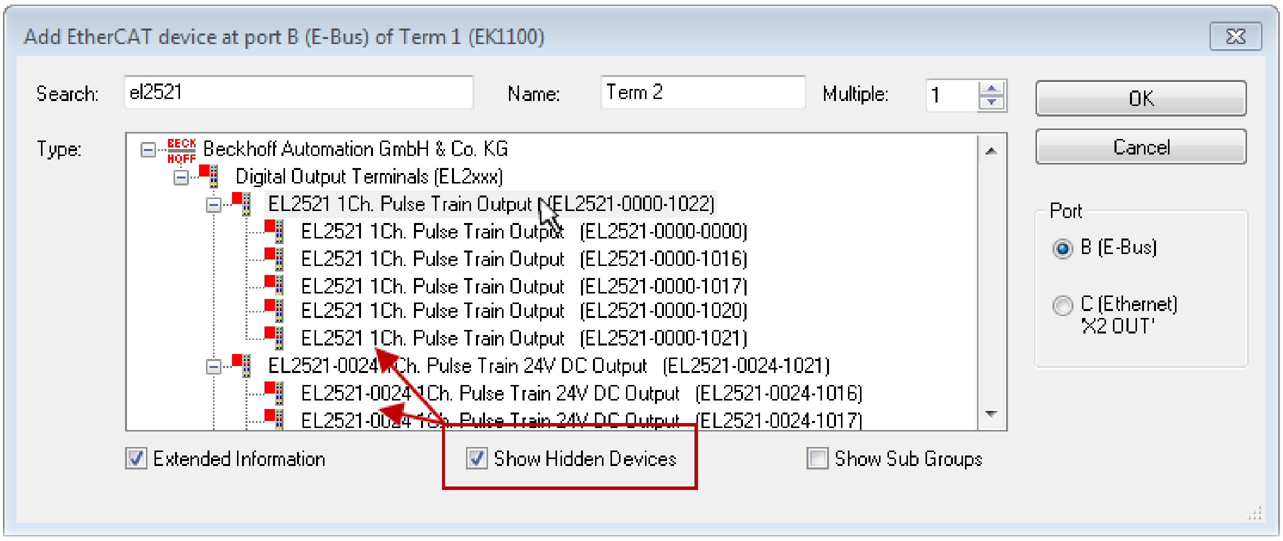
Type:
- Beckhoff Automation GmbH & Co. KG
  - Digital Output Terminals (EL2xxx)
    - EL2521 1Ch. Pulse Train Output (EL2521-0000-1022)
    - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1021)
    - EL2521-0025 1Ch. Pulse Train 24V DC Output negative (EL2521-0025-1021)
    - EL2521-0124 1Ch. Pulse Train 24V DC Output Capture/Compare (EL2521-0124-0020)
    - EL2521-1001 1Ch. Pulse Train Output (EL2521-1001-1020)

Port: ● B (E-Bus) ○ C (Ethernet) 'X2 OUT'

☑ Extended Information □ Show Hidden Devices ☑ Show Sub Groups

Fig. 94: 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”.



Add EtherCAT device at port B (E-Bus) of Term 1 (EK1100)

Search: el2521

Name: Term 2

Multiple: 1

OK

Cancel

Type:

- Beckhoff Automation GmbH & Co. KG
  - Digital Output Terminals (EL2xxx)
    - EL2521 1Ch. Pulse Train Output (EL2521-0000-1022)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-0000)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-1016)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-1017)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-1020)
      - EL2521 1Ch. Pulse Train Output (EL2521-0000-1021)
    - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1021)
      - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1016)
      - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1017)

Port

● B (E-Bus)

○ C (Ethernet) 'X2 OUT'

☑ Extended Information ☑ Show Hidden Devices □ Show Sub Groups

Fig. 95: Display of previous revisions

Device selection based on revision, compatibility

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

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (-**1019**, -**1020**) can be used in practice.

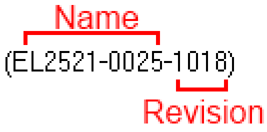


Fig. 96: Name/revision of the terminal

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

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

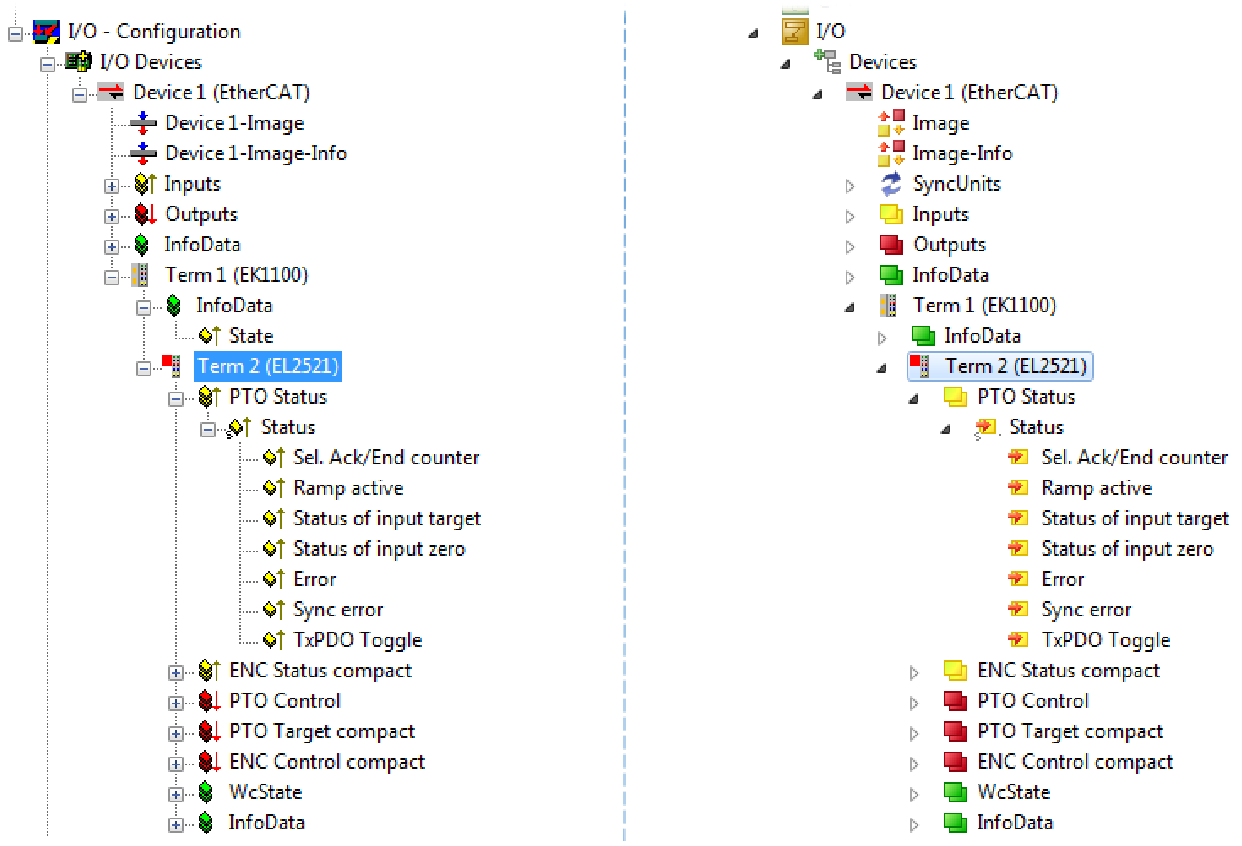


Fig. 97: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

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: Config Mode .
- on TwinCAT 3 within the user interface of the development environment by a symbol .

TwinCAT can be set into this mode:

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

Online scanning in Config mode

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

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

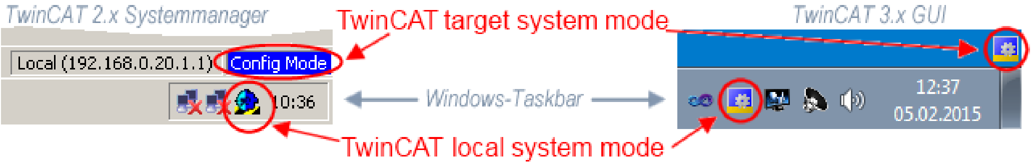


Fig. 98: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

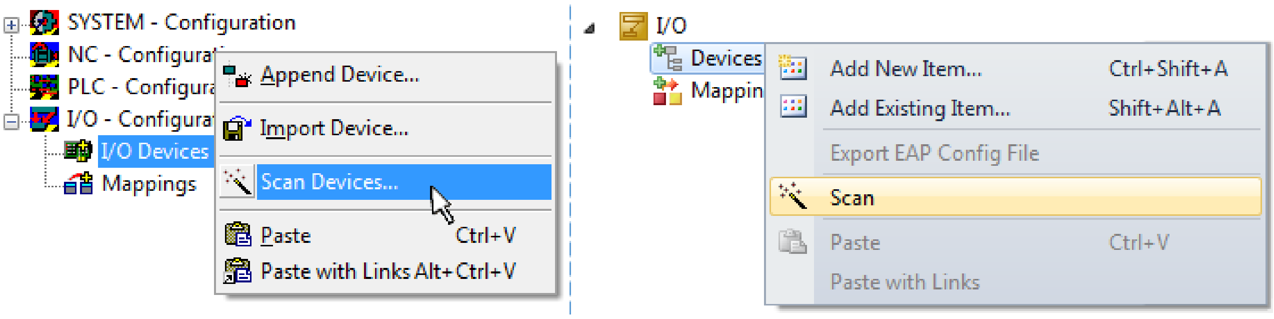
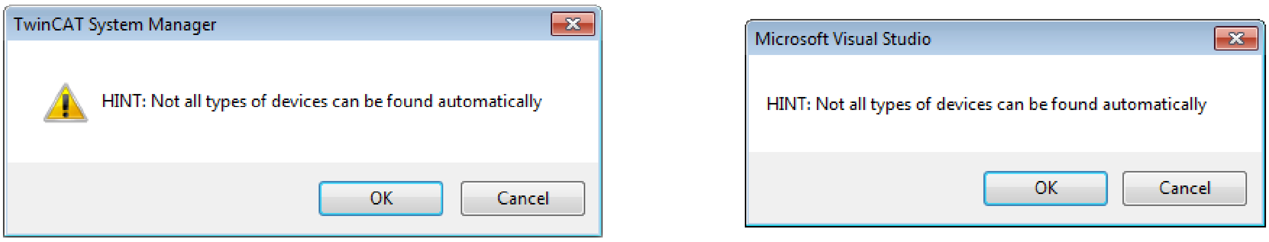


Fig. 99: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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.


TwinCAT System Manager

HINT: Not all types of devices can be found automatically

OK Cancel

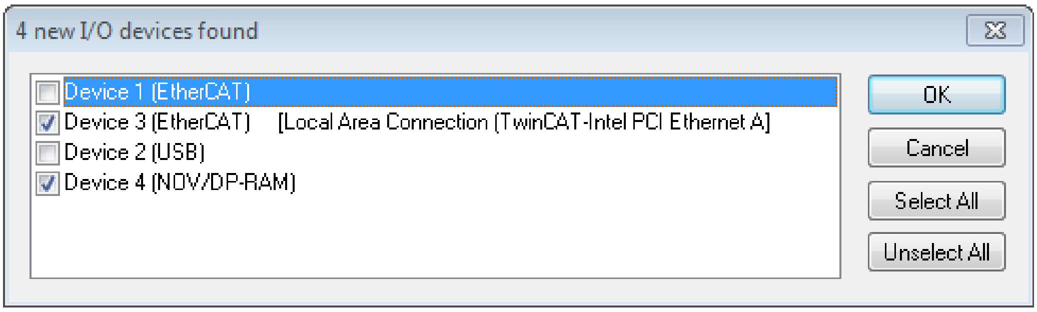
Microsoft Visual Studio

HINT: Not all types of devices can be found automatically

OK Cancel

Fig. 100: 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” .



4 new I/O devices found

□ Device 1 (EtherCAT)
☑ Device 3 (EtherCAT)   [Local Area Connection (TwinCAT-Intel PCI Ethernet A]
□ Device 2 (USB)
☑ Device 4 (NOV/DP-RAM)

Fig. 101: 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 [▸ 81].

Detecting/Scanning the EtherCAT devices

Online scan functionality

During a scan the master queries the identity information of the EtherCAT slaves from the slave EEPROM. The name and revision are used for determining the type. The respective devices are located in the stored ESI data and integrated in the configuration tree in the default state defined there.

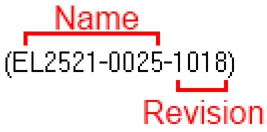


Fig. 102: Example default state

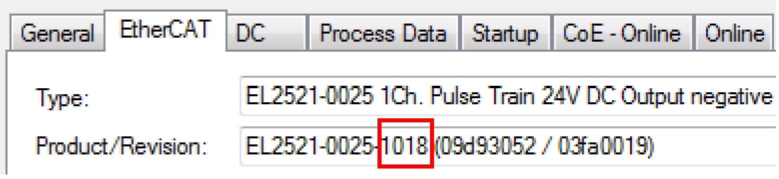
NOTICE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▸ 102] 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:



Type: EL2521-0025 1Ch. Pulse Train 24V DC Output negative

Product/Revision: EL2521-0025-1018 (09d93052 / 03fa0019)

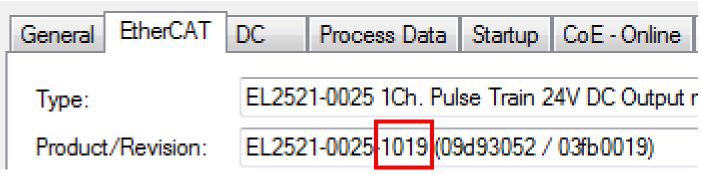
Fig. 103: Installing EthetCAT terminal with revision -1018

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 [▶ 102] 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:



Type: EL2521-0025 1Ch. Pulse Train 24V DC Output r

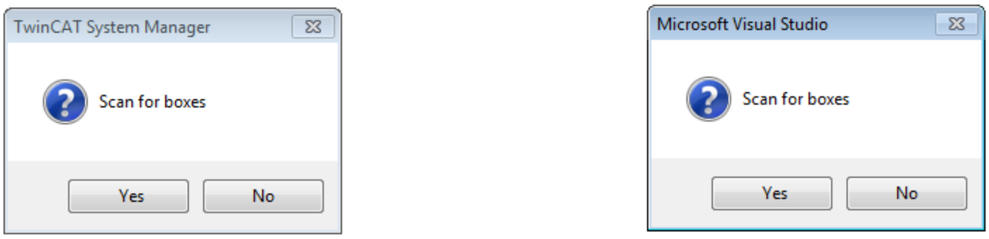
Product/Revision: EL2521-0025-1019 (09d93052 / 03fb0019)

Fig. 104: Detection of EtherCAT terminal with revision -1019

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

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration “B2.tsm” created in this way. Þ if series machine production is established, the scan should only be performed for informative purposes for comparison with a defined initial configuration. Changes are to be made with care!

If an EtherCAT device was created in the configuration (manually or through a scan), the I/O field can be scanned for devices/slaves.


TwinCAT System Manager

Scan for boxes

Microsoft Visual Studio

Scan for boxes

Fig. 105: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

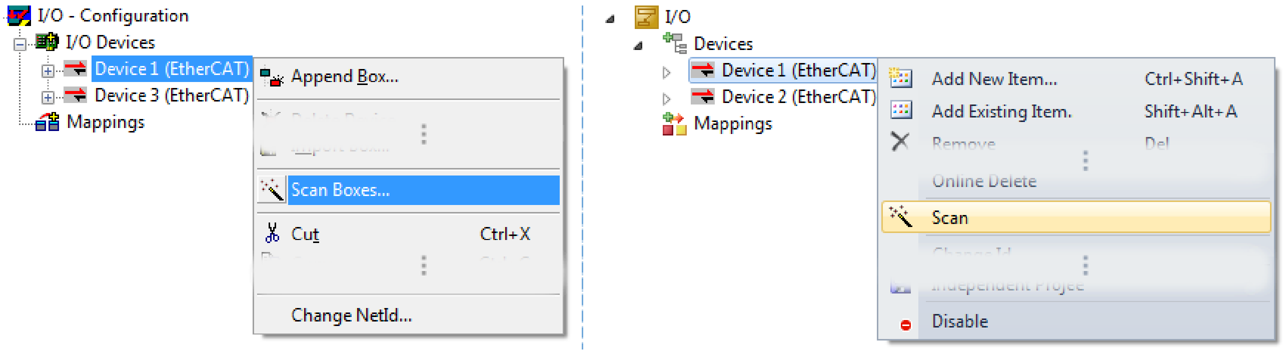


Fig. 106: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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

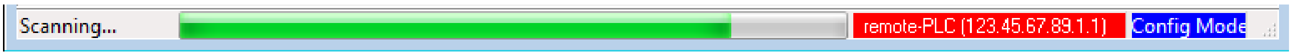


Fig. 107: *Scan progressexemplary by TwinCAT 2*

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

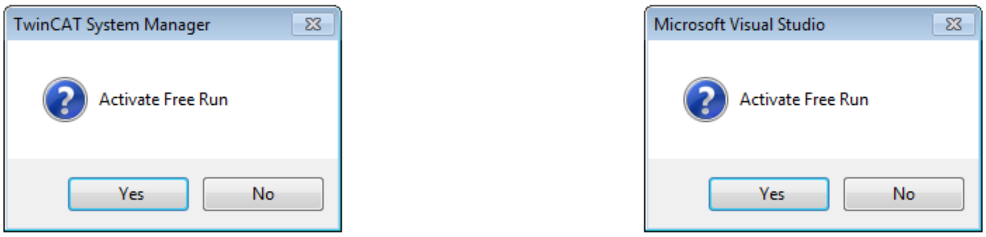


Fig. 108: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

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).

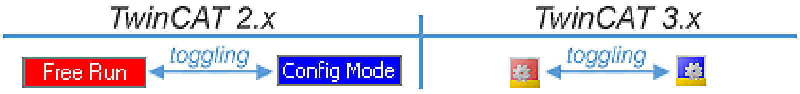


Fig. 109: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

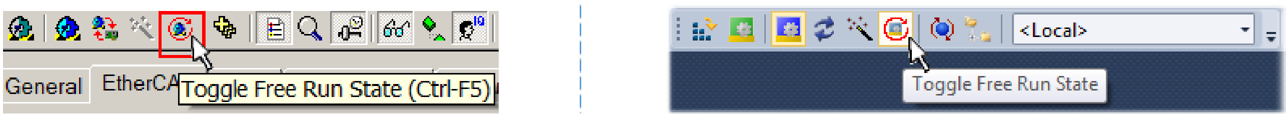
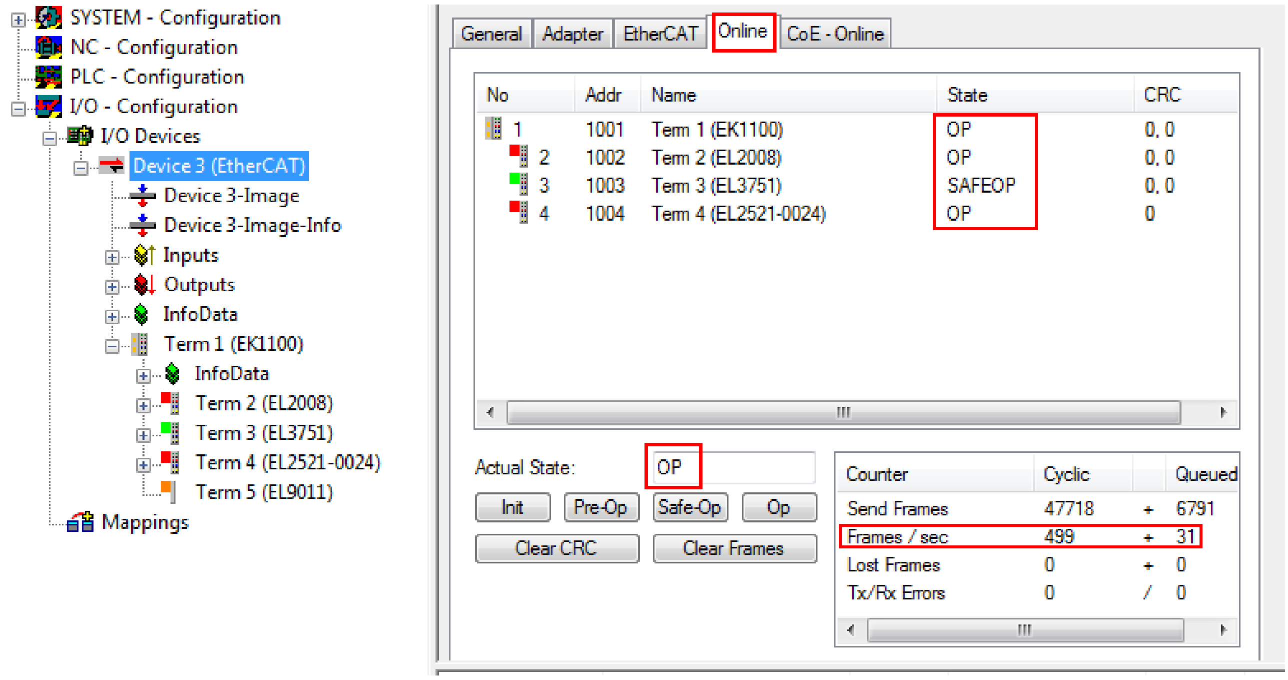


Fig. 110: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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



- SYSTEM - Configuration
- NC - Configuration
- PLC - Configuration
- I/O - Configuration
  - I/O Devices
    - Device 3 (EtherCAT)
      - Device 3-Image
      - Device 3-Image-Info
      - Inputs
      - Outputs
      - InfoData
      - Term 1 (EK1100)
        - InfoData
        - Term 2 (EL2008)
        - Term 3 (EL3751)
        - Term 4 (EL2521-0024)
        - Term 5 (EL9011)
  - Mappings

No	Addr	Name	State	CRC
1	1001	Term 1 (EK1100)	OP	0, 0
2	1002	Term 2 (EL2008)	OP	0, 0
3	1003	Term 3 (EL3751)	SAFEOP	0, 0
4	1004	Term 4 (EL2521-0024)	OP	0

Actual State: OP

Counter	Cyclic		Queued
Send Frames	47718	+	6791
Frames / sec	499	+	31
Lost Frames	0	+	0
Tx/Rx Errors	0	/	0

Fig. 111: 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

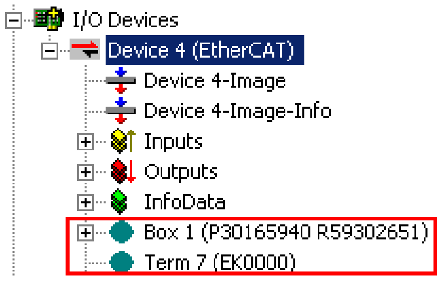
The configuration is now complete. It can be modified as described under manual procedure [▶ 92].

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.



- I/O Devices
  - Device 4 (EtherCAT)
    - Device 4-Image
    - Device 4-Image-Info
    - Inputs
    - Outputs
    - InfoData
    - Box 1 (P30165940 R59302651)
    - Term 7 (EK0000)

Fig. 112: Faulty identification

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

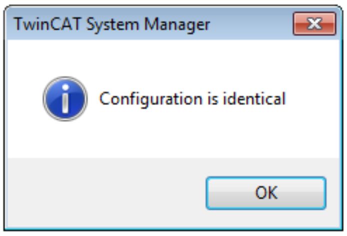
Scan over existing Configuration

NOTICE

Change of the configuration after comparison

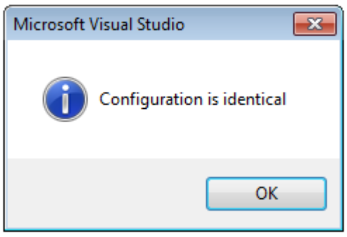
With this scan (TwinCAT 2.11 or 3.1) only the device properties vendor (manufacturer), device name and revision are compared at present! A “ChangeTo” or “Copy” should only be carried out with care, taking into consideration the Beckhoff IO compatibility rule (see above). The device configuration is then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

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



TwinCAT System Manager

Configuration is identical

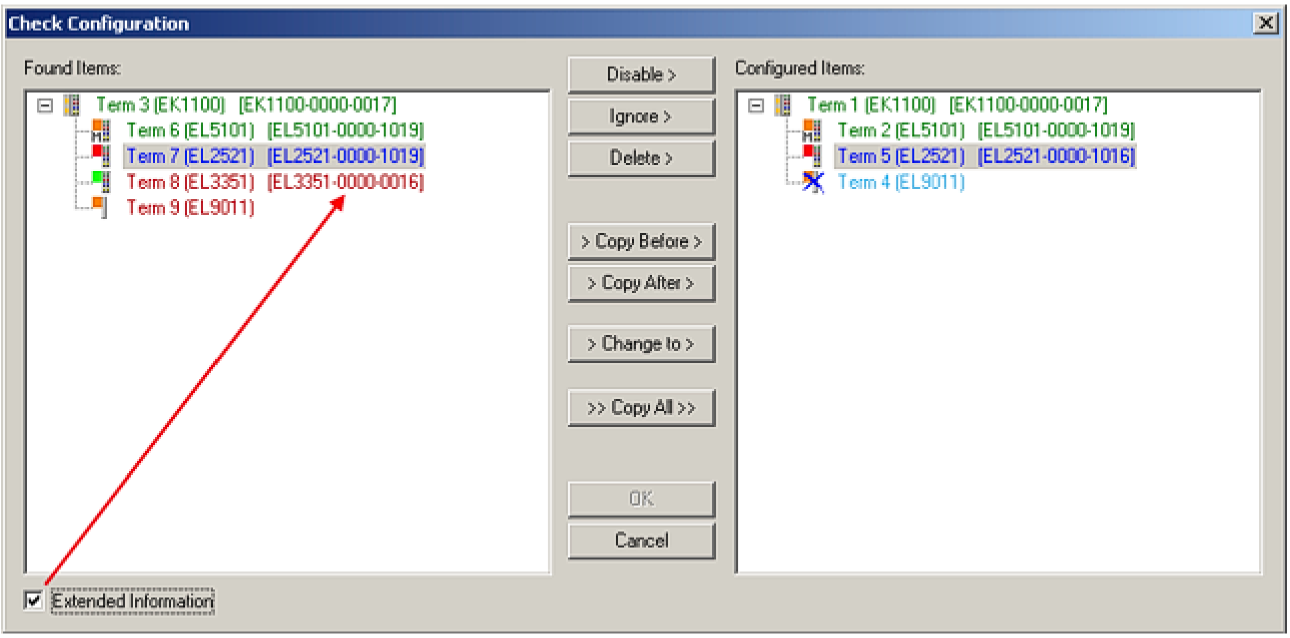


Microsoft Visual Studio

Configuration is identical

Fig. 113: 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.



Check Configuration

Found Items:

- Term 3 (EK1100) [EK1100-0000-0017]
  - Term 6 (EL5101) [EL5101-0000-1019]
  - Term 7 (EL2521) [EL2521-0000-1019]
  - Term 8 (EL3351) [EL3351-0000-0016]
  - Term 9 (EL9011)

Configured Items:

- Term 1 (EK1100) [EK1100-0000-0017]
  - Term 2 (EL5101) [EL5101-0000-1019]
  - Term 5 (EL2521) [EL2521-0000-1016]
  - Term 4 (EL9011)

☑ Extended Information

Fig. 114: Correction dialog

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

Color	Explanation
green	This EtherCAT slave matches the entry on the other side. Both type and revision match.
blue	This EtherCAT slave is present on the other side, but in a different revision. This other revision can have other default values for the process data as well as other/additional functions. If the found revision is higher than the configured revision, the slave may be used provided compatibility issues are taken into account. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.
light blue	This EtherCAT slave is ignored ("Ignore" button)
red	• This EtherCAT slave is not present on the other side. • It is present, but in a different revision, which also differs in its properties from the one specified. The compatibility principle then also applies here: if the found revision is higher than the configured revision, use is possible provided compatibility issues are taken into account, since the successor devices should support the functions of the predecessor devices. If the found revision is lower than the configured revision, it is likely that the slave cannot be used. The found device may not support all functions that the master expects based on the higher revision number.

Device selection based on revision, compatibility

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

device revision in the system >= device revision in the configuration

This also enables subsequent replacement of devices without changing the configuration (different specifications are possible for drives).

Example

If an EL2521-0025-**1018** is specified in the configuration, an EL2521-0025-**1018** or higher (-**1019**, -**1020**) can be used in practice.

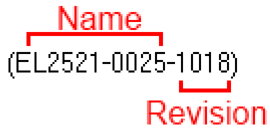
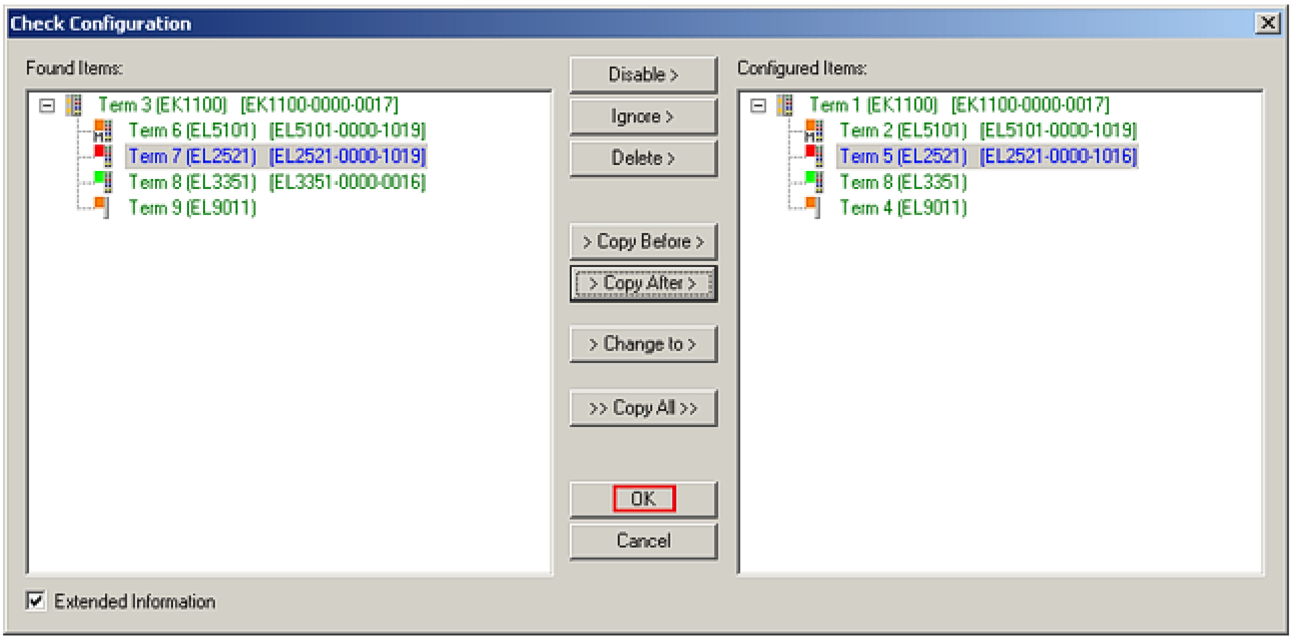


Fig. 115: Name/revision of the terminal

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

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



Check Configuration

Found Items:

- Term 3 (EK1100) [EK1100-0000-0017]
  - Term 6 (EL5101) [EL5101-0000-1019]
  - Term 7 (EL2521) [EL2521-0000-1019]
  - Term 8 (EL3351) [EL3351-0000-0016]
  - Term 9 (EL9011)

Configured Items:

- Term 1 (EK1100) [EK1100-0000-0017]
  - Term 2 (EL5101) [EL5101-0000-1019]
  - Term 5 (EL2521) [EL2521-0000-1016]
  - Term 8 (EL3351)
  - Term 4 (EL9011)

☑ Extended Information

Fig. 116: 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.

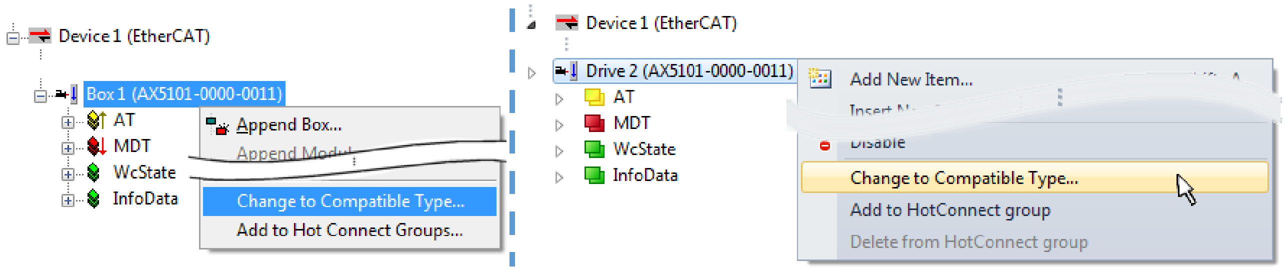


Fig. 117: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

The following elements in the ESI of an EtherCAT device are compared by TwinCAT and assumed to be the same in order to decide whether a device is indicated as "compatible":

- Physics (e.g. RJ45, Ebus...)

- FMMU (additional ones are allowed)

- SyncManager (SM, additional ones are allowed)

- EoE (attributes MAC, IP)

- CoE (attributes SdoInfo, PdoAssign, PdoConfig, PdoUpload, CompleteAccess)

- FoE

- PDO (process data: Sequence, SyncUnit SU, SyncManager SM, EntryCount, Ent-ry.Datatype)

This function is preferably to be used on AX5000 devices.

Change to Alternative Type

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

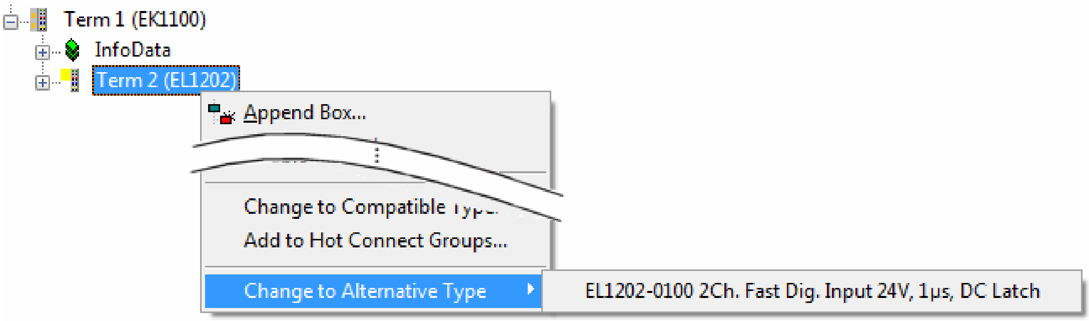

Term 1 (EK1100)
InfoData
Term 2 (EL1202)
Append Box...
Change to Compatible Type...
Add to Hot Connect Groups...
Change to Alternative Type
EL1202-0100 2Ch. Fast Dig. Input 24V, 1µs, DC Latch

Fig. 118: 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).

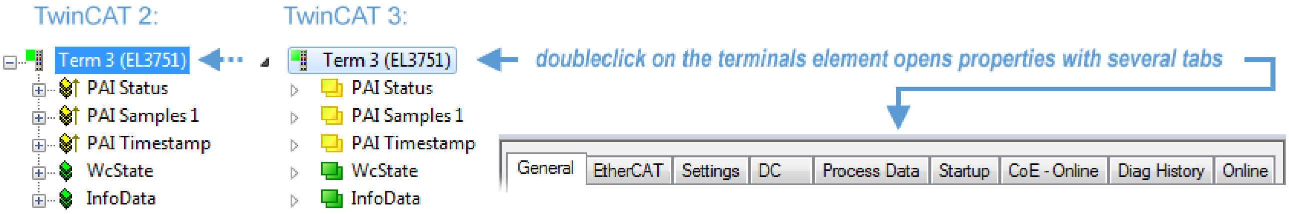
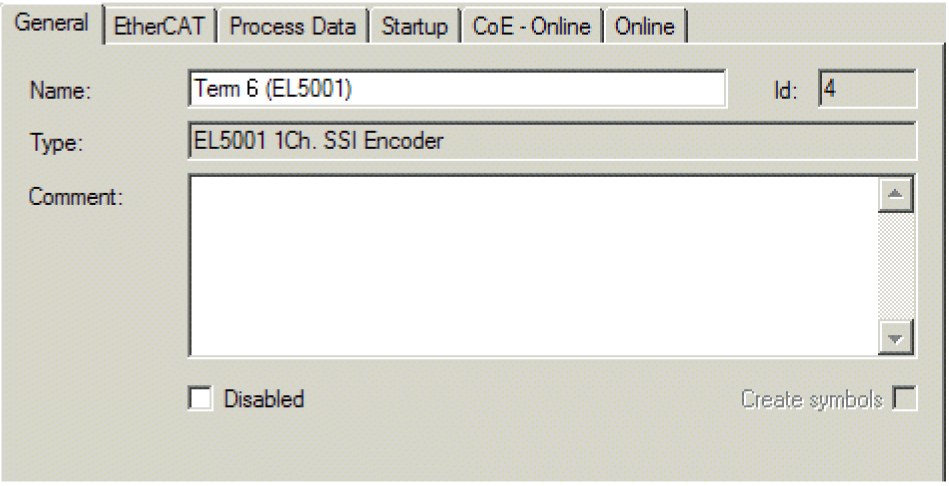


Fig. 119: Branch element as terminal EL3751

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


General | EtherCAT | Process Data | Startup | CoE - Online | Online

Name: Term 6 (EL5001) Id: 4

Type: EL5001 1Ch. SSI Encoder

Comment:

□ Disabled Create symbols □

Fig. 120: “General” tab

Name Name of the EtherCAT device

Id Number of the EtherCAT device

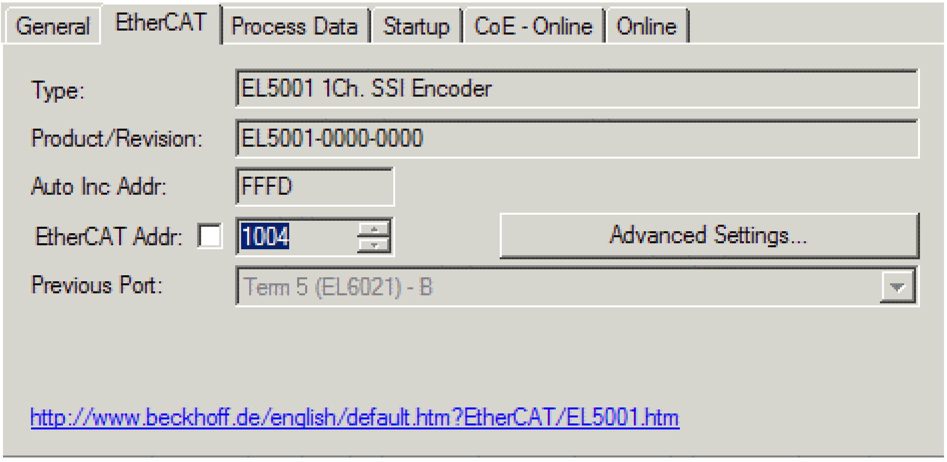
Type EtherCAT device type

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



Type: EL5001 1Ch. SSI Encoder

Product/Revision: EL5001-0000-0000

Auto Inc Addr: FFFD

EtherCAT Addr: □ 1004

Previous Port: Term 5 (EL6021) - B

http://www.beckhoff.de/english/default.htm?EtherCAT/EL5001.htm

Fig. 121: “EtherCAT” tab

Type EtherCAT device type

Product/Revision Product and revision number of the EtherCAT device

Auto Inc Addr. Auto increment address of the EtherCAT device. The auto increment address can be used for addressing each EtherCAT device in the communication ring through its physical position. Auto increment addressing is used during the start-up phase when the EtherCAT master allocates addresses to the EtherCAT devices. With auto increment addressing the first EtherCAT slave in the ring has the address 0000_{hex}. For each further slave the address is decremented by 1 ($FFFF_{hex}$, $FFFE_{hex}$ etc.).

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

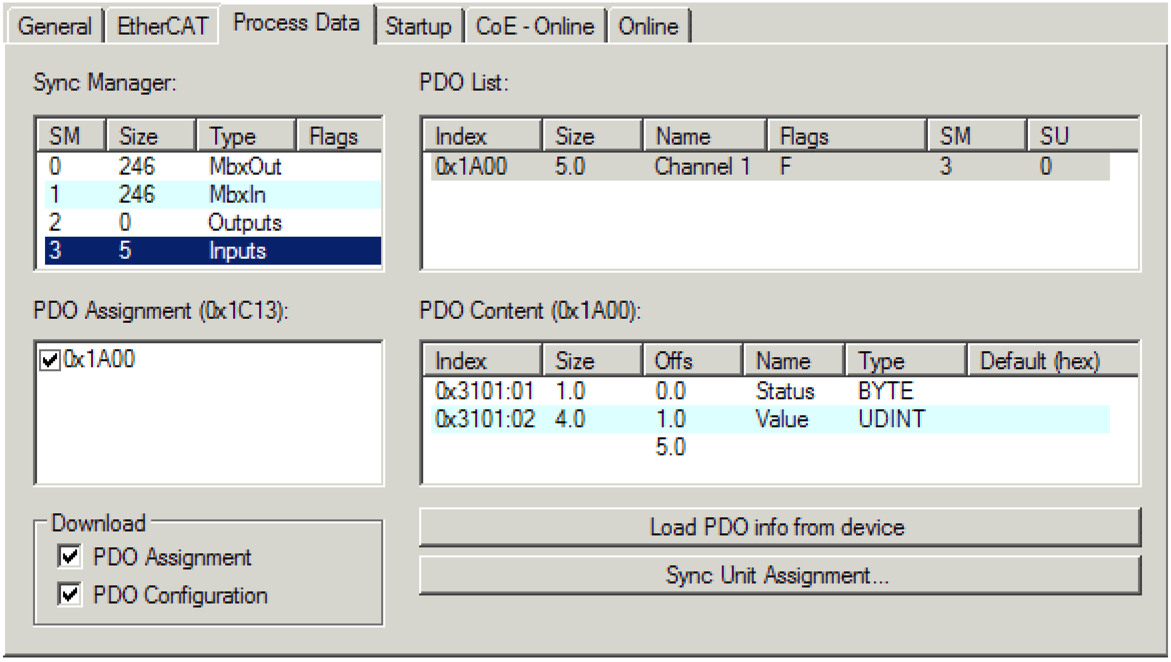
Previous Port Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this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

Advanced Settings This button opens the dialogs for advanced settings.

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

“Process Data” tab

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



Sync Manager:

SM	Size	Type	Flags
0	246	MbxOut	
1	246	MbxIn	
2	0	Outputs	
3	5	Inputs	

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	5.0	Channel 1	F	3	0

PDO Assignment (0x1C13):

☑ 0x1A00

PDO Content (0x1A00):

Index	Size	Offs	Name	Type	Default (hex)
0x3101:01	1.0	0.0	Status	BYTE	
0x3101:02	4.0	1.0	Value	UDINT	
		5.0			

Download

☑ PDO Assignment

☑ PDO Configuration

Fig. 122: “Process Data” tab

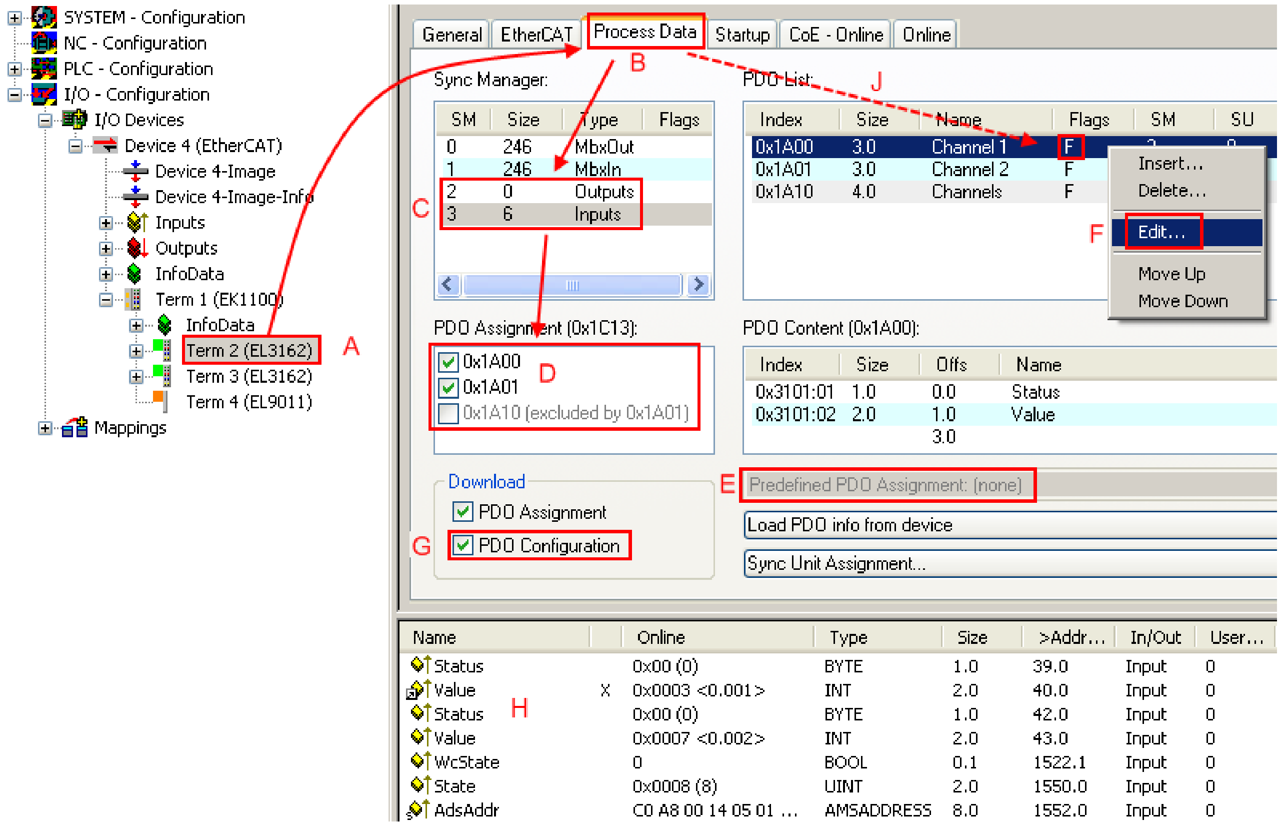
The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which 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”).



SM	Size	Type	Flags
0	246	MbxOut	
1	246	MbxIn	
2	0	Outputs	
3	6	Inputs	

Index	Size	Name	Flags	SM	SU
0x1A00	3.0	Channel 1	F	[illegible]	[illegible]
0x1A01	3.0	Channel 2	F		
0x1A10	4.0	Channels	F		

☑ 0x1A00
☑ 0x1A01
☐ 0x1A10 (excluded by 0x1A01)

Index	Size	Offs	Name
0x3101:01	1.0	0.0	Status
0x3101:02	2.0	1.0	Value
		3.0	

Predefined PDO Assignment: (none)

☑ PDO Assignment
☑ PDO Configuration

Name		Online	Type	Size	>Addr...	In/Out	User...
Status		0x00 (0)	BYTE	1.0	39.0	Input	0
Value	X	0x0003 <0.001>	INT	2.0	40.0	Input	0
Status		0x00 (0)	BYTE	1.0	42.0	Input	0
Value		0x0007 <0.002>	INT	2.0	43.0	Input	0
WcState		0	BOOL	0.1	1522.1	Input	0
State		0x0008 (8)	UINT	2.0	1550.0	Input	0
AdsAddr		C0 A8 00 14 05 01 ...	AMSADDRESS	8.0	1552.0	Input	0

Fig. 123: Configuring the process data

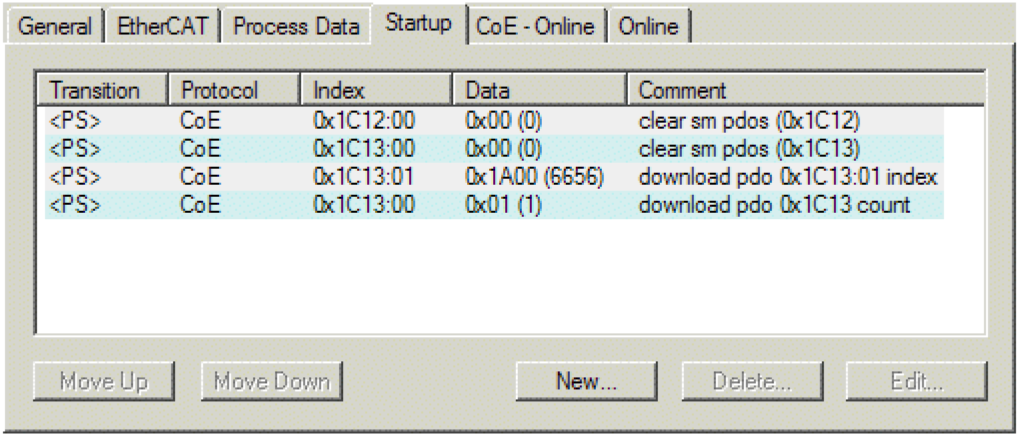
Manual modification of the process data

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

A detailed description [▶ 113] 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.



Transition	Protocol	Index	Data	Comment
<PS>	CoE	0x1C12:00	0x00 (0)	clear sm pdos (0x1C12)
<PS>	CoE	0x1C13:00	0x00 (0)	clear sm pdos (0x1C13)
<PS>	CoE	0x1C13:01	0x1A00 (6656)	download pdo 0x1C13:01 index
<PS>	CoE	0x1C13:00	0x01 (1)	download pdo 0x1C13 count

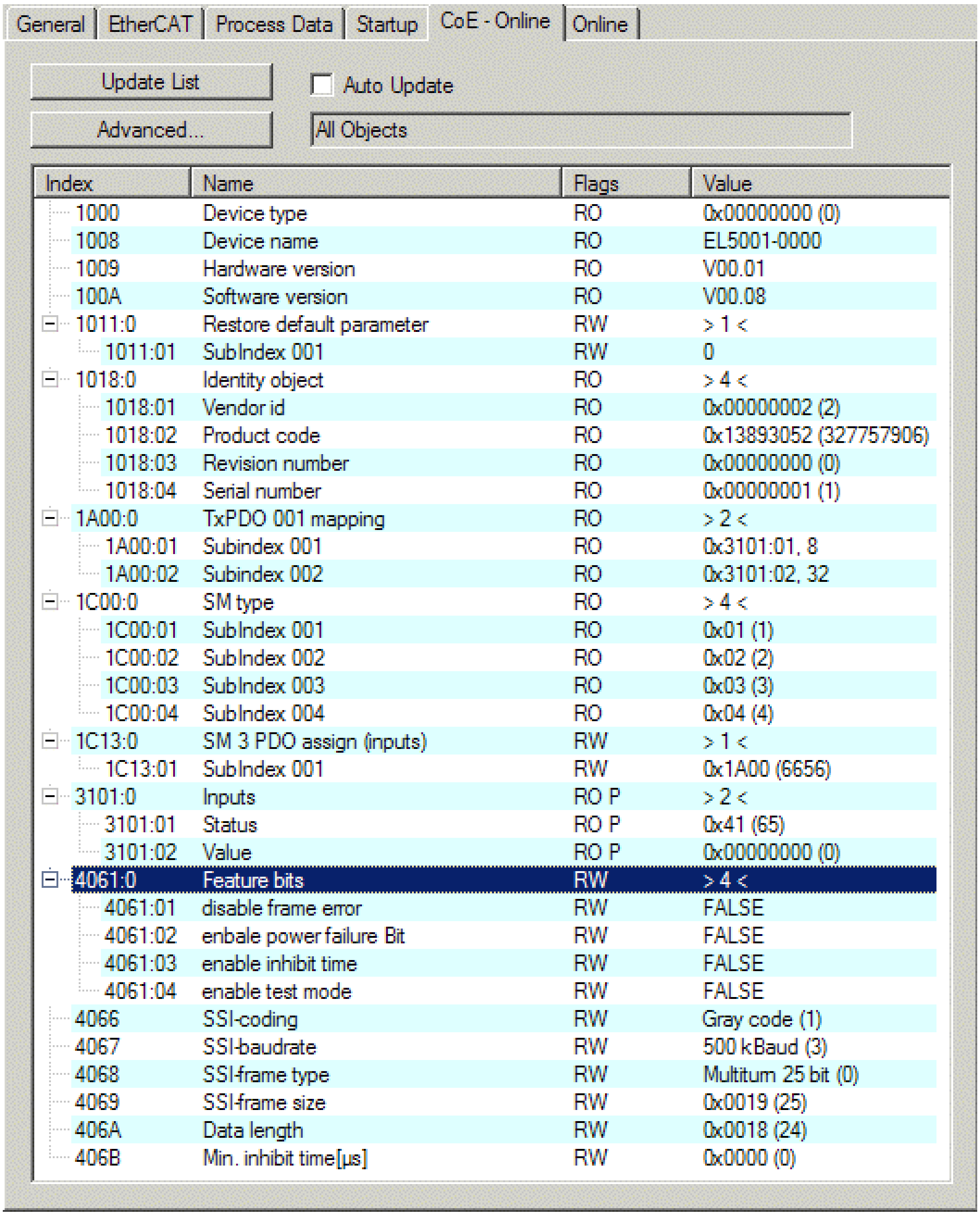
Fig. 124: “Startup” tab

Column	Description
Transition	Transition to which the request is sent. This can either be • the transition from pre-operational to safe-operational (PS), or • the transition from safe-operational to operational (SO). If the transition is enclosed in “<>” (e.g. <PS>), the mailbox request is fixed and cannot be modified or deleted by the user.
Protocol	Type of mailbox protocol
Index	Index of the object
Data	Date on which this object is to be downloaded.
Comment	Description of the request to be sent to the mailbox

Move Up	This button moves the selected request up by one position in the list.
Move Down	This button moves the selected request down by one position in the list.
New	This button adds a new mailbox download request to be sent during startup.
Delete	This button deletes the selected entry.
Edit	This button edits an existing request.

“CoE - Online” tab

The additional *CoE - Online* tab is displayed if the EtherCAT slave supports the *CANopen over EtherCAT* (CoE) protocol. This dialog lists the content of the object list of the slave (SDO upload) and enables the user to modify the content of an object from this list. Details for the objects of the individual EtherCAT devices can be found in the device-specific object descriptions.



□ Auto Update

All Objects

Index	Name	Flags	Value
1000	Device type	RO	0x00000000 (0)
1008	Device name	RO	EL5001-0000
1009	Hardware version	RO	V00.01
100A	Software version	RO	V00.08
1011:0	Restore default parameter	RW	> 1 <
1011:01	SubIndex 001	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	enbale 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	Multitum 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. 125: “CoE - Online” tab

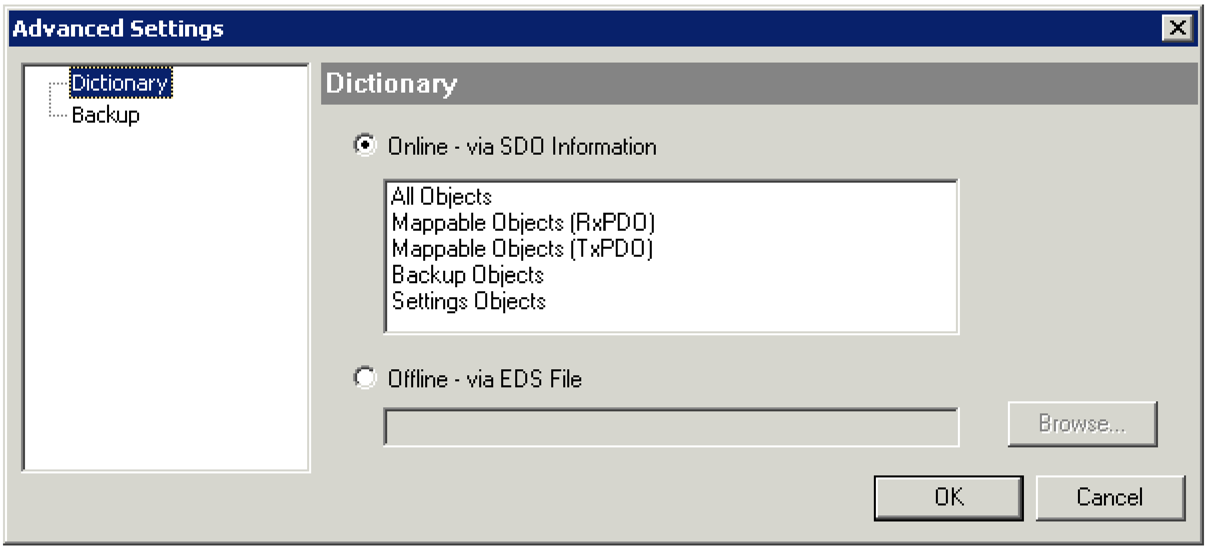
Object list display

Column	Description	
Index	Index and sub-index of the object	
Name	Name of the object	
Flags	RW	The object can be read, and data can be written to the object (read/write)
	RO	The object can be read, but no data can be written to the object (read only)
	P	An additional P identifies the object as a process data object.
Value	Value of the object	

Update List The *Update list* button updates all objects in the displayed list

Auto Update If this check box is selected, the content of the objects is updated automatically.

Advanced The *Advanced* button opens the *Advanced Settings* dialog. Here you can specify which objects are displayed in the list.



● Online - via SDO Information

All Objects
Mappable Objects (RxPDO)
Mappable Objects (TxPDO)
Backup Objects
Settings Objects

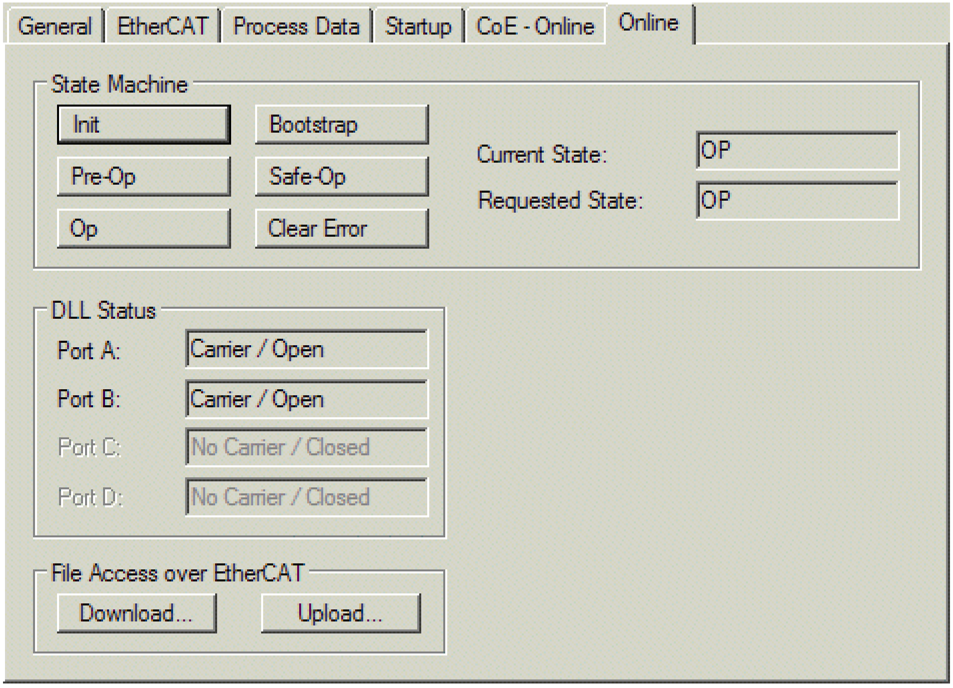
○ Offline - via EDS File

Fig. 126: 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



Current State: OP

Requested State: OP

Port A: Carrier / Open

Port B: Carrier / Open

Port C: No Carrier / Closed

Port D: No Carrier / Closed

Fig. 127: “Online” tab

State Machine

Init This button attempts to set the EtherCAT device to the *Init* state.

Pre-Op This button attempts to set the EtherCAT device to the *pre-operational* state.

Op This button attempts to set the EtherCAT device to the *operational* state.

Bootstrap This button attempts to set the EtherCAT device to the *Bootstrap* state.

Safe-Op This button attempts to set the EtherCAT device to the *safe-operational* state.

Clear Error This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag.

Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. When the *Clear Error* button is pressed the error flag is cleared, and the current state is displayed as PREOP again.

Current State Indicates the current state of the EtherCAT device.

Requested State Indicates the state requested for the EtherCAT device.

DLL Status

Indicates the DLL status (data link layer status) of the individual ports of the EtherCAT slave. The DLL status can have four different states:

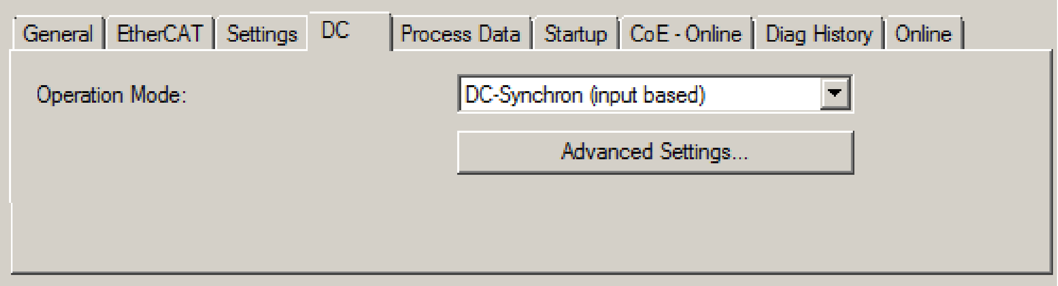
Status	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal is available at the port, but the port is closed.

File Access over EtherCAT

Download With this button a file can be written to the EtherCAT device.

Upload With this button a file can be read from the EtherCAT device.

“DC” tab (Distributed Clocks)



Operation Mode: DC-Synchron (input based)

Fig. 128: “DC” tab (Distributed Clocks)

Operation Mode Options (optional):

- FreeRun
- SM-Synchron
- DC-Synchron (Input based)
- DC-Synchron

Advanced Settings… Advanced settings for readjustment of the real time determinant TwinCAT-clock

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

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

5.2.7.1 Detailed description of Process Data tab

Sync Manager

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

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

PDO Assignment

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

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

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

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 [▶ 111]),

b) and the System Manager has to reload the EtherCAT slaves

(button for TwinCAT 2 or button for TwinCAT 3)

PDO list

List of all PDOs supported by this EtherCAT device. The content of the selected PDOs is displayed in the *PDO Content* list. The PDO configuration can be modified by double-clicking on an entry.

Column	Description	
Index	PDO index.	
Size	Size of the PDO in bytes.	
Name	Name of the PDO. If this PDO is assigned to a Sync Manager, it appears as a variable of the slave with this parameter as the name.	
Flags	F	Fixed content: The content of this PDO is fixed and cannot be changed by the System Manager.
	M	Mandatory PDO. This PDO is mandatory and must therefore be assigned to a Sync Manager! Consequently, this PDO cannot be deleted from the *PDO Assignment* list
SM	Sync Manager to which this PDO is assigned. If this entry is empty, this PDO does not take part in the process data traffic.	
SU	Sync unit to which this PDO is assigned.	

PDO Content

Indicates the content of the PDO. If flag F (fixed content) of the PDO is not set the content can be modified.

Download

If the device is intelligent and has a mailbox, the configuration of the PDO and the PDO assignments can be downloaded to the device. This is an optional feature that is not supported by all EtherCAT slaves.

PDO Assignment

If this check box is selected, the PDO assignment that is configured in the PDO Assignment list is downloaded to the device on startup. The required commands to be sent to the device can be viewed in the Startup [▸ 108] 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 Commissioning Instructions for an EtherCAT Slave

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

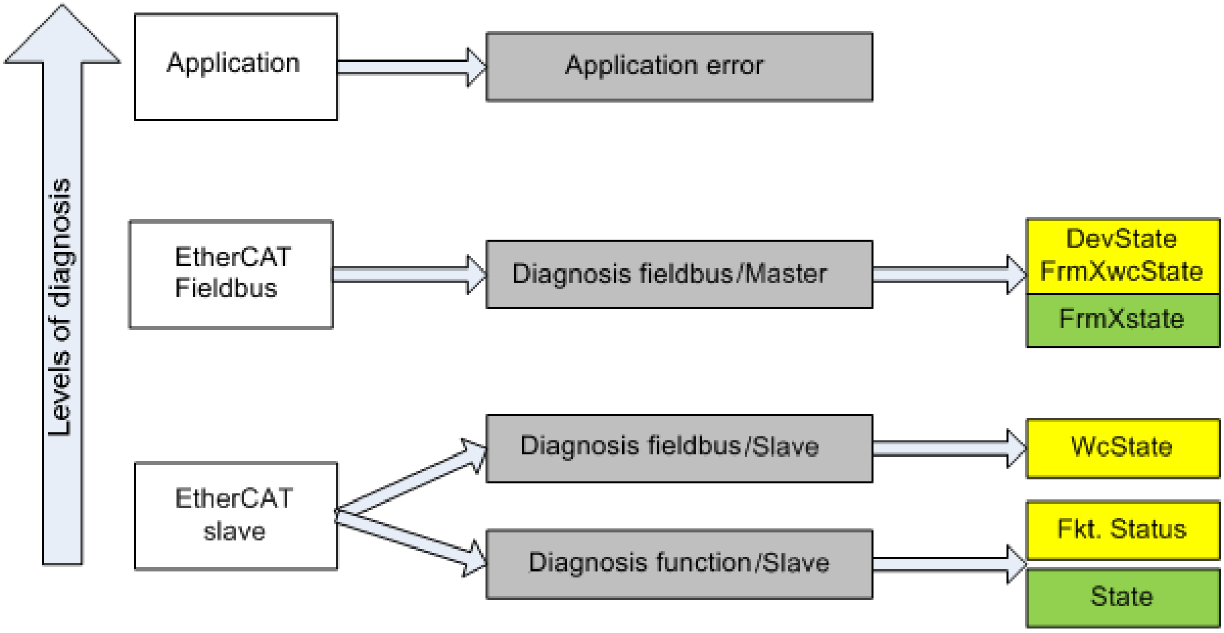


Fig. 129: Selection of the diagnostic information of an EtherCAT Slave

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.

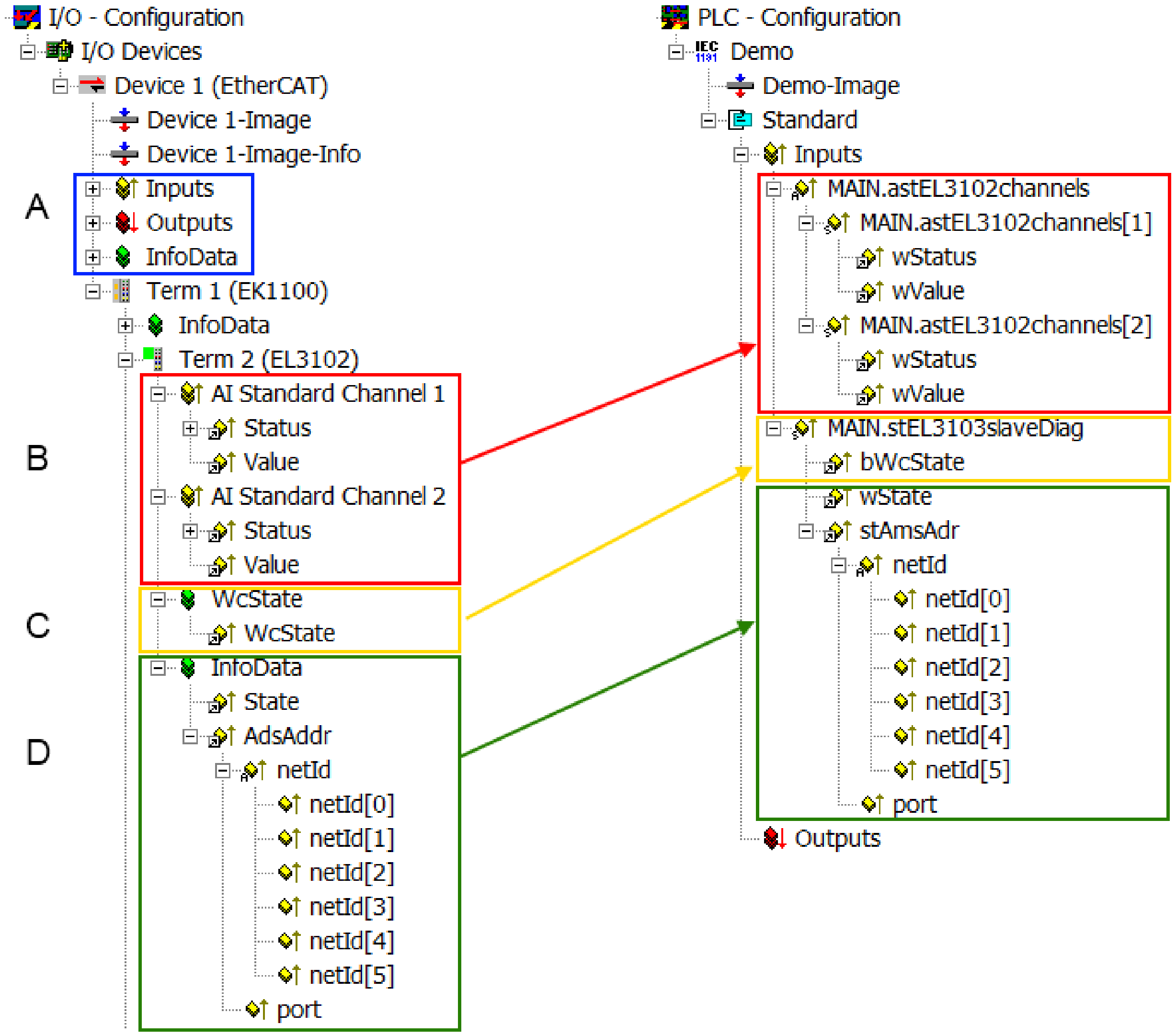


Fig. 130: Basic EtherCAT Slave Diagnosis in the PLC

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

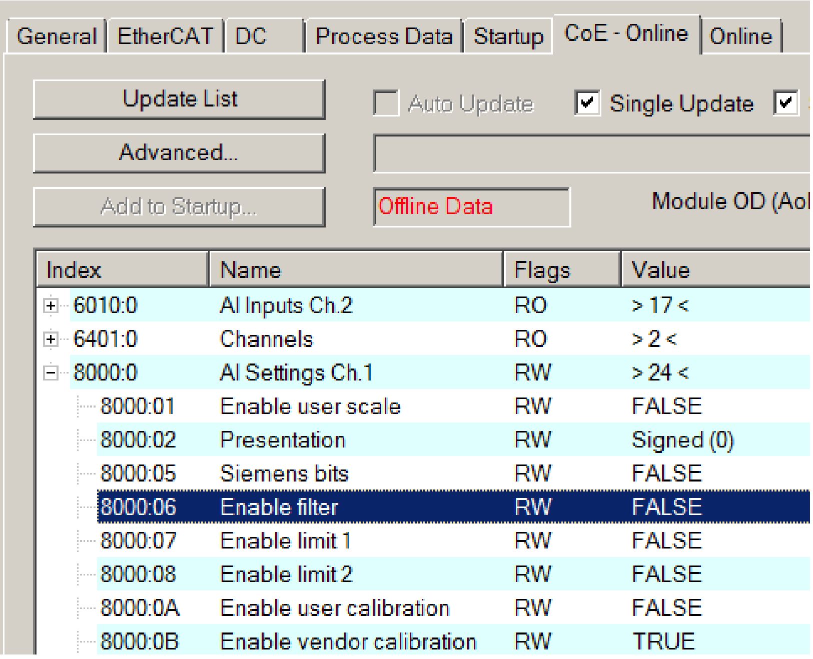
NOTICE

Diagnostic information

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

CoE Parameter Directory

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



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

Fig. 131: EL3102, CoE directory

EtherCAT System Documentation

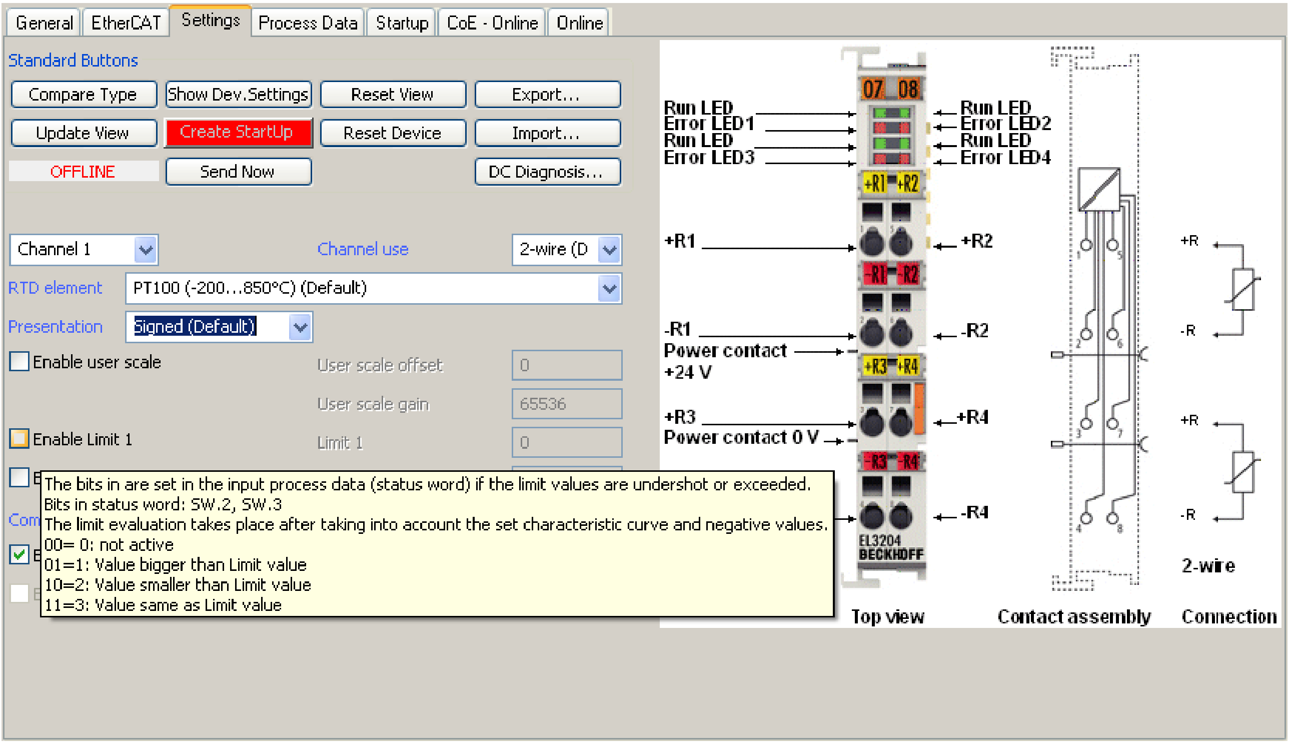
The comprehensive description in the EtherCAT System Documentation (EtherCAT Basics --> CoE Interface) must be observed!

A few brief extracts:

- Whether changes in the online directory are saved locally in the slave depends on the device. EL terminals (except the EL66xx) are able to save in this way.
- The user must manage the changes to the StartUp list.

Commissioning aid in the TwinCAT System Manager

Commissioning interfaces are being introduced as part of an ongoing process for EL/EP EtherCAT devices. These are available in TwinCAT System Managers from TwinCAT 2.11R2 and above. They are integrated into the System Manager through appropriately extended ESI configuration files.



Channel 1 | Channel use: 2-wire (D

RTD element: PT100 (-200...850°C) (Default)

Presentation: Signed (Default)

□ Enable user scale | User scale offset: 0

User scale gain: 65536

□ Enable Limit 1 | Limit 1: 0

The bits in are set in the input process data (status word) if the limit values are undershot or exceeded.
Bits in status word: SW.2, SW.3
The limit evaluation takes place after taking into account the set characteristic curve and negative values.
00= 0: not active
01=1: Value bigger than Limit value
10=2: Value smaller than Limit value
11=3: Value same as Limit value

Fig. 132: 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 [▶ 26]" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

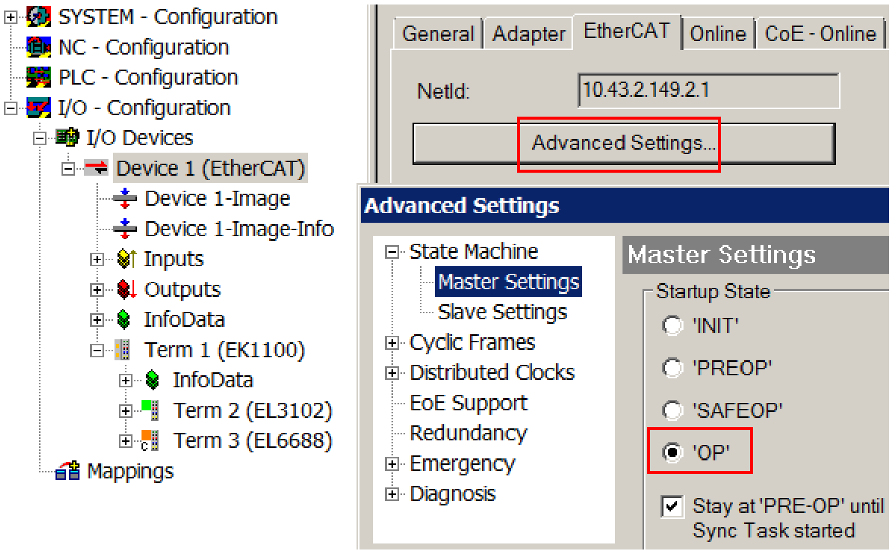
The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

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.



SYSTEM - Configuration
NC - Configuration
PLC - Configuration
I/O - Configuration
I/O Devices
Device 1 (EtherCAT)
Device 1-Image
Device 1-Image-Info
Inputs
Outputs
InfoData
Term 1 (EK1100)
InfoData
Term 2 (EL3102)
Term 3 (EL6688)
Mappings

General | Adapter | EtherCAT | Online | CoE - Online

NetId: 10.43.2.149.2.1

Advanced Settings...

Advanced Settings

State Machine
Master Settings
Slave Settings
Cyclic Frames
Distributed Clocks
EoE Support
Redundancy
Emergency
Diagnosis

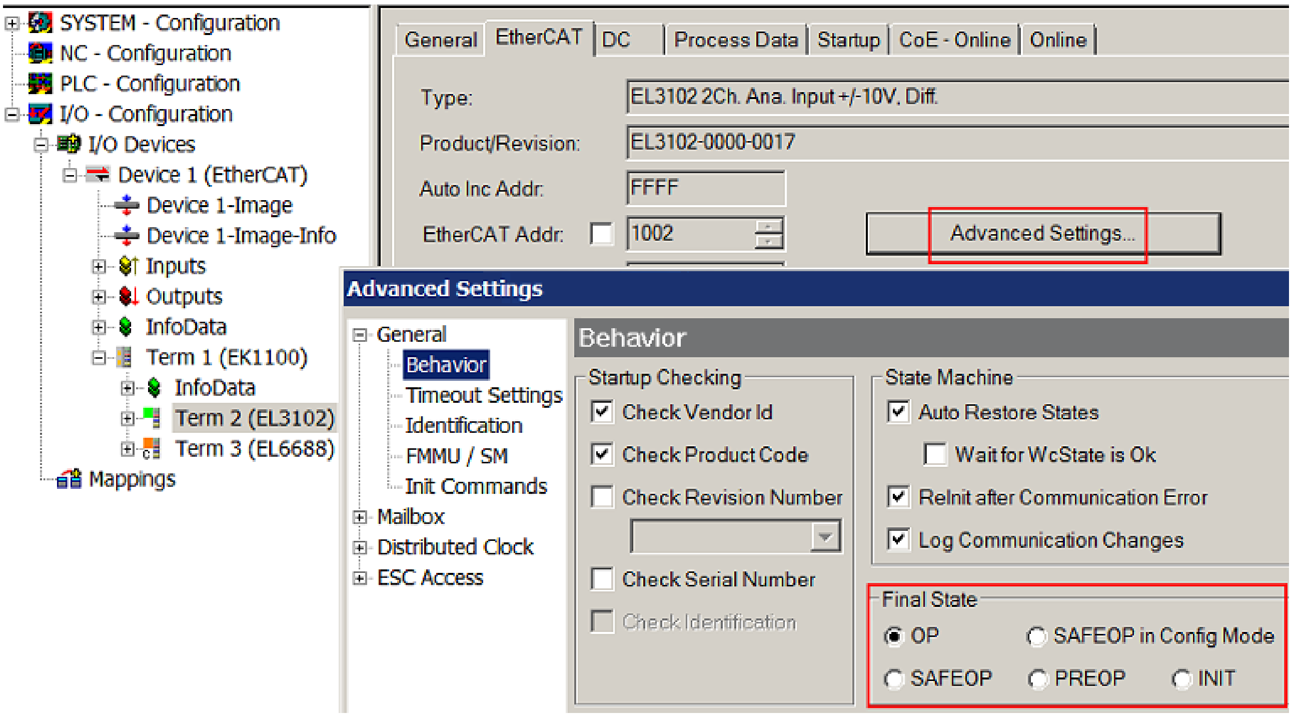
Master Settings

Startup State
○ 'INIT'
○ 'PREOP'
○ 'SAFEOP'
● 'OP'

☑ Stay at 'PRE-OP' until Sync Task started

Fig. 133: 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.



SYSTEM - Configuration
NC - Configuration
PLC - Configuration
I/O - Configuration
I/O Devices
Device 1 (EtherCAT)
Device 1-Image
Device 1-Image-Info
Inputs
Outputs
InfoData
Term 1 (EK1100)
InfoData
Term 2 (EL3102)
Term 3 (EL6688)
Mappings

General | EtherCAT | DC | Process Data | Startup | CoE - Online | Online

Type: EL3102 2Ch. Ana. Input +/-10V, Diff.
Product/Revision: EL3102-0000-0017
Auto Inc Addr: FFFF
EtherCAT Addr: □ 1002

Advanced Settings...

Advanced Settings

General
Behavior
Timeout Settings
Identification
FMMU / SM
Init Commands
Mailbox
Distributed Clock
ESC Access

Behavior

Startup Checking
☑ Check Vendor Id
☑ Check Product Code
□ Check Revision Number
□ Check Serial Number
□ Check Identification

State Machine
☑ Auto Restore States
□ Wait for WcState is Ok
☑ ReInit after Communication Error
☑ Log Communication Changes

Final State
● OP ○ SAFEOP in Config Mode
○ SAFEOP ○ PREOP ○ INIT

Fig. 134: Default target state in the Slave

Manual Control

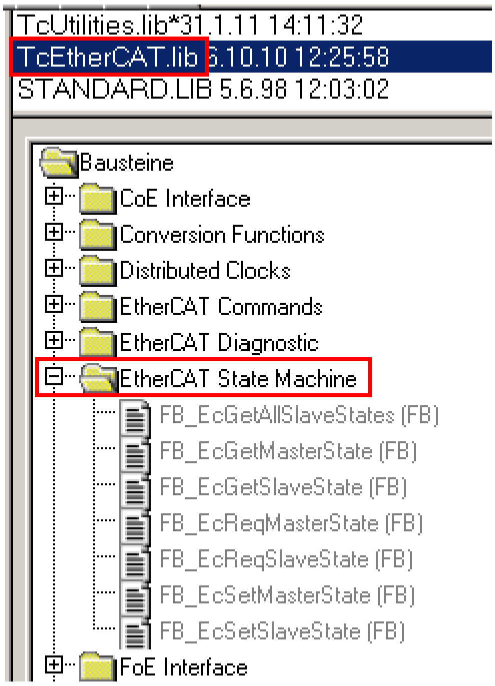
There are particular reasons why it may be appropriate to control the states from the application/task/PLC. For instance:

- 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.



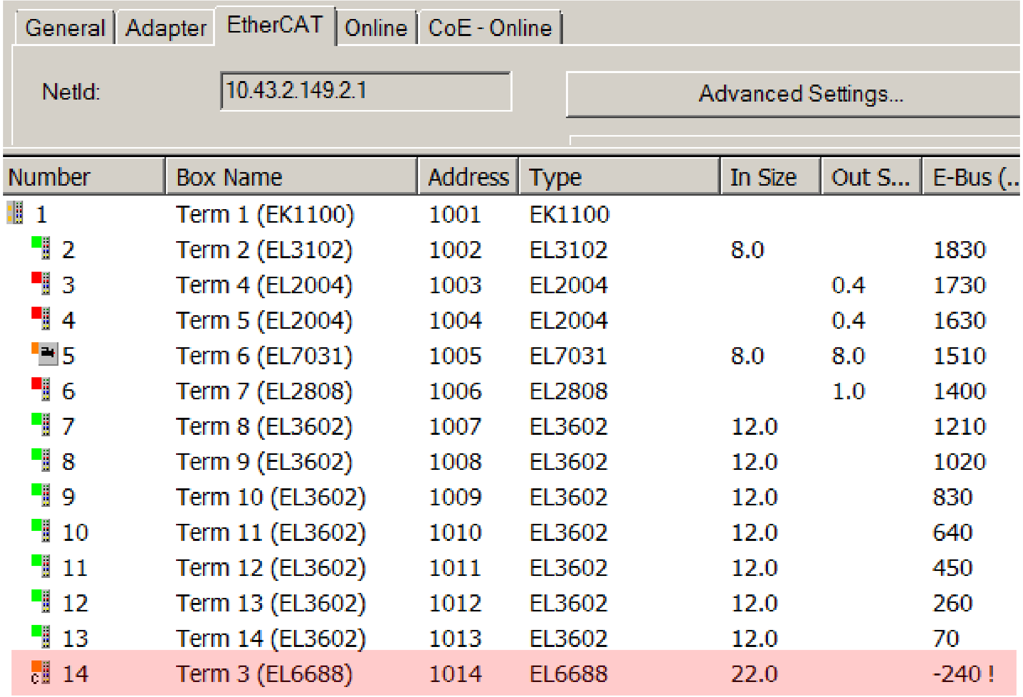
```
TcUtilities.lib*31.1.11 14:11:32
TcEtherCAT.lib 6.10.10 12:25:58
STANDARD.LIB 5.6.98 12:03:02

Bausteine
  CoE Interface
  Conversion Functions
  Distributed Clocks
  EtherCAT Commands
  EtherCAT Diagnostic
  EtherCAT State Machine
    FB_EcGetAllSlaveStates (FB)
    FB_EcGetMasterState (FB)
    FB_EcGetSlaveState (FB)
    FB_EcReqMasterState (FB)
    FB_EcReqSlaveState (FB)
    FB_EcSetMasterState (FB)
    FB_EcSetSlaveState (FB)
  FoE Interface
```

Fig. 135: 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.

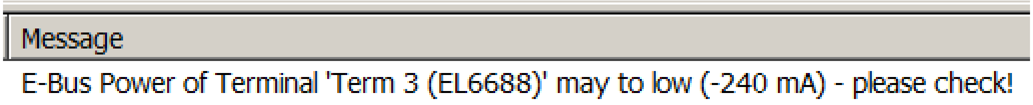


NetId: 10.43.2.149.2.1

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. 136: 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:



Message
E-Bus Power of Terminal 'Term 3 (EL6688)' may to low (-240 mA) - please check!

Fig. 137: Warning message for exceeding E-Bus current

NOTICE

Caution! Malfunction possible!

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

5.4 Notices on analog specifications

Beckhoff I/O devices (terminals, box modules, modules) with analog inputs and outputs 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.

Unless otherwise stated, the explanations apply accordingly to input and output signals.

5.4.1 Full scale value (FSV), output end value

An I/O device with analog input measures over a nominal measuring range, which is limited by an upper and a lower limit (start value and end value), which can usually already be taken from the device designation. The range between both limits is called measuring span and corresponds to the formula (end value - start 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 full scale value (FSV) of the respective product (also: reference value) is selected as the largest limit of the nominal measuring range and is given a positive sign. This applies to both symmetrical and asymmetrical measuring spans.

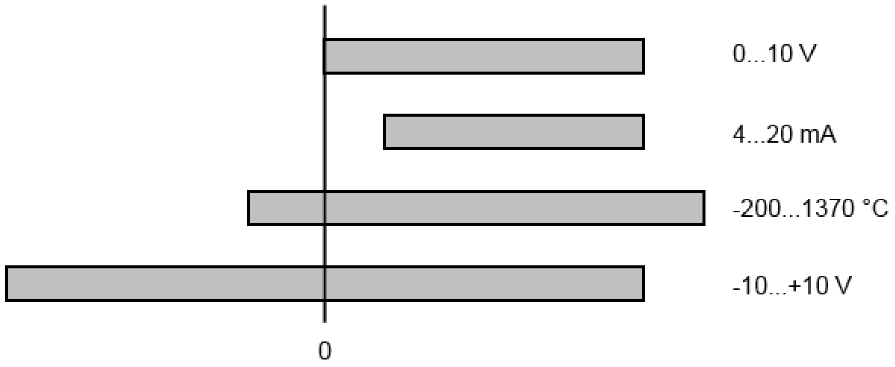


Fig. 138: Full scale value, measuring span

For the above **examples** this means:

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

Depending on the functionality, an analog input channel may have a technical measuring range that exceeds the nominal measuring range, e.g. to gain more diagnostic information about the signal.
The case-by-case information in the device documentation on the behavior outside the nominal measuring range (measurement uncertainty, display value) must be observed.

The above thoughts are correspondingly applicable to analog output devices:

- The full scale value (FSV) becomes the output end value
- Here, too, there can be a (larger) technical output range in addition to the nominal output range

5.4.2 Measurement error/measurement deviation/measurement uncertainty, output uncertainty

Analog output

The following information also applies analogously to the output end value of analog output devices.

The relative measuring error as a specification value of a Beckhoff analog device is specified in % of the nominal FSV (output end value) and calculated as the quotient of the numerically largest probable deviation from the true measured value (output value) with respect to the FSV (output end value):

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

It should be noted here that the "true measured value" cannot be determined with infinite accuracy either, but can only be determined via reference devices with a higher expenditure of technology and measuring time and thus a significantly lower measurement uncertainty.

The value therefore describes the result window in which the measured value determined by the device under consideration (Beckhoff analog device) lies with a very high probability in relation to the "true value". Thus, colloquially, this is a "typical" value (typ.); this expresses that the vast statistical majority of values will be within the specification window, but in rare cases there may/will be deviations outside the window.

For this reason, the term "measurement uncertainty" has become established for this window, since "error" is now used to refer to known disturbance effects that can generally be systematically eliminated.

The uncertainty of measurement must always be considered in relation to potential environmental influences:

- invariable electrical channel properties such as temperature sensitivity,
- variable settings of the channel (noise via filters, sampling rate, ...).

Measurement uncertainty specifications without further operational limitation (also called "service error limit") can be assumed as a value "over everything": entire permissible operating temperature range, default setting, etc.

The window is always to be understood as a positive/negative span with "±", even if occasionally indicated as a "half" window without "±".

The maximum deviation can also be specified directly.

Example: measuring range 0...10 V (FSV = 10 V) and measurement uncertainty < ± 0.3%$_{FSV}$ → the expected maximum usual deviation is ± 30 mV in the permissible operating temperature range.

Lower measurement uncertainty possible

If 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.

5.4.3 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. The basic accuracy is usually specified for 23 °C ambient temperature, in special cases also at other temperature.

Due to the extensive uncertainty considerations that are incorporated in the determination of the basic accuracy, Beckhoff recommends a quadratic summation.

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

$$G35 = \sqrt{(0.01\%)^2 + (12K \cdot 20\,\frac{ppm}{K})^2} = 0.026\%\text{ full scale value, typ.}$$

Remarks: ppm ≙ 10^{-6} % ≙ 10^{-2}

5.4.4 Long-term use

Analog devices (inputs, outputs) are subject to constant environmental influences during operation (temperature, temperature change, shock/vibration, irradiation, etc.) This can affect the function, in particular the analog accuracy (also: measurement or output uncertainty).

As industrial products, Beckhoff analog devices are designed for 24h/7d continuous operation.

The devices show that they generally comply with the accuracy specification, even in long-term use. However, as is usual for technical devices, an unlimited functional assurance (also applies to accuracy) cannot be given.

Beckhoff recommends checking the usability in relation to the application target within the scope of normal system maintenance, e.g. every 12-24 months.

5.4.5 Ground reference: 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.

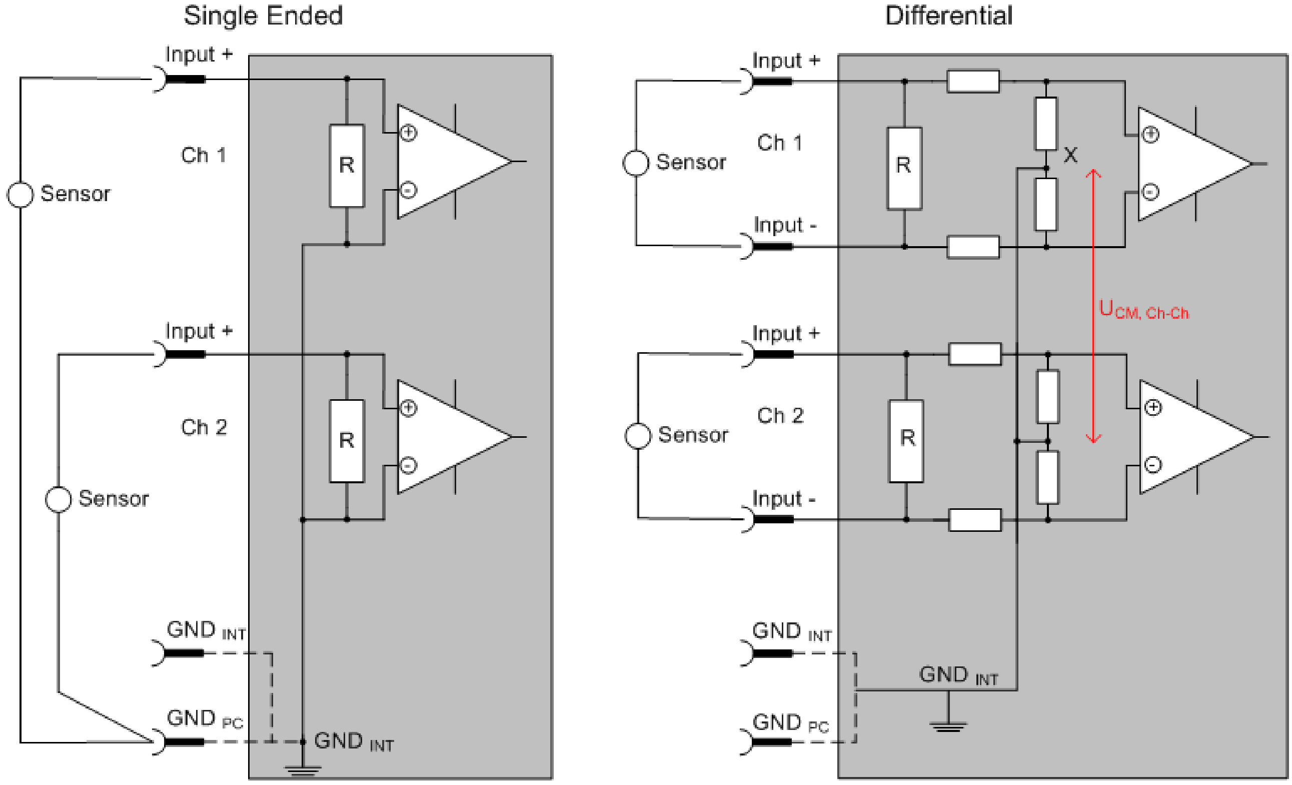


Fig. 139: SE and DIFF module as 2-channel version

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 suggests that only one wire is required.
- The term “electrical isolation” should be clarified in advance.
  Beckhoff I/O devices feature 1 to 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 I/O devices always feature electrical isolation between the field/analog side and the bus/EtherCAT side. In other words, if two analog I/O devices are not connected via the power contacts (cable), they are effectively electrically isolated.
- If channels within a device are electrically isolated, or if a single-channel device 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 device 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 device, in order to stabilize a defined GND potential in the device. 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 of the device 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 device (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 device 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’. See also the data of the sensor manufacturer.

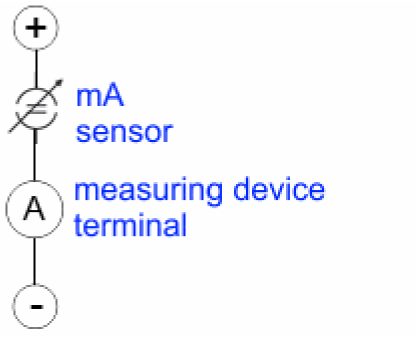


Fig. 140: 2-wire connection

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

preferably to **‘single-ended’ inputs** if the +Supply connections of the device 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 two 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/ box modules (and related product groups), in this case the polarity/direction of current have to be observed.

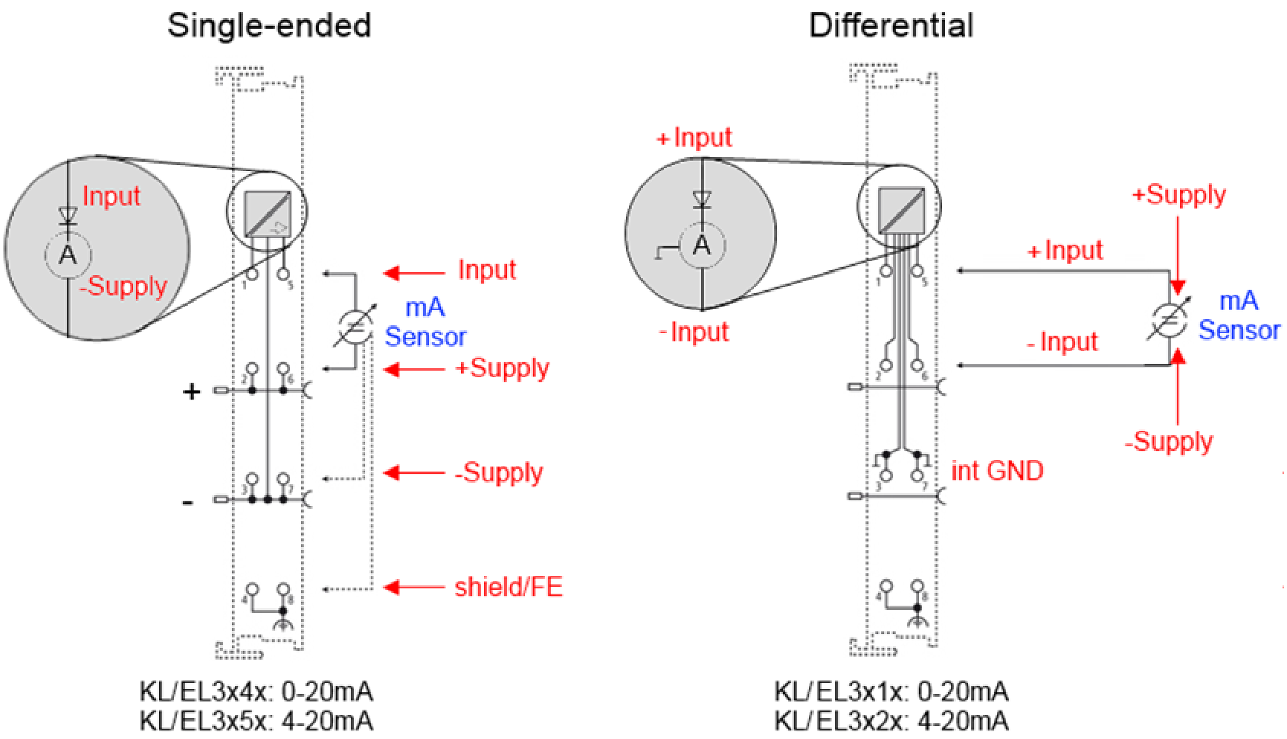


Fig. 141: Connection of externally supplied sensors

Classification of Beckhoff Terminals/ Box modules - Beckhoff 0/4-20 mA Terminals/ Box modules (and related product groups) are available as **differential** and **single-ended**:

Single-ended	**Differential**
EL3x4x: 0-20 mA, EL3x5x: 4-20 mA, same as KL and related product groups	EL3x1x: 0-20 mA, EL3x2x: 4-20 mA, same as KL and related product groups
Preferred current direction because of internal diode	Preferred current direction because of internal diode
Designed for the connection of externally-supplied sensors with a 3/4-wire connection.	The terminal/box is a passive differential current measuring device; passive means that the sensor is not supplied with power.
Designed for the connection of self-supplied sensors with a 2-wire connection	

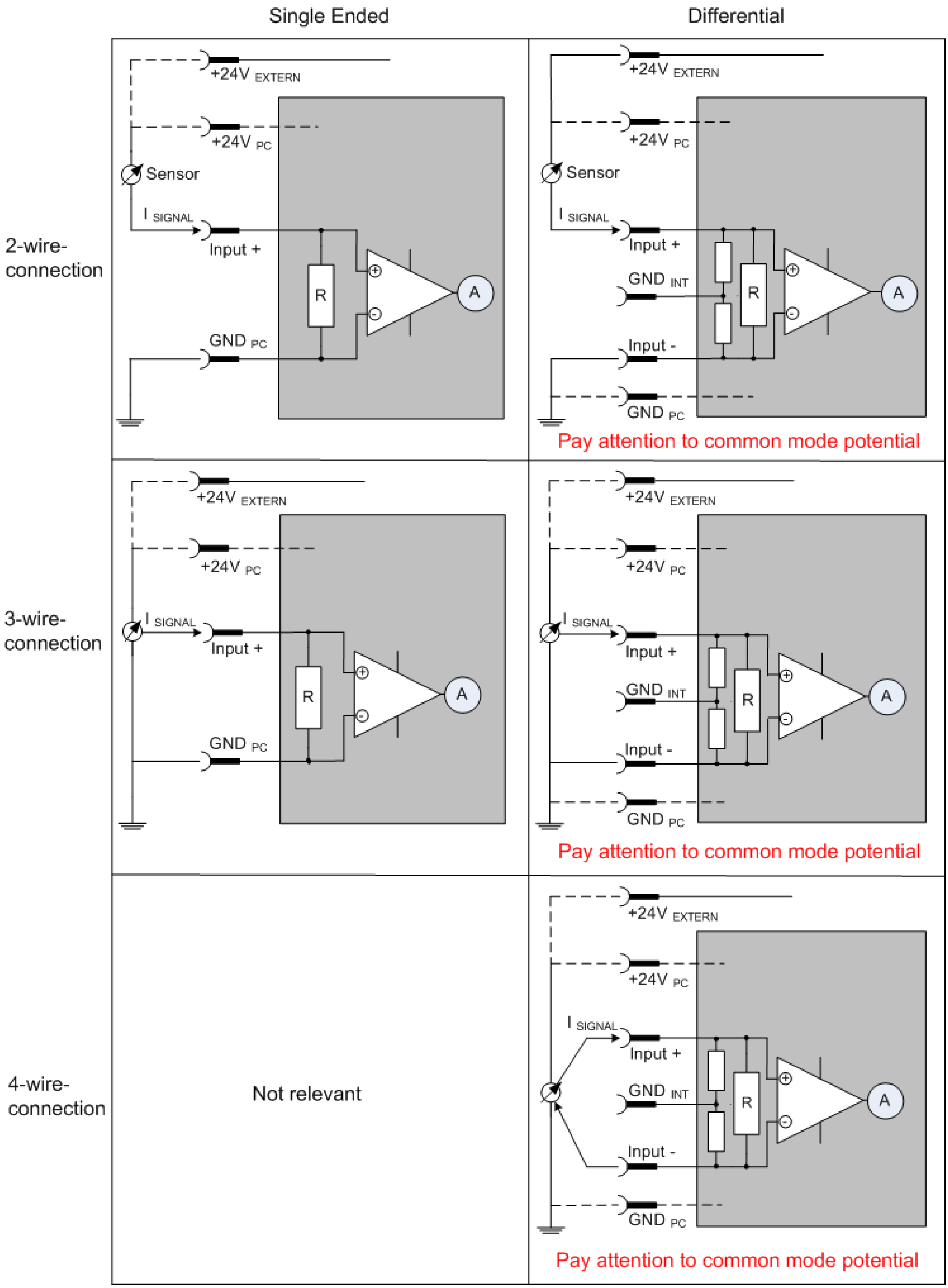


Fig. 142: 2-, 3- and 4-wire connection at single-ended and differential inputs

5.4.6 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.

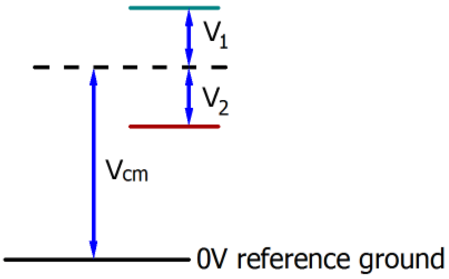


Fig. 143: Common-mode voltage (V_{cm})

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 I/O device. Locations for this can be terminal contacts, power contacts/power line or just a mounting rail.

Please refer to the documentation regarding positioning. The reference ground should be specified for the device under consideration.

For multi-channel I/O devices 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. 0 V power contact: EL3104/EL3114, resistive connection between the channels and AGND; AGND connected to 0 V 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

5.4.7 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

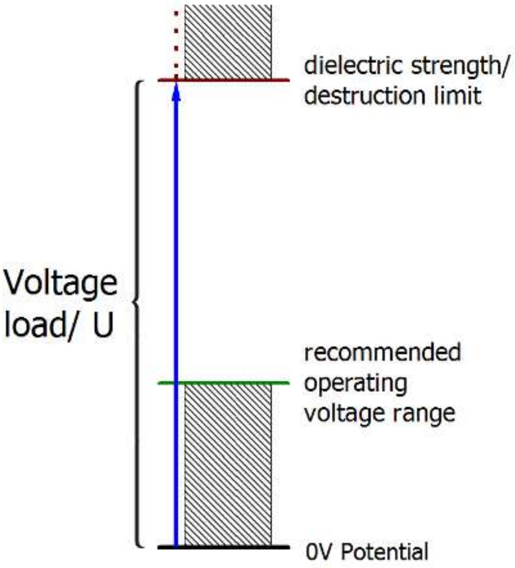


Fig. 144: Recommended operating voltage range

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)

5.4.8 Temporal aspects of analog/digital or digital/analog conversion

Analog output

The following information applies analogously to analog signal output via DAC (digital-to-analog converter).

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 common, from the user's point of view they all have one common feature: after the end of the conversion, a certain digital value is available for further processing in the controller. 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.)

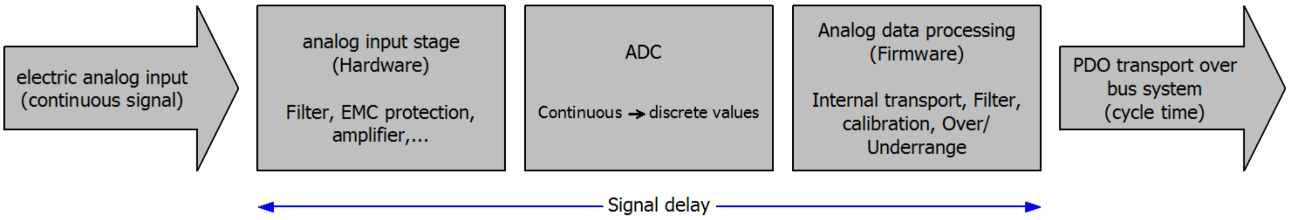

electric analog input (continuous signal)
analog input stage (Hardware)
Filter, EMC protection, amplifier,...
ADC
Continuous → discrete values
Analog data processing (Firmware)
Internal transport, Filter, calibration, Over/ Underrange
PDO transport over bus system (cycle time)
Signal delay

Fig. 145: 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?
  So hardware and firmware parts in toto. For technological reasons, the signal characteristics must be considered to determine this specification: depending on the signal frequency, there may be different propagation times through the system.

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]:
Specifies how often the analog channel provides a newly detected process data value 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 Chap. 7.10.2 2, "Sampling repeat time"

2. Typical signal delay

Corresponds to IEC 61131-2, Chapter 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 timestamp 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 runtime of a signal, a dedicated value can only be specified for a given signal. The value also depends on possibly changing filter settings of the channel.
A typical characterization in the device documentation can be:

2.1 Signal delay (step response)

Keyword 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 when the analog process value has reached the 90% amplitude.

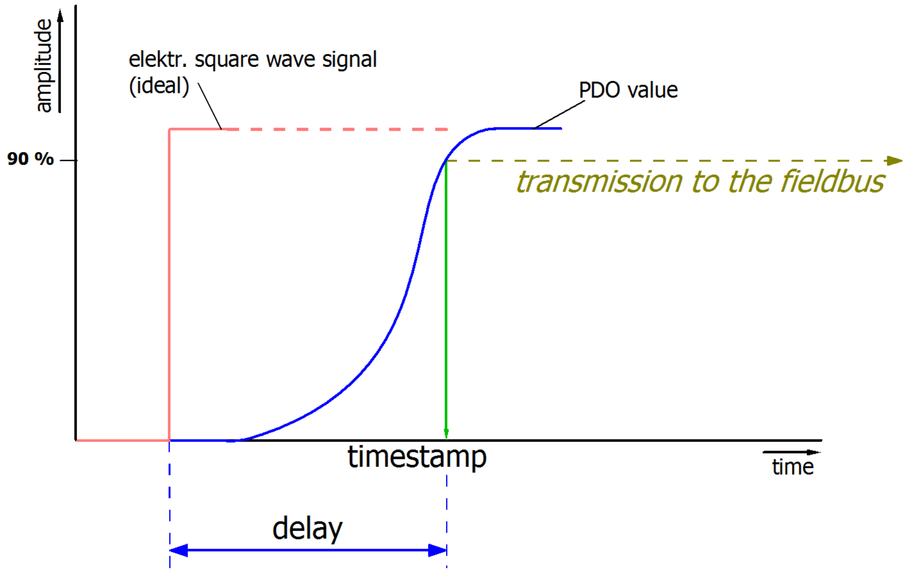


Fig. 146: Diagram Signal delay (step response)

2.2 Signal delay (linear)

Keyword group delay:
Describes the delay of a frequency-constant signal
Test signal can be generated externally with a frequency generator, e.g. as sawtooth or sine. Reference would then be a simultaneous square wave signal.
The signal delay [ms, µs] is then the time interval between the applied electrical signal of a certain amplitude and the moment when the analog process value reaches the same value.
For this purpose, the test frequency must be selected in a reasonable range; this can be, for example, 1/20 of the maximum sampling rate.

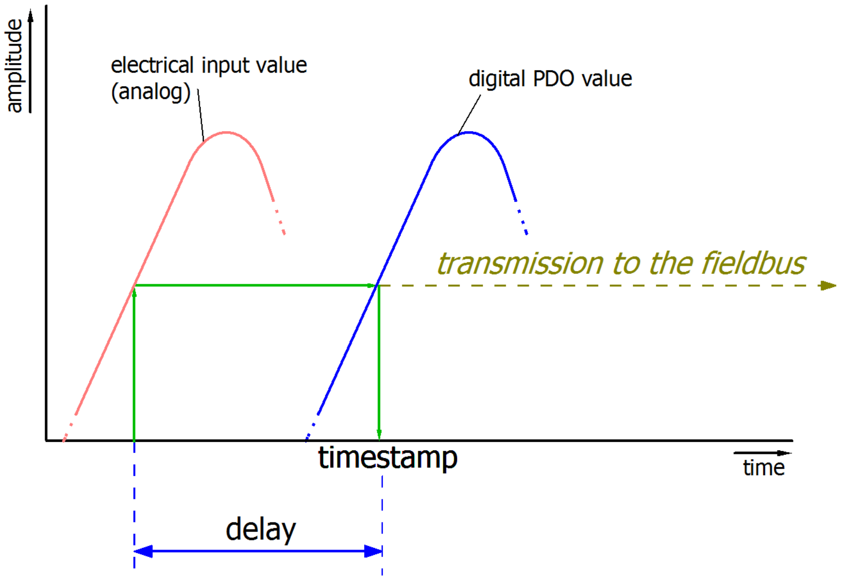


Fig. 147: Diagram Signal delay (linear)

3. Additional information

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 runtimes with different filter settings
- etc.

5.4.9 Explanation of the term GND/Ground

I/O devices always have a reference potential somewhere. After all, the working voltage is only created by two points having different potentials – one of these points is then called the reference potential.

In the Beckhoff I/O area and in particular with the analog products, various reference potentials are used and named. These are defined, named and explained here.

Note: for historical reasons, different names are used with various Beckhoff I/O products. The following explanations place them on a uniform foundation.

SGND

- Also called: FE, Functional Earth, Shield GND, Shield.
- Use: Dissipation of interference and radiation, predominantly currentless.
- Symbol: ⏚.
- Notes and recommendations on SGND/FE can be found in the separate Analog Manual in the chapter "Notes regarding analog equipment - shielding and earth"
- SGND usually ends at the structural earth star point.
- In order to be usable for its intended purpose, SGND itself should be a low noise/noise-free "clean" current and voltage sink.

PE

- Also called: Protective Earth.
- Use: Protective measure to prevent the occurrence of hazardous touch voltages by dissipating these touch voltages and then tripping upstream protective devices. If installed correctly, the PE conductor is currentless, but according to specification it must be capable of conducting for the protection case.
- Symbol: ⏚.
- PE usually ends at the structural earth star point.
- For specifications and notes on PE, please refer to the relevant rules.

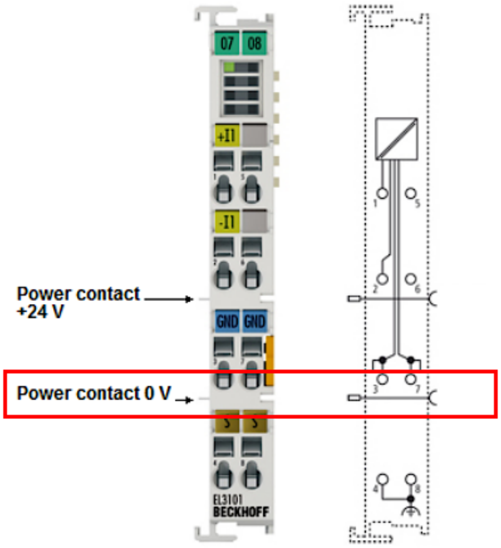
PGND, AGND

- Use: Reference ground or return line of analog or digital signals.
- Depending on use, nominally currentless as reference potential or conducting as return line.
- In the analog area, the so-called standard signals can be 0...10 V and 4...20 mA, measuring bridge signals and thermocouples can be in the range of a few mV and resistance measurement in any Ohm range, and voltages can be from µV to a few thousand Volts.
- In the digital area they can be, for example, 0/24 V, -5/+5 V etc.
- Symbols:

  preferred: ⊥;

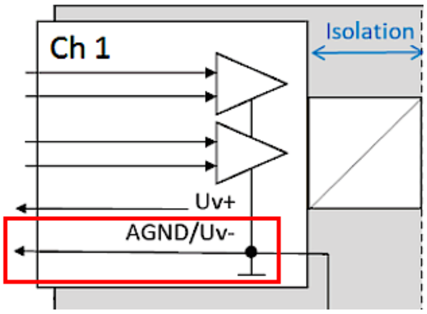
  hardly used any more, but actually means earth potential: ⏚.
- There may be several PGND/AGND networks in a system that are electrically isolated from one another.
- If a device has several AGNDs, due to isolation by channel, these can be numbered: AGND1, AGND2, …
- PGND
  - also called: GND_{PC}, 0 V, power contact 0 V, GND.
  - Version: PGND is a structural description of the "negative" power contact rail of the Bus Terminal system.

  - Can be connected to the device electronics, for example for supplying power to the device or as a signal feedback (see chapter Ground reference: single-ended/differential typification [▶ 126]). Refer to the respective device documentation.
  - Example: PGND is not connected to the device electronics:



.

- AGND
  - Also called: GND$_{int}$, GND, analog ground, GND$_{analog}$.
  - AGND electrically designates the device's analog reference ground.
  - AGND can, for example, be internally connected to PGND or to a connection point so that it can be connected externally to a desired potential. Electrical restrictions according to the device documentation must be observed, e.g. common mode limits.
  - AGND is usually a currentless reference potential. The action of interference on AGND must be avoided.
  - Example: AGND is fed out on the device plug:



.

5.4.10 Sampling type: Simultaneous vs. multiplexed

Analog inputs and outputs in Beckhoff devices can operate in two different ways in terms of time: "simultaneous sampling" or "multiplex sampling". This so-called sampling type has a decisive influence on the performance of such a device and must be taken into consideration when selecting a product, at least when it comes to very complex timing control tasks. Whether an analog device operates simultaneously or multiplexed can be taken from the respective device documentation.

This question is relevant for control tasks as well as for measurement tasks (DataRecording), if the timing of the analog value acquisition is sensitive.

Note: The terms "simultaneous" and "multiplex" have been used for a long time and in many contexts, so they have different meanings depending on the historical background and the subject area. In this chapter and in relation to I/O, the terms are used as Beckhoff understands them as an I/O manufacturer for the benefit of the user:

- If a test signal is applied electrically to all channels of a multi-channel device at the same time and the measurements are evaluated in software, e.g. in TwinCAT Scope, and if no significant offset/delay can be observed between the channels, then it is a **simultaneously sampling device** [1)]
- If an offset can be seen, it is a **multiplex sampling device**
- The easiest **test** to perform is with a square wave signal because an offset can then be easily observed. However, the rare special case could occur (especially if the test signal is generated from an EL2xxx/EL4xxx from the same IO line) that the square wave signal runs synchronously to the EtherCAT for several minutes and then no offset can be seen.

Absolutely safe is a test with a sinusoidal signal, but then it must be considered that measurement deviations (related to the amplitude) of the channels in the device are also represented as time offset!

Ideally, one should concentrate on the zero crossing.

- 1-channel devices are considered as simultaneous sampling by definition.

Explanation with the example "analog input": if a continuous analog signal is to be digitized and thus fed to the further programmatic processing, it is digitized by a so-called ADC (AnalogDigitalConverter), e.g. with 16 bit resolution:

Fig. 148: Schematic representation of sampling with ADC converter

This represents an analog input channel that is functional in itself. It samples (measures) as often as desired, e.g. 1,000 times per second, and thus sends 1,000 measured values equidistant in time (= at equal time intervals) for further processing.

Often several channels are combined in one device, in this case the question arises about the sampling type: simultaneous or multiplex.

[1]) For experts: such a device could also be equipped with a multiplexing ADC, which works with sample-and-hold on all channels. Then technically multiplex is built in, but from the outside the device works simultaneously, because all channels are electrically read in at the same time.

Simultaneous

As in the 1-channel example, each channel can have its own ADC, shown here for 4 channels:

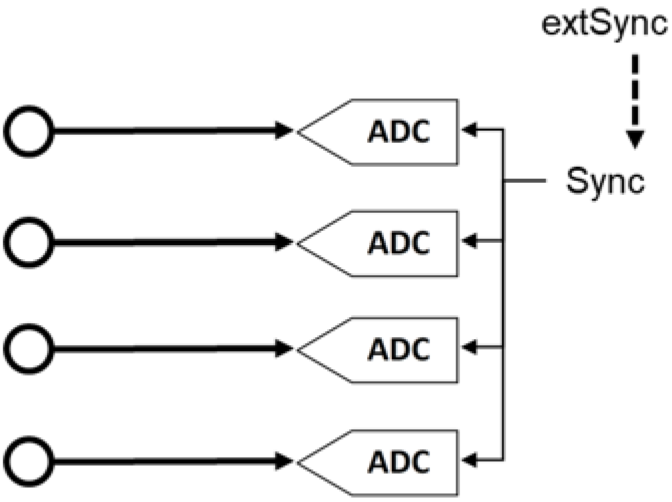


Fig. 149: Schematic representation simultaneous sampling with 4 ADC converters

These ADCs rarely run free in time and sample independently but are normal triggered in some way (the measurement is triggered) to achieve the most desired effect that the n channels sample simultaneously. This gives the analog input device the property that all (4) measurements are obtained at the same time. This gives a temporally consistent view of the machine situation and makes measurement evaluations in the controller very easy. If the ADC are triggered simultaneously by the sync signal, this is called simultaneous sampling.

A special added value arises when such devices are synchronized externally, e.g. via EtherCAT DistributedClocks, and then all analog channels of all devices of a plant operate simultaneously: either really simultaneously without offset among each other or with the same frequency but with constant, known and thus compensatable offset among each other.

As shown above, this requires extensive electronics with multiple identical structures. For this reason, parallel analog devices are usually always simultaneously sampling. Free-running or non-triggered, multiple ADCs are conceivable (and can then no longer be called "simultaneous"), but are rather uncommon.

Multiplex

Simultaneous sampling is often not required for simple automation tasks. This may be because the simplest analog electronics are to be used for cost reasons, or the control cycle time is relatively slow compared to the conversion time in the ADC. Then the advantages of the multiplex concept can be used: Instead of 4 ADC only one ADC is installed, for this a channel switch (from the device manufacturer) must be installed, which switches through the 4 input channels to the ADC quickly one after the other in the µs range. The switching process is performed by the device itself and is usually not accessible from the outside.

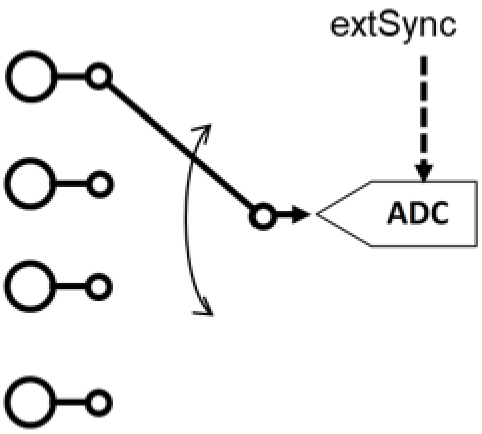


Fig. 150: Schematic representation of multiplex sampling with an ADC converter

This is therefore a time multiplex. As a rule the ADC samples equally clocked, the time intervals of the channels are therefore equal, whereby the start of channel 1 is usually done by the communication cycle (EtherCAT) or DistributedClocks. For further details please refer to the device documentation.

Advantage: cheaper electronics compared to simultaneous setup.

Disadvantage: the measured values are no longer acquired simultaneously, but one after the other.

Both circuits have their technical and economic justification, for time demanding automation tasks simultaneous circuits should always be chosen, because with them it is easier to keep the temporal overview.

For analog outputs the same explanations apply, they can also be equipped with multiple simultaneous DACs or output a multiplexed DAC to several outputs.

5.5 Quick start

Table of contents

EL3692 default setting

The default state for both EL3692 channels is

- CoE mode (parameter settings via CoE)
- alternating measuring mode
- 4-wire mode
- Autorange for the measuring ranges
- Process data: fixpoint and float
- Filter setting 10 Hz (approx. 100 ms conversion time)
- Delay time 10 (200 or 500 ms depending on the measuring range)

Time behavior in Autorange mode

In standard measuring mode the EL3692 keeps switching between the two channels. Depending on the connected measuring object significant charging effects may occur, which may not be completed within the specified measuring time. In Autorange mode, with significant changes in resistance and associated changes in measuring range, this may result in the correct measuring range not being reached, which would impair the measuring accuracy.
By extending the delay before the measurement, sufficient time for charging processes can be made available to ensure that the correct measuring range is reached in Autorange mode. Should the automatic setting be inadequate, the user can change the measuring time in the CoE, 0x80n0:31 [▶ 165].

Step-by-step instructions

Proceed as follows for commissioning with the above-mentioned settings:

1. Install the EL3692 as described in section Installation and wiring [▶ 34].
2. Activate the EtherCAT master and start the terminal in OP state. In the input variables the EL3692 must supply state=OP and WC=0.
3. Connect the resistance in 2-wire or 4-wire mode according to the figure below

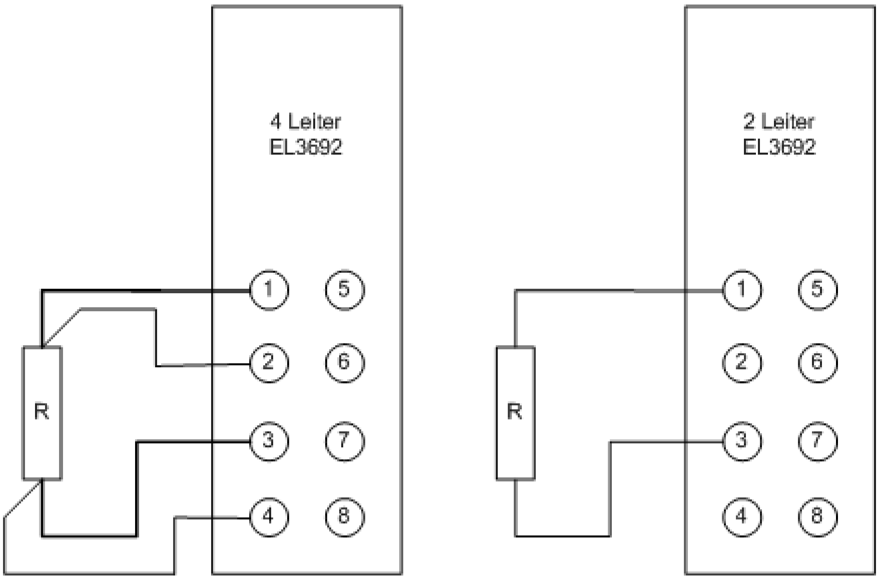
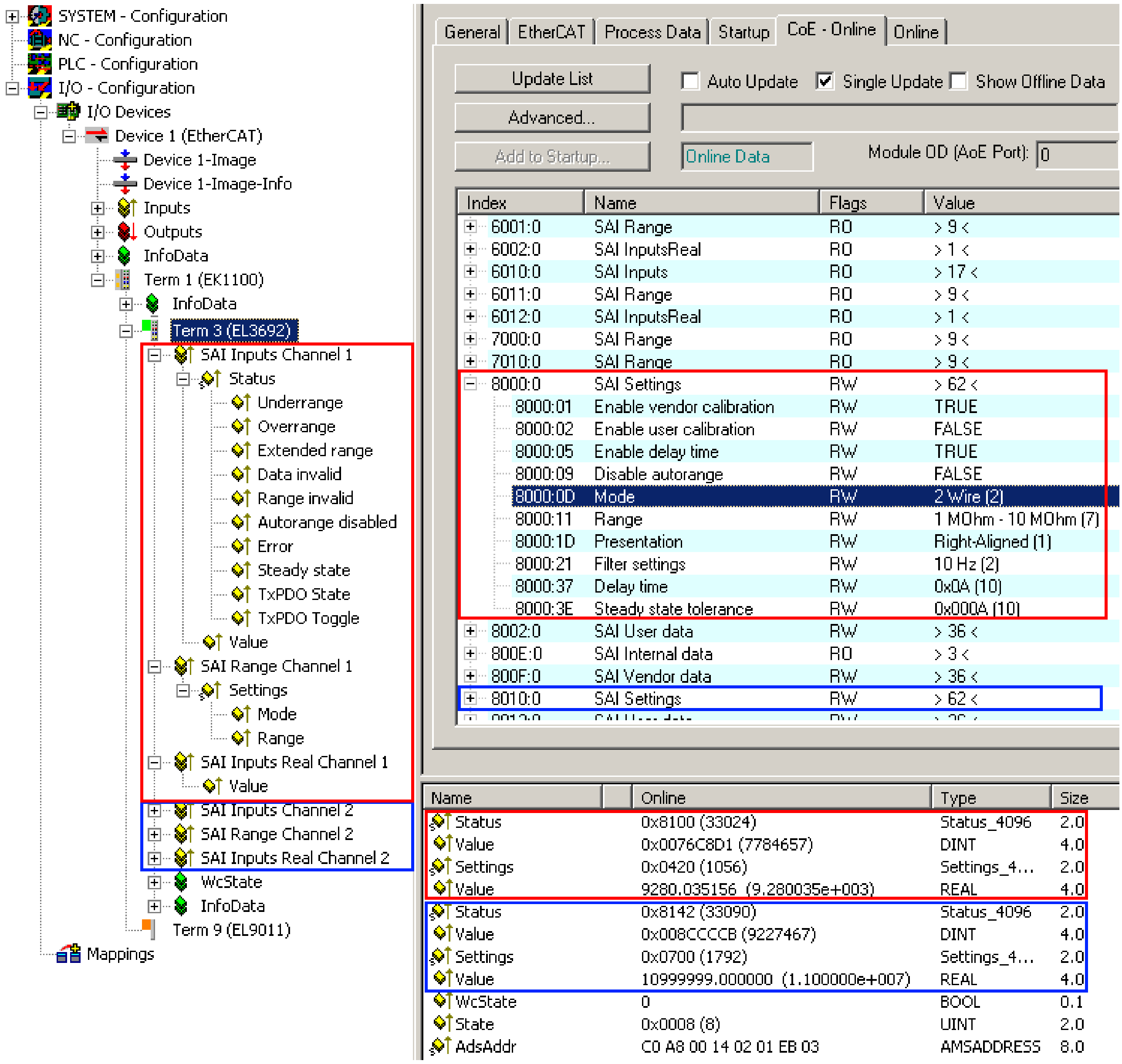


Fig. 151: Connection examples

4. In TwinCAT the EL3692 appears as follows (channel 1 red grouped, channel 2 blue grouped):



```
SYSTEM - Configuration
NC - Configuration
PLC - Configuration
I/O - Configuration
  I/O Devices
    Device 1 (EtherCAT)
      Device 1-Image
      Device 1-Image-Info
      Inputs
      Outputs
      InfoData
      Term 1 (EK1100)
        InfoData
        Term 3 (EL3692)
          SAI Inputs Channel 1
            Status
              Underrange
              Overrange
              Extended range
              Data invalid
              Range invalid
              Autorange disabled
              Error
              Steady state
              TxPDO State
              TxPDO Toggle
            Value
          SAI Range Channel 1
            Settings
              Mode
              Range
          SAI Inputs Real Channel 1
            Value
          SAI Inputs Channel 2
          SAI Range Channel 2
          SAI Inputs Real Channel 2
          WcState
          InfoData
        Term 9 (EL9011)
  Mappings
```

□ Auto Update ☑ Single Update □ Show Offline Data

Online Data

Module OD (AoE Port): 0

Index	Name	Flags	Value
6001:0	SAI Range	RO	> 9 <
6002:0	SAI InputsReal	RO	> 1 <
6010:0	SAI Inputs	RO	> 17 <
6011:0	SAI Range	RO	> 9 <
6012:0	SAI InputsReal	RO	> 1 <
7000:0	SAI Range	RO	> 9 <
7010:0	SAI Range	RO	> 9 <
8000:0	SAI Settings	RW	> 62 <
8000:01	Enable vendor calibration	RW	TRUE
8000:02	Enable user calibration	RW	FALSE
8000:05	Enable delay time	RW	TRUE
8000:09	Disable autorange	RW	FALSE
8000:0D	Mode	RW	2 Wire (2)
8000:11	Range	RW	1 MOhm - 10 MOhm (7)
8000:1D	Presentation	RW	Right-Aligned (1)
8000:21	Filter settings	RW	10 Hz (2)
8000:37	Delay time	RW	0x0A (10)
8000:3E	Steady state tolerance	RW	0x000A (10)
8002:0	SAI User data	RW	> 36 <
800E:0	SAI Internal data	RO	> 3 <
800F:0	SAI Vendor data	RW	> 36 <
8010:0	SAI Settings	RW	> 62 <

Name		Online	Type	Size
Status		0x8100 (33024)	Status_4096	2.0
Value		0x0076C8D1 (7784657)	DINT	4.0
Settings		0x0420 (1056)	Settings_4...	2.0
Value		9280.035156 (9.280035e+003)	REAL	4.0
Status		0x8142 (33090)	Status_4096	2.0
Value		0x008CCCCB (9227467)	DINT	4.0
Settings		0x0700 (1792)	Settings_4...	2.0
Value		10999999.000000 (1.100000e+007)	REAL	4.0
WcState		0	BOOL	0.1
State		0x0008 (8)	UINT	2.0
AdsAddr		C0 A8 00 14 02 01 EB 03	AMSADDRESS	8.0

Fig. 152: EL3692 representation in the TwinCAT System Manager

- The EL3692 provides the following outputs for each channel:
  - *SAI Inputs:* current status
  - *SAI Inputs:* current measured value 32 bit integer, *measuring range-dependent*
  - *SAI Range::* current range (automatic measuring range selection)
  - *SAI Inputs Real:* current measured value 32 bit PLC fixed-point float, *measuring range-independent, scaled representation*

Please note the information relating to the process data [▶ 149].

5. for 2-wire measurement: In CoE entry x80n0:0D set the mode to 2 (n stands for the channel: 0 = channel 1, 1 = channel 2).
If necessary the EL3692 can be reset to the factory settings via a CoE reset [▶ 190].

Measured value display

The EL3692 can output the current channel-based measured value depending on the PDO option "*Predefined PDO assignment*" in 2 different ways. As "**fixpoint**" and "**float**". See following explanation:

- as 32 bit integer, measuring range dependent value, with 24 bit content **"fixpoint"**.
  Explanation:
  The 9 measuring ranges of the EL3692 are named after their recommended range of use e.g. 1 ... 10 Ω, but they all measure down to 0 Ohm. The value range of Value thus extends over x0 ... x00 7F FF FF for each measurement range, with "x0 = 0 ohms" and "x00 7F FF FF ≈ nominal full scale".
  However, the nominal measuring range is exceeded by up to 10% in autorange mode to ensure measuring range switching, so-called "extended measuring range".
  The information in the status must be observed.
  In the following an example for the measuring range "1 ... 10 Ω":

Input resistance		32-bit value	Interpretation	Status bits
0 Ω		x00 00 00 00	0	Extended Range = 0 Overrange = 0
10 Ω	Nominal measuring range end value	x00 7F FF FF	8.388.607	extended range = 1 "extended measuring range" Overrange = 0
10 .. 11 Ω	extended measuring range			Extended Range = 1 Overrange = 0
> 11 Ω		x00 8C CC CB	9.227.467	Extended Range = 0 Overrange = 1

Please note that the relative measurement error is very large at the lower end of the measuring range. When measuring in the range < 10% of the nominal measuring range end value, the next lower measuring range should therefore be used. The autorange function works accordingly.

- as 32-bit fixed-point float-value with mantissa and exponent, measuring range-independent **"float"**
  Explanation:
  This 32-bit variable can be linked directly with a FLOAT variable of the PLC according to IEC61131. The conversion takes place in the EL3692; no measuring range-dependent conversion by the user is required.
  See additional information on page Process data [▸ 149]

6. You have to check the correct operation of the range change function in the autorange opration mode for all test specimens. This is mandatory in the special case, if test specimens with very different resistance/capacity/inductive values are connected to channel 1 and 2 (> 3 measuring ranges). An unoccupied channel will measure in the measuring range 7 (1 MΩ .. 10 MΩ).
This can be compensated channel-wise by the extension of the delay time (CoE 0x80n0:31 [▸ 168]), if the automatic setting is not sufficient. The delay time should be increased stepwise for the channel, which has not the optimal range setting.

7. During the operation, the channel-wise process data value has to be verified by checking of the state variables *Error, DataInvalid* and *SteadyState*.

For further configuration in TwinCAT please refer to sections Process data [▸ 149] and Important CoE entries [▸ 165].

5.6 Operating behavior, diagnostics

The following section describes certain aspects of the EL3692 operating behavior.

5.6.1 Control

The EL3692 is either controlled via the **CoE settings** or via the **control word** in the output process image.

- CoE mode (default setting)
  The settings for Autorange activation, Range (measuring range) and Mode (2/4-wire measurement) are made via the CoE objects. During commissioning they are accessible online via the TwinCAT System Manager. During system runtime these objects can be addressed through ADS via acyclic EtherCAT telegrams.
- Control Word
  When the 16-bit control word is activated in the PDO settings, the EL3692 will only respond to instructions from the control word: Disable Autorange, Start conversion (SingleShot mode), Range and Mode. The respective CoE entries are then ignored.

5.6.2 Basic operation

We recommend operating the EL3692 as follows. Other operating modes referred to in this documentation are, of course, also permitted.

1. CoE mode
2. Autorange enabled
3. Process data
   - Utilization/linking of the measuring range-independent *fixpoint-float* process variable *value*
   - Monitoring of *WcState* (=0) and *State* (=8) in each cycle
   - Monitoring of TxPDO state (=0) and TxPDO toggle (toggling)
   - Status.Error = 0
   - Status.DataInvalid = 0
4. The *value* can be used when *Status.steadyState* = 1

5.6.3 Default state

- Process image with range-dependent and range-independent value
- 4-wire mode
- CoE mode
- AutoRange active

5.6.4 Process data

See page Process data. [▶ 149]

5.6.5 4/2-wire mode

For measuring a resistance via the 9 available measuring ranges on 2 channels in 2- and 4-wire mode the EL3692 has 9*2*2 = 36 manufacturer calibration data, consisting of gain and offset. Accordingly, 36 user calibration entries are available for user gain and user offset.

Calibration in 2-wire mode

For the lower 3 measuring ranges (10mΩ - 10 Ω) 2-wire mode only makes sense if the connection characteristics (line lengths, contact resistances) are known exactly. In these 3 measuring ranges the EL3692 is therefore not calibrated by the manufacturer for 2-wire measurement mode.
In Autorange mode with 2-wire connection these 3 measuring ranges are therefore not used.

The 3 measuring ranges can be used through manual settings (CoE or control word). Previous user calibration by the customer is essential.

The default state of the EL3692 is 4-wire mode. In this mode the measuring current flows via terminal points 1(5) through the resistance to terminal point 3(7). The voltage measurement takes place via terminal points 2(6) and 4(8).

5.6.6 Measuring mode

The terminal supports 4 measuring modes. The measuring mode is selected via the Mode entries in the process data (control word) or in CoE 0x80n0:0D [▶ 166]. The measuring mode determines the time of the measurement and the wire connection.

- 0: 4-wire measurement
- 1: 4-wire measurement (single-shot mode)
- 2: 2-wire measurement
- 3: 2-wire measurement (single-shot mode)

Measurements

Measurements cannot take place simultaneously on both channels. 4-wire connection is recommended for the EL3692.

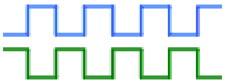
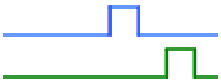
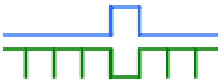
Operating mode	Mode entry	Principle of operation
Alternating mode	Both channels measuring mode "0" or "2"	
Single shot mode	Both channels measuring mode "1" or "3"	
Mixed mode	One channel measuring mode "0" or "2", the other channel mode "1" or "3"	

Table 1: Operating modes

Alternating mode

This mode is the delivery state of the terminal. Both channels alternately measure a resistance at their connection points. The measurement is indicated by the lit status LED.

Single shot mode

Each channel only measures on request by the application. A rising edge in the "Start Conversion" bit in the control word triggers a measurement. If both channels are triggered simultaneously, the channels are processed sequentially. The first channel has priority.

This mode is suitable for measurements during which heating of the test specimen due to repeated measurements is undesirable.

Mixed mode

One channel measures continuously, the other only on request. This mode is a combination of the two previous modes. Once channel measures continuously and interrupts its measurement only if a measurement is requested on the other channel.

This mode is suitable for measuring high-resistance test specimens. Constant current flow (provided the other channel is never activated) means that all charging processes are complete. A delay after CoE 0x80n0:31 [▸ 166] is therefore not required.

In single-channel mode measurement of the second channel can be triggered by writing 1 or 3 in the entry 0x80n0:0D [▸ 166] (mode).

Broken wire

The only way in which the EL3692 can detect a "broken wire" is therefore by registering sustained exceedance of the measuring range ("Overrange").
In 4-wire mode, there may be combinations of wire breakage that may not result in an unambiguous "Overrange" error and can only be detected based on sustained measured value variations in an implausible range.

5.6.7 Autorange function

If active, the Autorange function always selects a measuring range in which a value can be displayed.

Measurements

In 2-wire mode the Autorange function does not assume values below range 2 (10-100 ohm), since the three lower measuring ranges are not calibrated by the manufacturer. See also 4/2-wire mode.

5.6.8 Filter

In the COE (object 0x80n0:21 [▸ 168]) the EL3692 has a filter setting option for each channel. The principle is as follows:

- the raw resistance value is determined in rapid sequence
- the values are averaged over time x
- the time x is determined through inversion of the set filter frequency. Example: 10 Hz --> 100 ms averaging period
- Once the filter"time" has elapsed the determined resistance is forwarded for processing (diagrams etc.)

This process results in a notch characteristic as shown Fig. *Filter characteristic*. The filter frequency entered corresponds to the smallest blocked frequency. All multiples of the specified frequency are also blocked. The default setting "10 Hz" blocks 50 Hz and 60 Hz interference.

The filter setting has therefore direct influence on the conversion time of the respective channel.

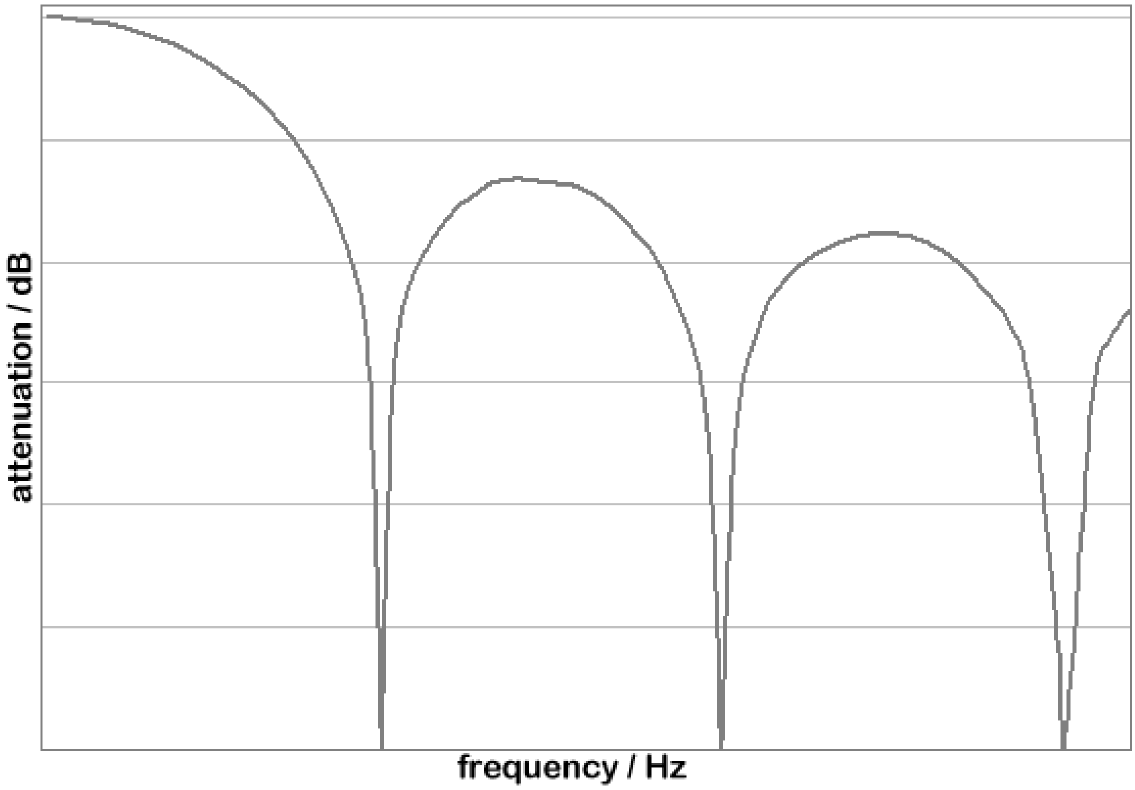


Fig. 153: Filter characteristic

5.6.9 Conversion time

The conversion time per channel depends on the selected filter setting and the set delay time (CoE 0x80n0:31) [▸ 168]. It applies from the start of the measurement (automatic or single-shot mode) and is calculated as follows:

$T_{conversion}$=1 / f_{filter} + T_{delay_time}

Note: in general, the EL3692 terminal operates in a two channel alternating mode, thus the conversion time doubles.

Filter setting	Conversion time after filtering
2.5 Hz	400 ms
5 Hz	200 ms
10 Hz (default)	100 ms
15 Hz	66 ms
25 Hz	40 ms
30 Hz	33 ms
50 Hz	20 ms
60 Hz	17 ms
100 Hz	10 ms

Table 2: Conversion times depending on filter settings

The specified values correspond to the approximate update time of the process image, if the field bus cycle time is smaller than or equal to the conversion time.

The ToggleBit *TxPDO toggle* can be used for verification.

5.6.10 Error Codes

The terminal reports its status through various bits in the status word. The possible combinations are listed and explained in the following table.

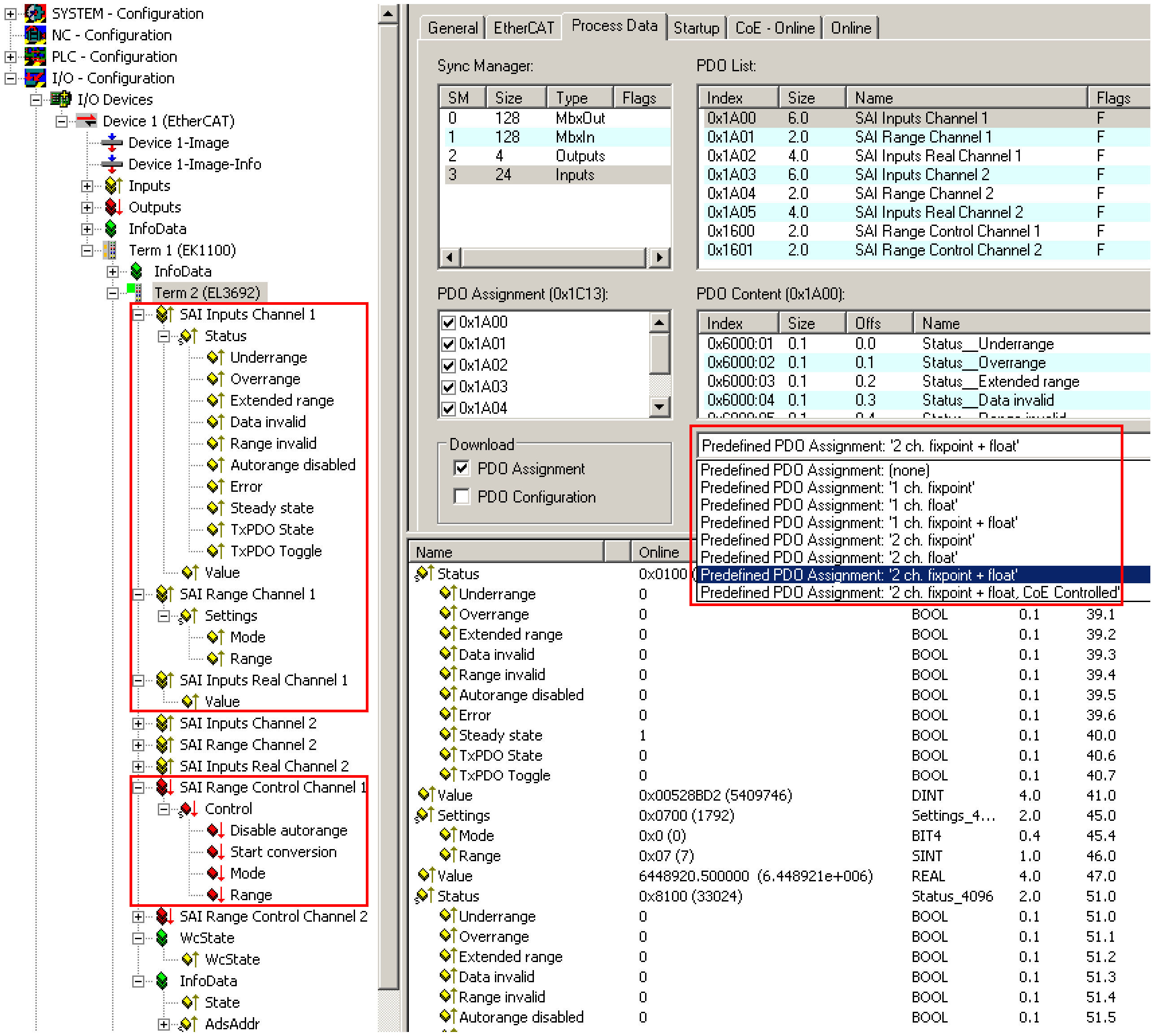
Error	Underrange	Overrange	Data invalid	TxPDO State	Range invalid	Error LED	Error description	Remedy/Error
1	1	0	0	0	0	off	Measurement is below range	If possible use a smaller measuring range; otherwise increase the input signal
1	0	1	0	0	0	on	Measuring range exceeded	If possible use a larger measuring range; otherwise reduce the input value
1	0	0	0	0	0	off	Display error	Select a suitable presentation
1	0	0	0	1	0	off	Conversion error	None (ADC does not respond, automatic restart)
1	0	0	0	0	1	off	Measuring range invalid	Select valid measuring range; with Autorange perhaps select Presentation: Scaled or activate SAI range (outputs)
0	0	0	1	0	0	off	Data invalid	Request conversion (after switch from continuous to single shot); ensure that measuring range is known
1	0	0	1	1	0	off	Conversion error after switching on	None; switch on again

Table 3: Error Codes

5.7 Process data

Introduction

This section describes the individual PDOs and their content. In each case only the first channel is described. The second channel is analogous. The individual PDOs can be activated or deactivated separately in the TwinCAT System Manager. The "Process data" tab is used for this (visible only if the terminal is selected on the left).



- SYSTEM - Configuration
- NC - Configuration
- PLC - Configuration
- I/O - Configuration
  - I/O Devices
    - Device 1 (EtherCAT)
      - Device 1-Image
      - Device 1-Image-Info
      - Inputs
      - Outputs
      - InfoData
      - Term 1 (EK1100)
        - InfoData
        - Term 2 (EL3692)
          - SAI Inputs Channel 1
            - Status
              - Underrange
              - Overrange
              - Extended range
              - Data invalid
              - Range invalid
              - Autorange disabled
              - Error
              - Steady state
              - TxPDO State
              - TxPDO Toggle
            - Value
          - SAI Range Channel 1
            - Settings
              - Mode
              - Range
          - SAI Inputs Real Channel 1
            - Value
          - SAI Inputs Channel 2
          - SAI Range Channel 2
          - SAI Inputs Real Channel 2
          - SAI Range Control Channel 1
            - Control
              - Disable autorange
              - Start conversion
              - Mode
              - Range
          - SAI Range Control Channel 2
          - WcState
            - WcState
          - InfoData
            - State
            - AdsAddr

Sync Manager:

SM	Size	Type	Flags
0	128	MbxOut	
1	128	MbxIn	
2	4	Outputs	
3	24	Inputs	

PDO List:

Index	Size	Name	Flags
0x1A00	6.0	SAI Inputs Channel 1	F
0x1A01	2.0	SAI Range Channel 1	F
0x1A02	4.0	SAI Inputs Real Channel 1	F
0x1A03	6.0	SAI Inputs Channel 2	F
0x1A04	2.0	SAI Range Channel 2	F
0x1A05	4.0	SAI Inputs Real Channel 2	F
0x1600	2.0	SAI Range Control Channel 1	F
0x1601	2.0	SAI Range Control Channel 2	F

PDO Assignment (0x1C13):

☑ 0x1A00
☑ 0x1A01
☑ 0x1A02
☑ 0x1A03
☑ 0x1A04

PDO Content (0x1A00):

Index	Size	Offs	Name
0x6000:01	0.1	0.0	Status__Underrange
0x6000:02	0.1	0.1	Status__Overrange
0x6000:03	0.1	0.2	Status__Extended range
0x6000:04	0.1	0.3	Status__Data invalid
0x6000:05	0.1	0.4	Status__Range invalid

Download
☑ PDO Assignment
☐ PDO Configuration

Predefined PDO Assignment: '2 ch. fixpoint + float'

Predefined PDO Assignment: (none)
Predefined PDO Assignment: '1 ch. fixpoint'
Predefined PDO Assignment: '1 ch. float'
Predefined PDO Assignment: '1 ch. fixpoint + float'
Predefined PDO Assignment: '2 ch. fixpoint'
Predefined PDO Assignment: '2 ch. float'
Predefined PDO Assignment: '2 ch. fixpoint + float'
Predefined PDO Assignment: '2 ch. fixpoint + float, CoE Controlled'

Name	Online			
Status	0x0100 (			
Underrange	0			
Overrange	0	BOOL	0.1	39.1
Extended range	0	BOOL	0.1	39.2
Data invalid	0	BOOL	0.1	39.3
Range invalid	0	BOOL	0.1	39.4
Autorange disabled	0	BOOL	0.1	39.5
Error	0	BOOL	0.1	39.6
Steady state	1	BOOL	0.1	40.0
TxPDO State	0	BOOL	0.1	40.6
TxPDO Toggle	0	BOOL	0.1	40.7
Value	0x00528BD2 (5409746)	DINT	4.0	41.0
Settings	0x0700 (1792)	Settings_4...	2.0	45.0
Mode	0x0 (0)	BIT4	0.4	45.4
Range	0x07 (7)	SINT	1.0	46.0
Value	6448920.500000 (6.448921e+006)	REAL	4.0	47.0
Status	0x8100 (33024)	Status_4096	2.0	51.0
Underrange	0	BOOL	0.1	51.0
Overrange	0	BOOL	0.1	51.1
Extended range	0	BOOL	0.1	51.2
Data invalid	0	BOOL	0.1	51.3
Range invalid	0	BOOL	0.1	51.4
Autorange disabled	0	BOOL	0.1	51.5

Fig. 154: Screenshot of System Manager, Process data

Fig. *Screenshot of System Manager, Process data*, left, shows the EL3692 with expanded process data, variables of channel 1 are highlighted in red. On the right the process data selection is highlighted in red.

Name of the PDO	Data direction	PDO number channel 1	PDO number channel 2
SAI Inputs	Input	0x1A00	0x1A03
SAI Range	Input	0x1A01	0x1A04
SAI Inputs Real	Input	0x1A02	0x1A05
SAI Range	Output	0x1600	0x1601

Table 1: PDO/channel numbers

Measured value display

The EL3692 can output the current channel-based measured value depending on the PDO option "*Predefined PDO assignment*" in 2 different ways. As "**fixpoint**" and "**float**". See following explanation:

- as 32-bit integer measuring range-dependent value, with 24-bit content "**fixpoint**"
  Explanation:
  Each of the 9 measuring ranges of the EL3692 extends over 1 decade (e.g. 0 .. 10 Ω). The regular value range of *value* extends from x0 .. x00 7F FF FF for each measuring range with "x00 7F FF FF ≈ full-scale value". In Autorange mode the measuring range is exceeded by up to 10%, in order to ensure the measuring range changeover. Note the status information.
  The following example is for the measuring range 10 Ω:

Input resistance	32-bit value, right-aligned display	Interpretation	Status bits
0 Ω	x00 00 00 00	0	Extended Range = 0 Overrange = 0
10 Ω	x00 7F FF FF	8.388.607	Extended Range = 1 Overrange = 0
> 11 Ω	x00 8C CC CB	9.227.467	Extended Range = 0 Overrange = 1

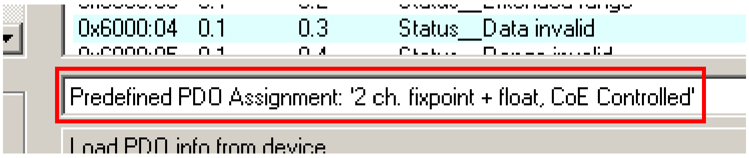
Please note that that the *relative* measuring error is very large near the lower end of the measuring range. For measurements within the range of less than 10% of the lower end of the measuring range the next lower measuring range should be used. The AutoRange function works accordingly.

- as 32-bit fixed-point float-value with mantissa and exponent, measuring range-independent **"float"**
  Explanation:
  This 32-bit variable can be linked directly with a FLOAT variable of the PLC according to IEC61131. The conversion takes place in the EL3692; no measuring range-dependent conversion by the user is required.

Predefined PDO Assignment

To simplify the configuration typical configurations based on process data are stored in the EtherCAT XML Device Description. The predefined configurations can be selected in the process data overview. The function is therefore only available if the XML files on the system are current (available for download from the Beckhoff website). The following combinations are possible:

- *1st ch fixpoint* only channel 1 measures, no REAL value
- *1st ch float* only channel 1 measures, with REAL but no settings
- *1st ch fixpoint* + *float* only channel 1 measures, REAL value and settings
- *2nd ch fixpoint* both channels, no REAL value
- *2nd ch float* both channels, with REAL but no settings
- *2nd ch fixpoint* + *float* both channels, REAL value and settings (standard)



```
0x6000:04  0.1   0.3   Status__Data invalid
```

Predefined PDO Assignment: '2 ch. fixpoint + float, CoE Controlled'

Fig. 155: Selection dialog predefined PDO

Selection CoE/PDO mode

The EL3692 requires an instruction as to whether it should work with or without Autorange etc. This can be set via CoE or the process data (PDO). This distinction is therefore also made via the predefined PDO selection: If the control word is displayed as PDO, the EL3692 primarily responds to the cyclic instructions from the control word. See also Explanations.

SAI Inputs 0x1A00 (0x1A03)

This PDO contains the status data and the resistance of the terminal. The resistance value is issued as a 32-bit value. The terminal status is shown as WORD.

Name	Explanation	Position, size	Bit in status WORD
Status → Underrange	Value below measuring range. Note: since each measuring range also measures up to 0 Ohm and there are no negative resistance values, this bit is never set.	0x6000:01, 1 bit	SW.0
Status → Overrange	Measuring range exceeded.	0x6000:02, 1 bit	SW.1
Status → Extended Range	Extended measuring range is used.	0x6000:03, 1 bit	SW.2
Status → Data invalid	Input data are invalid.	0x6000:04, 1 bit	SW.3
Status → Data invalid	No valid range selected.	0x6000:05, 1 bit	SW.4
Status → Autorange disabled	The Autorange function is disabled.	0x6000:06, 1 bit	SW.5
Status → Error	The error bit is set if the value is invalid (wire breakage, overrange, underrange)	0x6000:07, 1 bit	SW.6
Status → Steady state	If the last four values are no more than x / 1024 of end value apart, the "Steady state" bit is set to TRUE	0x6000:09, 1 bit	SW.8
Status → TxPDO State	Validity of the data of the associated TxPDO (0 = valid, 1 = invalid).	0x6000:0F, 1 bit	SW.14
Status → TxPDO Toggle	The TxPDO toggle is toggled by the slave when the data of the associated TxPDO is updated.	0x6000:10, 1 bit	SW.15
Value	Current measured value as INT32, measuring range end = 0x7F FF FF (standard)	0x6000:11, 4 bytes	-

Table 2: Process data 0x1A00

Example Program [▶ 161]: Decomposition of WORDS into bits.

SAI Inputs 0x1A01 (0x1A04)

This PDO contains information on the selected measuring mode and the current range. The two values are also available within a settings WORD.

Name	Explanation	Position, size
Mode	Current measuring mode: 0: 4-wire measurement 1: 4-wire measurement, single-shot mode 2: 2-wire measurement 3: 2-wire measurement, single-shot mode	0x6001:05, 4 bit
Range	Current measuring range (lower value is decisive for autorange operation): -1: 10 mΩ - 100 mΩ 0: 100 mΩ - 1 Ω 1: 1 Ω - 10 Ω 2: 10 Ω - 100 Ω 3: 100 Ω - 1 kΩ 4: 1 kΩ - 10 kΩ 5: 10 kΩ - 100 kΩ 6: 100 kΩ - 1 MΩ 7: 1 MΩ - 10 MΩ	0x6001:09, 1 byte

Table 3: Process data 0x1A01

SAI Inputs 0x1A02 (0x1A05)

This PDO contains the current measured value as REAL value. The format matches the REAL format from the TwinCAT PLC Control.

The format matches the REAL format of IEC 61131-3, which in turn is based on the REAL format of IEC 559. A REAL number (single precision) is defined as follows (See also Beckhoff InfoSys: TwinCAT PLC Control: standard data types).

Bit position (from left)	1	8	23 (+1 "hidden bit", see IE559)
Function	Sign	Exponent	Mantissa

Table 4: Structure of the REAL value

Name	Explanation	Position, size
Value	Current measured value as REAL	0x6002:01, 4 bytes

Table 5: Process data 0x1A02

SAI Inputs 0x1600 (0x1601)

The terminal can be controlled via this PDO. The data are also available as Control WORD.

Name	Explanation	Position, size	Bit in Control WORD
Control → Disable autorange	Request to disable Autorange.	0x6000:01, 1 bit	CW.0
Control → Start conversion	Measurement request (rising edge), only required in single shot mode.	0x6000:04, 1 bit	CW.3
Control → Mode	Current measuring mode: 0: 4-wire measurement 1: 4-wire measurement, single-shot mode 2: 2-wire measurement 3: 2-wire measurement, single-shot mode	0x6001:05, 4 bit	CW.4 - CW.7
Control → Range	Current measuring range (lower value is decisive for autorange operation): -1: 10 mΩ - 100 mΩ 0: 100 mΩ - 1 Ω 1: 1 Ω - 10 Ω 2: 10 Ω - 100 Ω 3: 100 Ω - 1 kΩ 4: 1 kΩ - 10 kΩ 5: 10 kΩ - 100 kΩ 6: 100 kΩ - 1 MΩ 7: 1 MΩ - 10 MΩ	0x6001:09, 1 byte	CW.8 - CW.15

Table 6: Process data 0x1600

Example Program [▶ 161]: Decomposition of WORDS into bits.

5.8 Specific data

This section describes various terminal-specific data.

5.8.1 Measuring currents and voltages

Measuring currents and voltages used

The following measuring currents and voltages are used, depending on the measuring range.

Measuring range	Typical measuring voltage (max.)	Typical measuring current.
-1 (0 -100 mΩ)	~ 4.5 mV (3.0 V)	~ 45 mA
0 (100 mΩ -1 Ω)	~ 45 mV (3.0 V)	~ 45 mA
1 (1 Ω-10 Ω)	~ 450 mV (3.0 V)	~ 45 mA
2 (10 Ω-100 Ω)	~ 16 mV (3.0 V)	~ 0.16 mA
3 (100 Ω -1 kΩ)	~ 160 mV (3.0 V)	~ 0.16 mA
4 (1 kΩ -10 kΩ)	~ 1.6 V (3.0 V)	~ 0.16 mA
5 (10 kΩ -100 kΩ)	~ 16 mV (3.0 V)	<< 16 µA
6 (100 kΩ -1 MΩ)	~ 160 mV (3.0 V)	<< 16 µA
7 (1 MΩ -10 MΩ)	~ 1.6 V (3.0 V)	<< 16 µA

Table 1: Measuring ranges and associated measuring currents and voltages

5.8.2 Capacitive and inductive influences

Internal and external capacitive and inductive influences may distort the measurement reading through their dynamic characteristics. In order to minimize these influences, a delay is set between switching on of the measuring current and the start of the measurement. The delay time can be set via object 0x80n0:31 [▶ 168] and switched on or off for each channel via object 0x80n0:05 [▶ 168]. The delay time is set in 0x80n0:37 as a channel-specific value. In the measuring ranges -1 to 5 this value is multiplied with 20 ms, in the measuring ranges 6 and 7 with 50 ms. The standard delay time is therefore 200 or 500 ms.
The delay time does not become active if two consecutive measurements take place in the same measuring range and on the same channel.
The delay time largely allows charging processes at the test specimen to complete. Once the charging process is complete based on the sensitivity setting (steady-state tolerance object 0x80n0:3E [▶ 168]), the steady state bit is set to TRUE.

5.8.3 Heating of the test specimen

In the lower 3 measuring ranges the measuring current may result in heating of the test specimen. In the upper measuring ranges the current through the test specimen is so low that heating of the test specimen is not an issue.

Measuring range	Max. power in the test specimen (alternating mode)
-1 (10 mΩ -100 mΩ)	~450 µW (~220 µW)
0 (100 mΩ -1 Ω)	~4.5 mW (~2.2 mW)
1 (1 Ω-10 Ω)	~45 mW (~22 mW)

Table 2: Power in the test specimen

5.8.4 Calculating the resistance value

The terminal calculates the value of the ADC (filtered, X_{adc}) with the manufacturer calibration offset ($Offset_{Vendor}$) and the manufacturer gain Gain ($Gain_{Vendor}$). The user offset ($Offset_{User}$) and user gain ($Gain_{User}$) are also taken into account.
The two calculations can be activated or deactivated via CoE object 0x80n0:01 [▶ 168] and 0x80n0:02 [▶ 168].

$X_{Vendor} = (X_{adc} - Offset_{Vendor}) * Gain_{Vendor}$ / 0x10000 (object 0x80n0:01 [▶ 168] activated)
$X_{Vendor} = X_{adc}$ (object 0x80n0:01 [▶ 168] deactivated)

PDO = $(X_{Vendor} - Offset_{User}) * Gain_{User}$ / 0x10000 (object 0x80n0:02 [▶ 168] activated)
PDO = X_{Vendor} (object 0x80n0:02 [▶ 168] deactivated)

This is followed by scaling for the INT value and the REAL value.

Note: the unit of X, Offset, Gain depends on the resolution (see chapter Resolution [▶ 156]).

Extended measuring range

All measuring ranges have an option to extend the range by 10% beyond the upper limit. This range is used to allow for the hysteresis of the Autorange function.
If the measured value is within extended measuring range, the Extended Range bit is set to TRUE.

Left-aligned presentation

The extended measuring range cannot be used in left-aligned mode.
In this case the Autorange function operates without the 10% measuring range extension. This combination is not permitted.

INT value

The measured value is displayed based on the set presentation.

Presentation	Meaning	Example
0: left-aligned	31 bit+sign, end value:0x7FFFFF00 (extended measuring range cannot be used)	1,000,000,000$_{dec}$~4.7 kΩ (range 4)
1: right aligned	24 bit+sign, end value:0x7FFFFF	5,000,000$_{dec}$~6 kΩ (range 4)
2: scaled 1 bit / µΩ	Measured value in µΩ (upper limit at 2.147 kΩ)	1,000,000$_{dec}$=1 Ω
3: scaled 1 bit / mΩ	Measured value in mΩ (upper limit at 2.147 MΩ, not recommended for measuring ranges below 10 Ω)	1,000,000$_{dec}$=1 kΩ
4: scaled 1 bit / Ω	Measured value in Ω (not recommended for measuring ranges below 10 kΩ)	1,000,000$_{dec}$=1 MΩ
5: as REAL value	The calculated REAL value is entered instead of the INT value.	-

Table 3: Presentation

Only the REAL representation covers the whole measuring range including the measuring range with uniform accuracy.

REAL value

The following equation is used for calculating the REAL value from the value that has already been offset against the manufacturer and user values (F_{scal} the respective full scale value (100 mΩ, 1 Ω, 10 Ω, 100 Ω, 1 kΩ, 10 kΩ, 100 kΩ, 1 MΩ and 10 MΩ):

$Y_{real}=Y_A * F_{scal} * 2^{-23}$

5.8.5 Error analysis

The main difference between the EL3692 and other analog EL3xxx input terminals from Beckhoff is the fact that it has not just 1, but 9 consecutive measuring ranges.
This opens up new aspects for error and measurement uncertainty analysis.
The absolute measuring error of the EL3692 with 4-wire connection is <0.5% relative to the respective full-scale value.
The thus assured maximum absolute error (relative to the respective full-scale value) is specified as the upper error limit below.

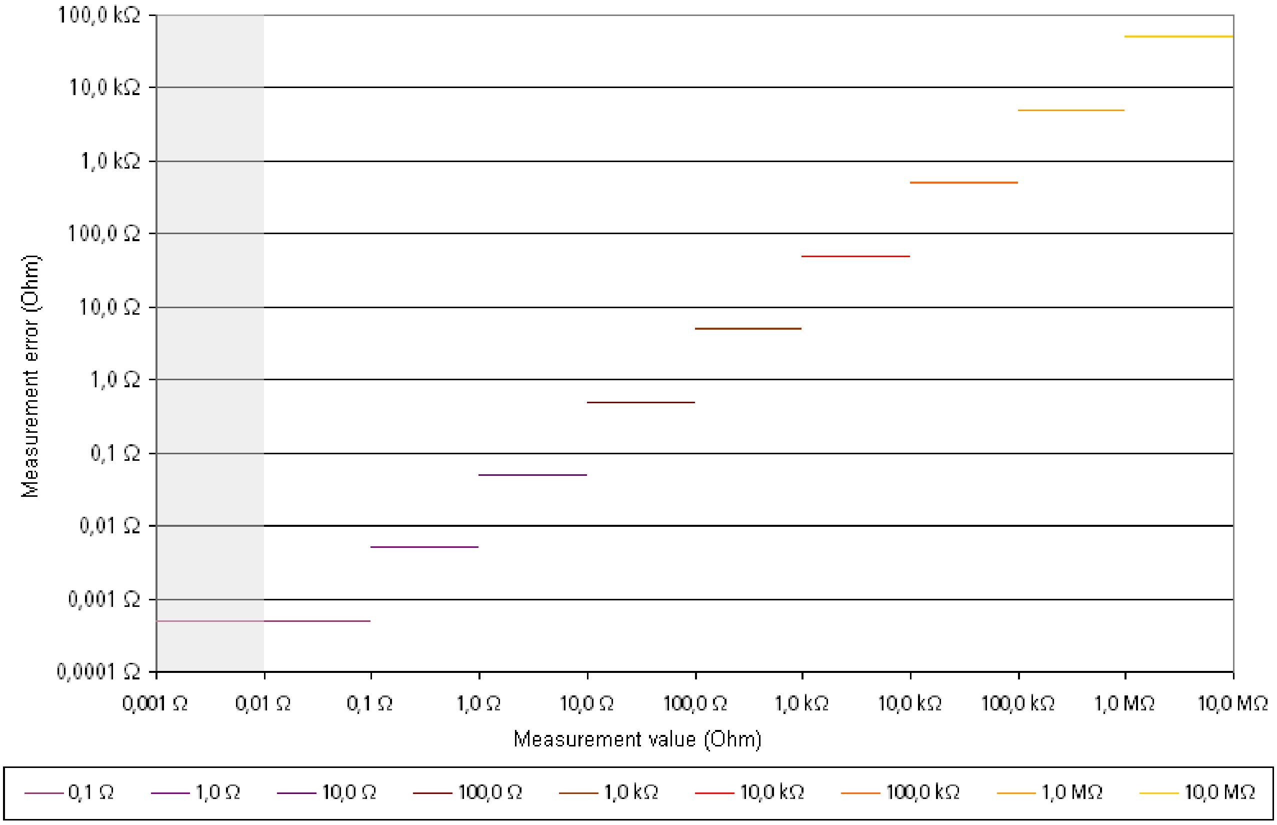


Fig. 156: Max. absolute measuring error for each measuring range, full logarithmic diagram

Example: Measurement of 50 Ω in the range 10-100 Ω --> maximum error: 0.5 Ω

This upper error limit results in the following values for the maximum relative measuring error relative to the **current measured value** (logarithmic diagram).

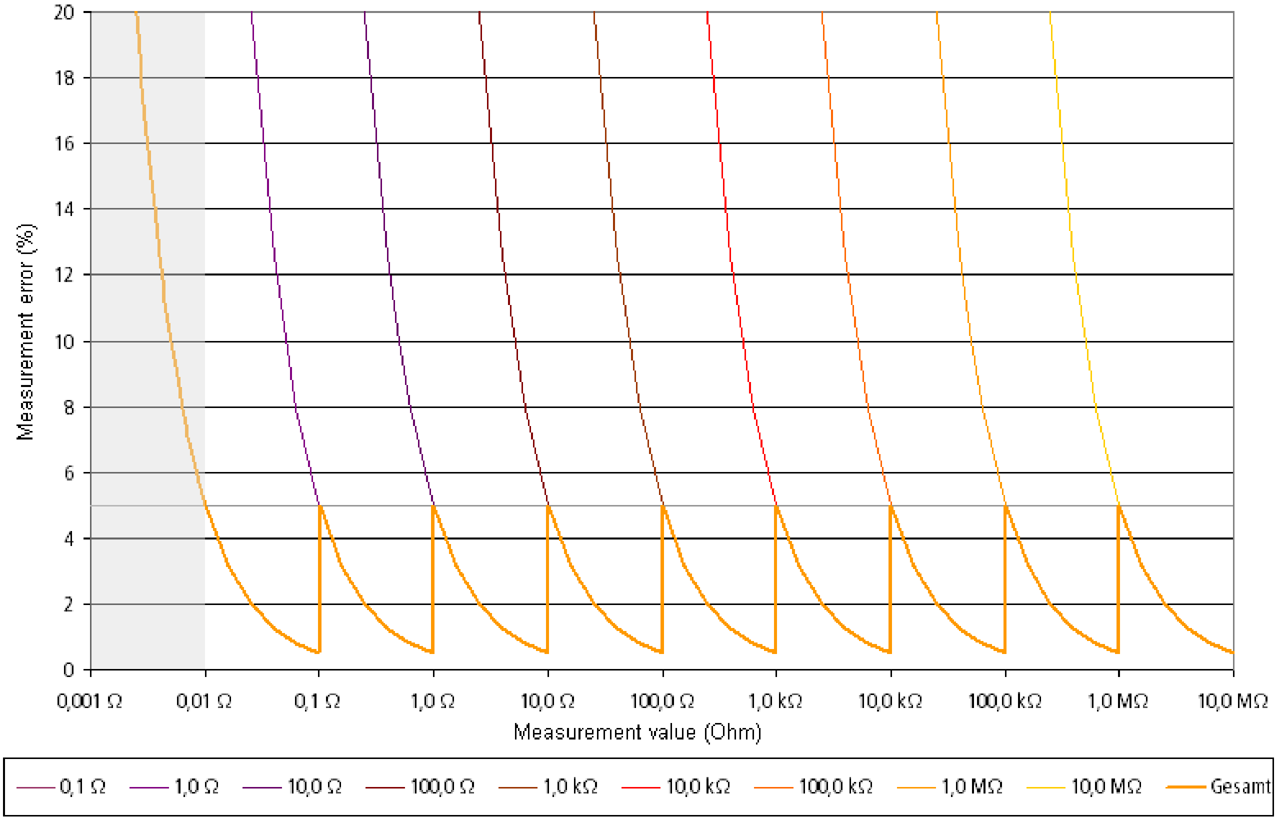


Fig. 157: Calculated maximum relative measuring error relative to the current measured value in the respective measuring range

Example from above: 50 Ω -> max. error 0.5 Ω, corresponding to 1% relative measuring error. The grey area is only shown for illustration and is not a valid measuring range.

5.8.6 Resolution

The following table shows the resolution for each measuring range.

Measuring range	Measured variable	Resolution
-1 (0xFF/255)	0 - 0.1 Ω	11.9 nΩ/Digit
0	100 mΩ - 1 Ω	119 nΩ/Digit
1	1 Ω - 10 Ω	1.19 µΩ/Digit
2	10 Ω - 100 Ω	11.9 µΩ/Digit
3	100 Ω - 1 kΩ	119 µΩ/Digit
4	1 kΩ - 10 kΩ	1.19 mΩ/Digit
5	10 kΩ - 100 kΩ	11.9 mΩ/Digit
6	100 kΩ - 1 MΩ	119 mΩ/Digit
7	1 MΩ - 10 MΩ	1.19 Ω/Digit

5.9 Data processing

Table of contents

The following two diagrams illustrate the internal measurement sequence in the terminal. The first diagram shows the route for measurement of a value, the second diagram shows processing of the value in the terminal.

Measurement sequence

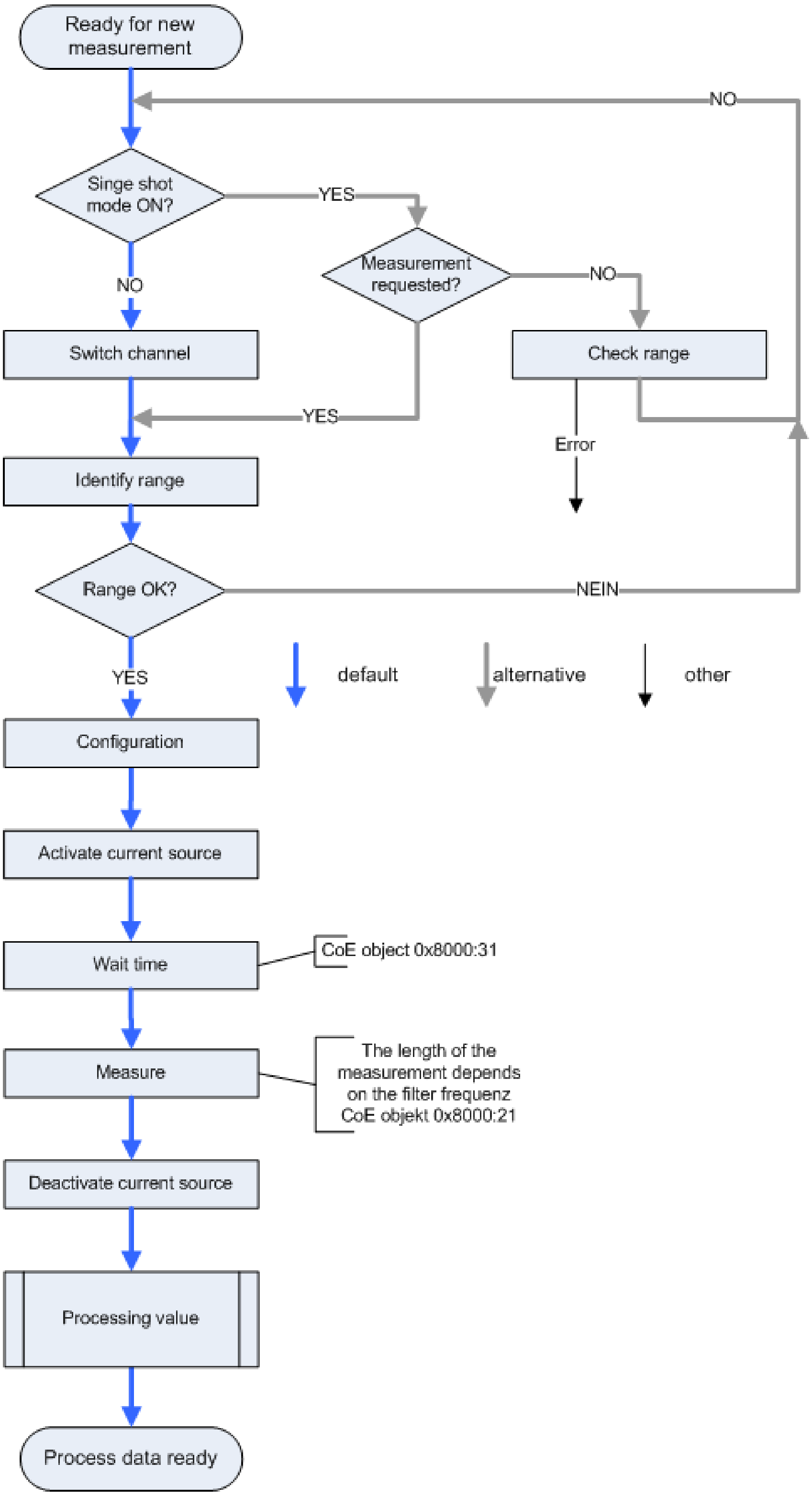


Fig. 158: Measurement sequence

Processing sequence

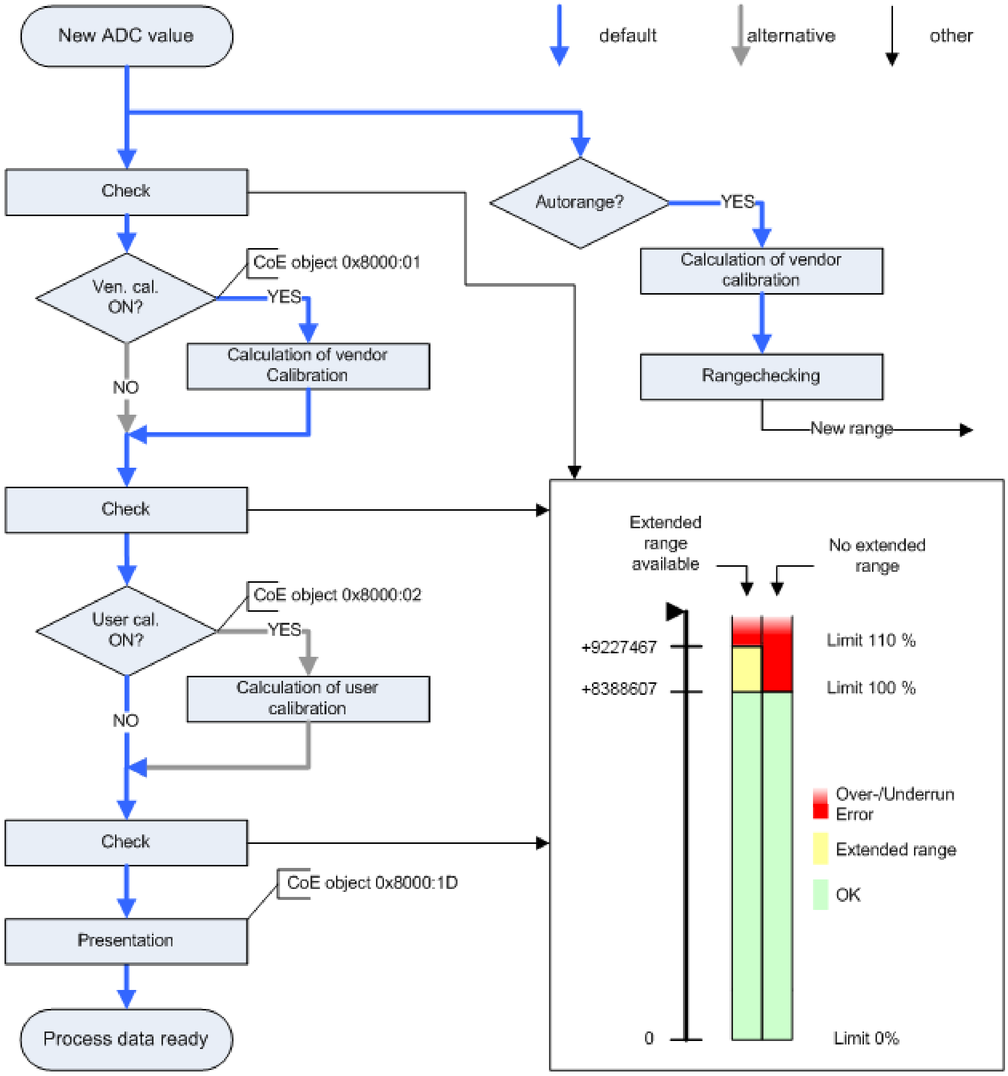


Fig. 159: Data processing sequence

5.10 DC operation mode

The EL3692 does not support Distributed Clocks operation mode.

5.11 Example program

Using the example programs

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

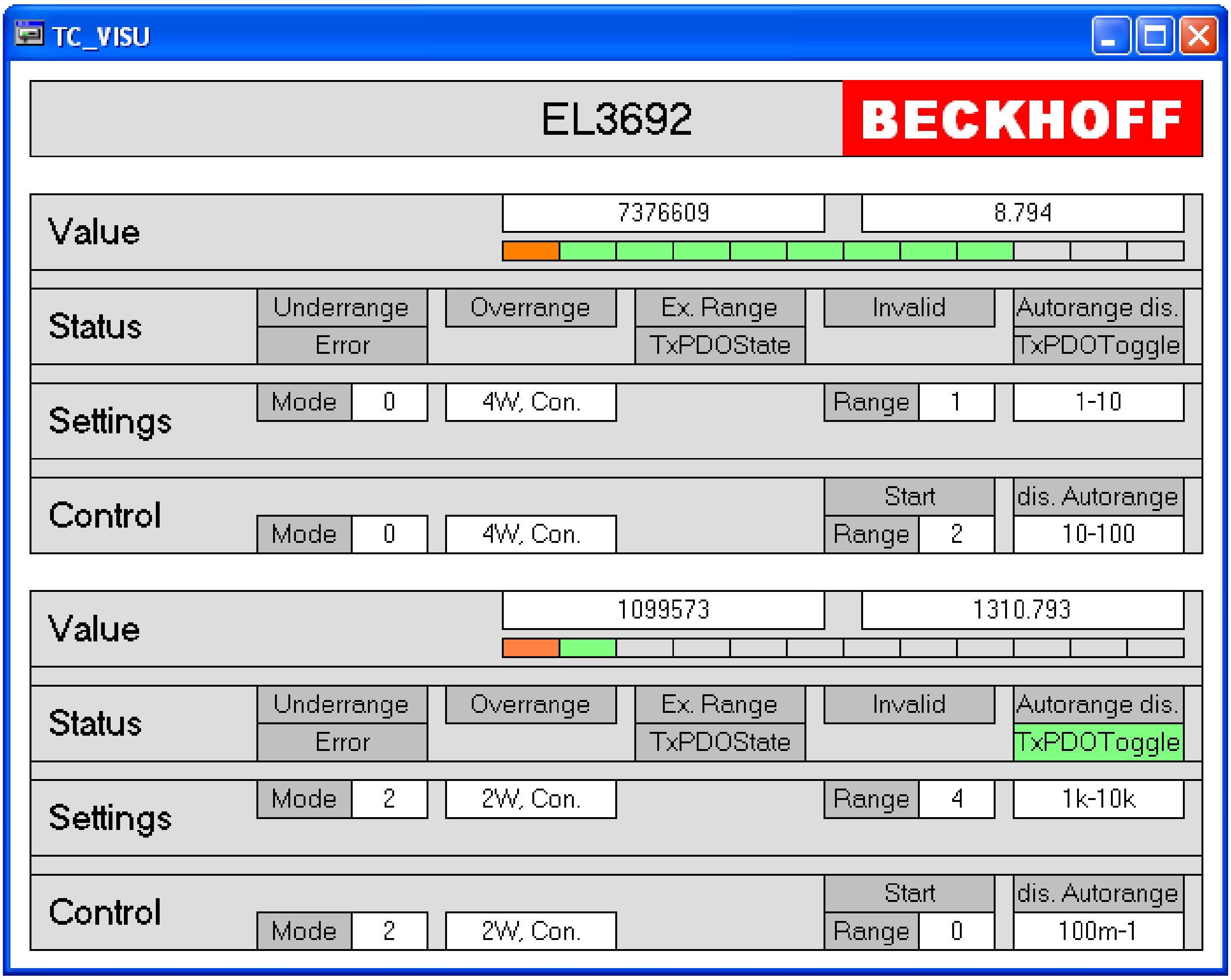
The program presented here reads the individual values from the terminal and visualizes them. The process data "Status" and "Control" are linked as WORD in the System Manager and decomposed into the individual bits in the PLC.

The visualization displays both channels above each other; the control takes place via the "Control" section. If a field is highlighted in color, the associated bit is TRUE.

The measuring mode can be changed via the numerical entry in the field on the right next to the mode entry. In the two single shot settings a new measurement can be requested by clicking on Start.

The Autorange function can be disabled by clicking on "Dis. Autorange". The measuring range can then be selected manually via the numeric field to the right of the range.

Download (https://infosys.beckhoff.com/content/1033/el3692/Resources/2079184779/.zip)



EL3692

Value: 7376609 | 8.794

Status: Underrange | Error | Overrange | Ex. Range | TxPDOState | Invalid | Autorange dis. | TxPDOToggle

Settings: Mode 0 | 4W, Con. | Range 1 | 1-10

Control: Start | dis. Autorange | Mode 0 | 4W, Con. | Range 2 | 10-100

Value: 1099573 | 1310.793

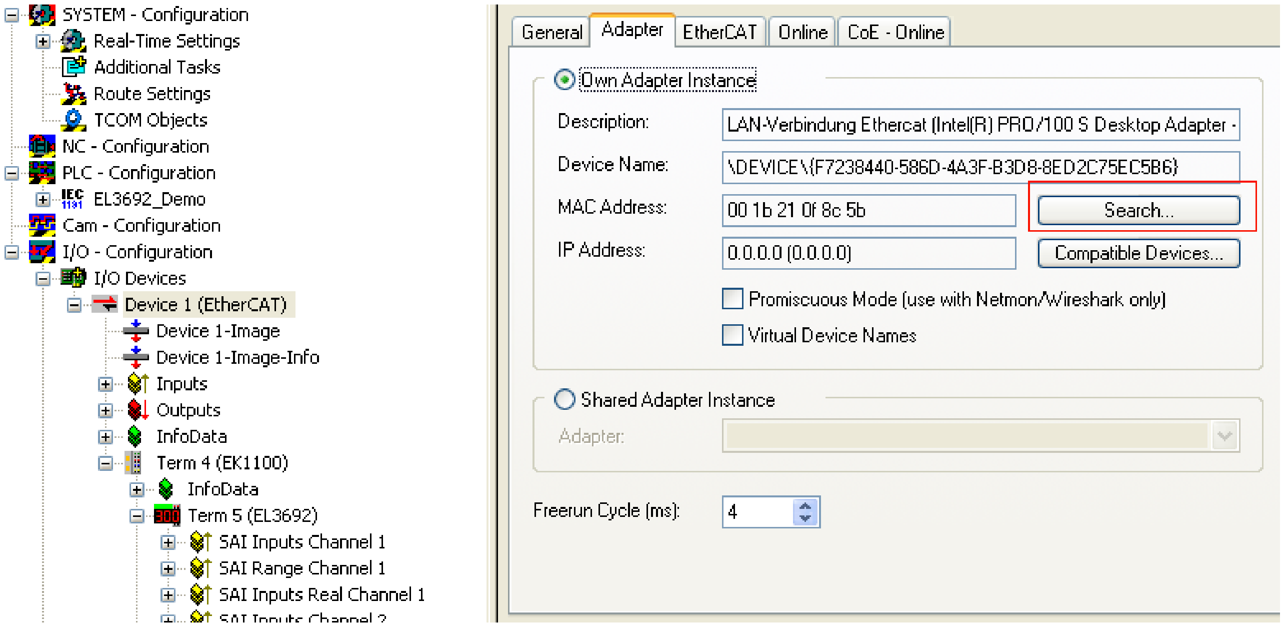
Status: Underrange | Error | Overrange | Ex. Range | TxPDOState | Invalid | Autorange dis. | TxPDOToggle

Settings: Mode 2 | 2W, Con. | Range 4 | 1k-10k

Control: Start | dis. Autorange | Mode 2 | 2W, Con. | Range 0 | 100m-1

Fig. 160: Example program

After opening the .tsm file with the System Manager the EtherCAT master must be notified of the current MAC address. Click the "Search..." button:



● Own Adapter Instance

Description: LAN-Verbindung Ethercat (Intel(R) PRO/100 S Desktop Adapter -

Device Name: \DEVICE\{F7238440-586D-4A3F-B3D8-8ED2C75EC5B6}

MAC Address: 00 1b 21 0f 8c 5b

IP Address: 0.0.0.0 (0.0.0.0)

□ Promiscuous Mode (use with Netmon/Wireshark only)

□ Virtual Device Names

○ Shared Adapter Instance

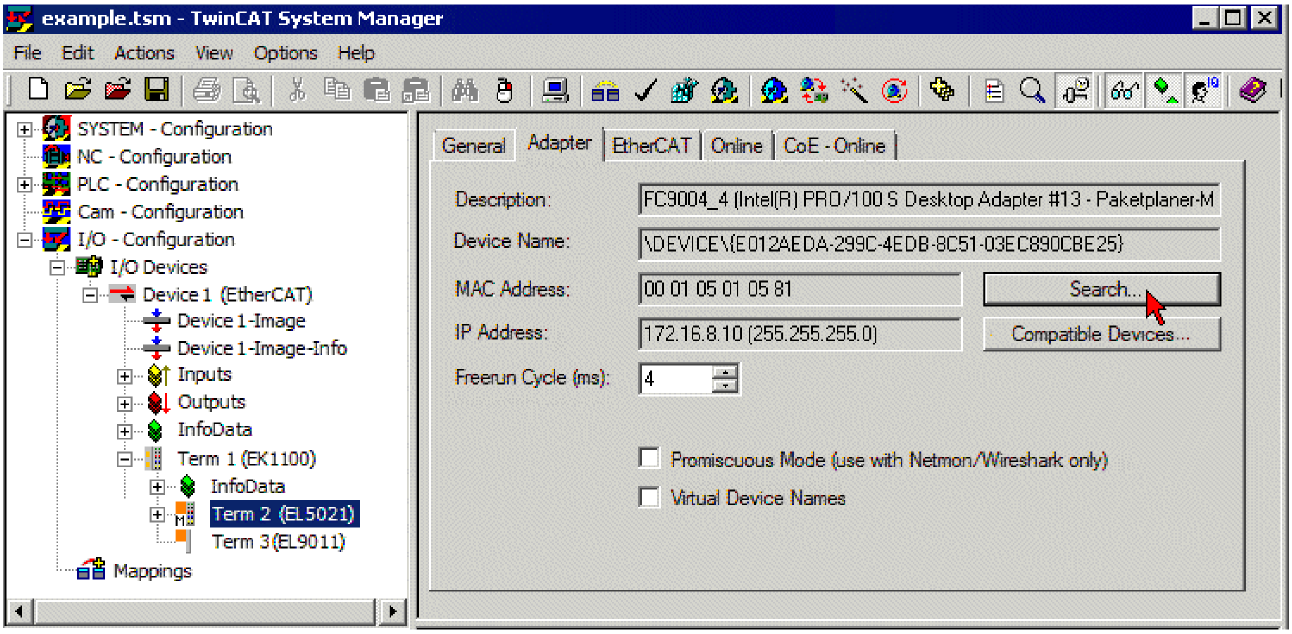
Adapter:

Freerun Cycle (ms): 4

Fig. 161: Find MAC address

Procedure for starting the program

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



Description: FC9004_4 (Intel(R) PRO/100 S Desktop Adapter #13 - Paketplaner-M

Device Name: \DEVICE\{E012AEDA-299C-4EDB-8C51-03EC890CBE25}

MAC Address: 00 01 05 01 05 81

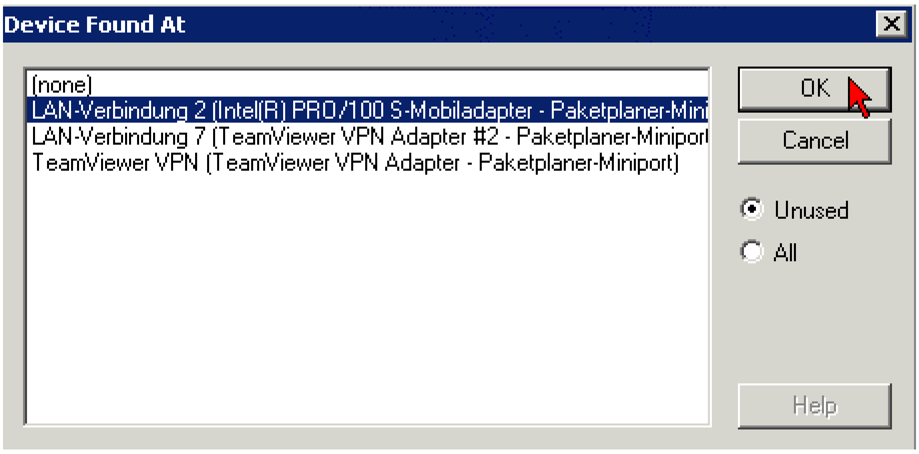
IP Address: 172.16.8.10 (255.255.255.0)

Freerun Cycle (ms): 4

□ Promiscuous Mode (use with Netmon/Wireshark only)

□ Virtual Device Names

Fig. 162: Searching the Ethernet adapter



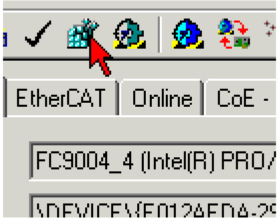
Device Found At

(none)
LAN-Verbindung 2 (Intel(R) PRO/100 S-Mobiladapter - Paketplaner-Mini
LAN-Verbindung 7 (TeamViewer VPN Adapter #2 - Paketplaner-Miniport
TeamViewer VPN (TeamViewer VPN Adapter - Paketplaner-Miniport)

● Unused
○ All

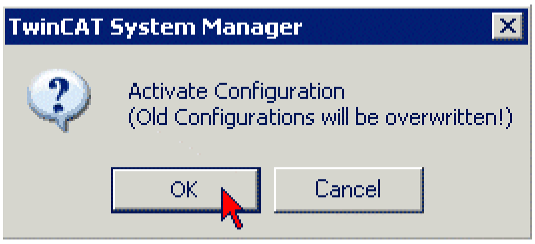
Fig. 163: Selection and confirmation of the Ethernet adapter

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



FC9004_4 (Intel(R) PRO/
\DEVICE\{E012AEDA-2

Fig. 164: Activation of the configuration

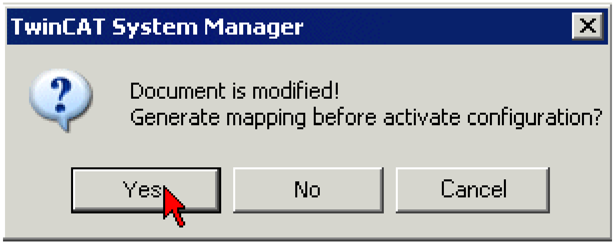


TwinCAT System Manager

Activate Configuration
(Old Configurations will be overwritten!)

Fig. 165: Confirming the activation of the configuration

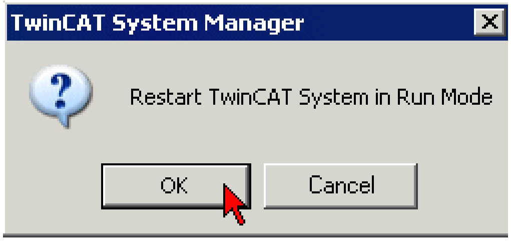
- Confirm new variable mapping, restart in RUN mode (Fig. *Generate variable mapping* + *Restarting TwinCAT in RUN mode*)



TwinCAT System Manager

Document is modified!
Generate mapping before activate configuration?

Fig. 166: Generating variable mapping



TwinCAT System Manager

Restart TwinCAT System in Run Mode

Fig. 167: Restarting TwinCAT in RUN mode

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

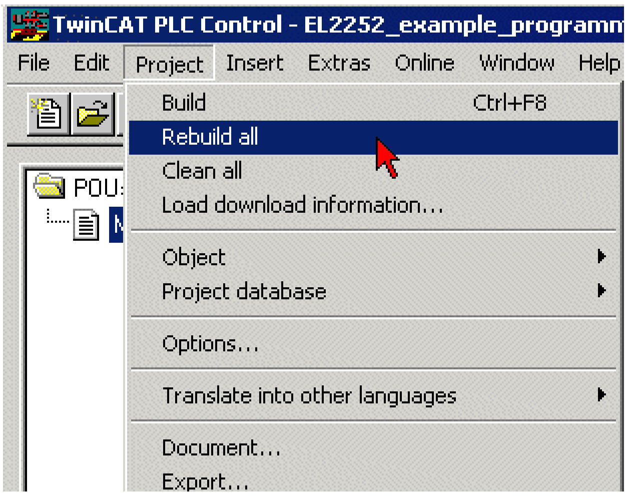
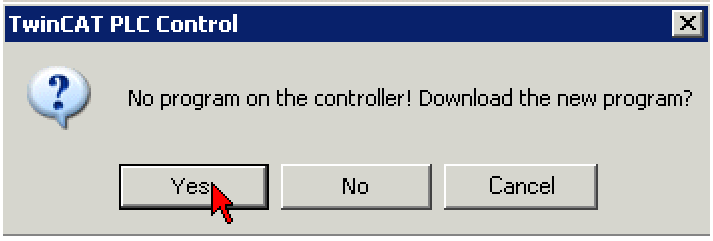


Fig. 168: Compile project

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


TwinCAT PLC Control

No program on the controller! Download the new program?

Fig. 169: Confirming program start

5.12 CoE

5.12.1 Most important CoE entries

Table of contents

This chapter describes the CoE objects required for commissioning and user calibration.

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

Objects for commissioning

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters	Restore default parameters	UINT8	RO	0x01 (1_{dez})
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_{dez})

Index 80n0 SAI Settings (for Ch.1 (n=0); Ch.2 (n=1)

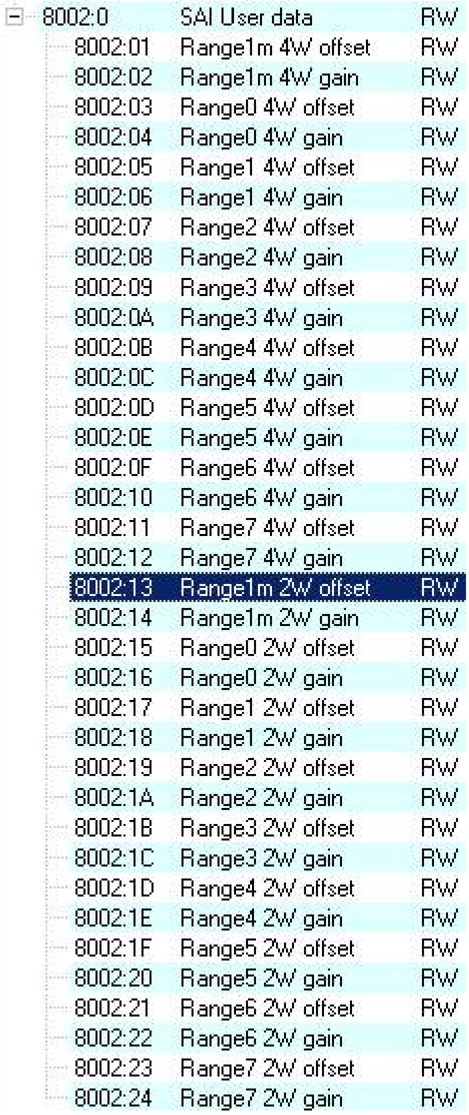
Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	SAI Settings	Maximum subindex	UINT8	RW	0x3E (62_{dez})
80n0:01	Enable vendor calibration	Manufacturer calibration is active.	BOOLEAN	RW	0x01 (1_{dec})
80n0:02	Enable user calibration	User calibration is active.	BOOLEAN	RW	0x00 (0_{dec})
80n0:05	Enable delay time	The set delay time [▸ 166] for this channel is active.	BOOLEAN	RW	0x01 (1_{dec})
80n0:09	Disable Autorange	The Autorange function is disabled. **Is only saved if the output data are not active.**	BOOLEAN	RW	0x00 (0_{dec})
80n0:0D	Mode	Measuring mode of the terminal: 0: 4-wire measurement 1: 4-wire measurement (single-shot mode) 2: 2-wire measurement 3: 2-wire measurement (single-shot mode) **Is only saved if the output data are not active.**	Bit-Field[4]	RW	0x00 (0_{dec})
80n0:11	Range	Measuring range: -1: 10 mΩ - 100 mΩ 0: 100 mΩ - 1 Ω 1: 1 - 10 Ω 2: 10 - 100 Ω 3: 100 - 1 kΩ 4: 1 k - 10 kΩ 5: 10 k - 100 kΩ 6: 100 k - 1 MΩ 7: 1 M - 10 MΩ **Is only applied if the output data are not active.**	INT8	RW	0x00 (0_{dec})
80n0:1D	Presentation	0: left-aligned 1: right-aligned 2: scaled 1 bit / µΩ 3: scaled 1 bit / mΩ 4: scaled 1 bit / Ω 5: Display as REAL value See also: Specific data [▸ 153]	Bit-Field[4]	RW	0x01 (1_{dec})
80n0:21	Filter Settings	Setting the hardware filter in ADC 0: 2.5 Hz 1: 5 Hz 2: 10 Hz 3: 15 Hz 4: 25 Hz 5: 30 Hz 6: 50 Hz 7: 60 Hz 8: 100 Hz	UINT16	RW	0x02 (2_{dec})
80n0:31	Delay time	Delay time for this channel n*20 ms in the measuring ranges <1 MΩ, n*50 ms in the measuring range 7 (1 MΩ - 10 MΩ)	UINT8	RW	0x0A (10_{dec})
80n0:3E	Steady state tolerance	If the last four values are no more than x / 1024 of end value apart, the "Steady state" bit is set to TRUE	UINT16	RW	0x0A (10_{dec})

Objects for the user calibration

User calibration offers the option to process the value of the terminal further after manufacturer calibration. This can relieve the PLC.

Index 80n2 SAI User Data (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n2:0	SAI User Data	Maximum subindex	UINT8	RO	0x24 (36_{dec})
80n2:0m (odd)	Range x 4 wire / 2 wire offset	one offset value per measuring range for user calibration (*see fig. below*)	UINT32	RW	0x00000000 (0_{dec})
80n2:0m (even)	Range x 2/4W gain	One gain value per measuring range for user calibration 0x10000 corresponds to 1	UINT32	RW	0x10000 (65536_{dec})



8002:0	SAI User data	RW
8002:01	Range1m 4W offset	RW
8002:02	Range1m 4W gain	RW
8002:03	Range0 4W offset	RW
8002:04	Range0 4W gain	RW
8002:05	Range1 4W offset	RW
8002:06	Range1 4W gain	RW
8002:07	Range2 4W offset	RW
8002:08	Range2 4W gain	RW
8002:09	Range3 4W offset	RW
8002:0A	Range3 4W gain	RW
8002:0B	Range4 4W offset	RW
8002:0C	Range4 4W gain	RW
8002:0D	Range5 4W offset	RW
8002:0E	Range5 4W gain	RW
8002:0F	Range6 4W offset	RW
8002:10	Range6 4W gain	RW
8002:11	Range7 4W offset	RW
8002:12	Range7 4W gain	RW
8002:13	Range1m 2W offset	RW
8002:14	Range1m 2W gain	RW
8002:15	Range0 2W offset	RW
8002:16	Range0 2W gain	RW
8002:17	Range1 2W offset	RW
8002:18	Range1 2W gain	RW
8002:19	Range2 2W offset	RW
8002:1A	Range2 2W gain	RW
8002:1B	Range3 2W offset	RW
8002:1C	Range3 2W gain	RW
8002:1D	Range4 2W offset	RW
8002:1E	Range4 2W gain	RW
8002:1F	Range5 2W offset	RW
8002:20	Range5 2W gain	RW
8002:21	Range6 2W offset	RW
8002:22	Range6 2W gain	RW
8002:23	Range7 2W offset	RW
8002:24	Range7 2W gain	RW

Fig. 170: Offset value per measuring range for user calibration

5.12.2 Object description and parameterization

Introduction

The CoE overview contains objects for different intended applications. The following section describes the standard objects, followed by the objects that are specific for this terminal.

Parameterization

The terminal is parameterized via the CoE - Online tab [▸ 109] (double-click on the respective object) or via the Process Data tab [▸ 106](allocation of PDOs).

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.

Startup list

The startup list can be used in cases where values are to be written whenever the system starts. The list can be found in the Startup tab of the TwinCAT System Manager (if the terminal is selected on the left).

5.12.3 Restore object

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters [▸ 190]	Restore default parameters	UINT8	RO	0x01 (1_{dec})
1011:01	SubIndex 001	If this object is set to **"0x64616F6C"** in the set value dialog, all backup objects are reset to their delivery state.	UINT32	RW	0x00000000 (0_{dec})

5.12.4 Configuration data

Index 80n0 SAI Settings (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	SAI Settings	Maximum subindex	UINT8	RW	0x31 (49_{dec})
80n0:01	Enable vendor calibration	Manufacturer calibration is active.	BOOLEAN	RW	0x01 (1_{dec})
80n0:02	Enable user calibration	User calibration is active.	BOOLEAN	RW	0x00 (0_{dec})
80n0:05	Enable delay time	The set delay time [▸ 168] is active.	BOOLEAN	RW	0x00 (0_{dec})
80n0:09	Disable Autorange	The Autorange function is disabled. **Is only applied if the output data are not active.**	BOOLEAN	RW	0x00 (0_{dec})
80n0:0D	Mode	Measuring mode of the terminal: 0: 4-wire measurement 1: 4-wire measurement, single-shot mode 2: 2-wire measurement 3: 2-wire measurement, single-shot mode **Is only applied if the output data are not active.**	Bit-Field[4]	RW	0x00 (0_{dec})
80n0:11	Range	Measuring range (lower value as guide value for the AutoRange operating mode): -1: 10m - 100m Ω 0: 100m - 1 Ω 1: 1 - 10 Ω 2: 10 - 100 Ω 3: 100 - 1k Ω 4: 1k - 10k Ω 5: 10k - 100k Ω 6: 100k - 1M Ω 7: 1M - 10M Ω **Is only applied if the output data are not active.**	INT8	RW	0x00 (0_{dec})
80n0:1D	Presentation	0: left-aligned 1: right-aligned 2: Scaled 1 bit / µΩ 3: Scaled 1 bit / mΩ 4: Scaled 1 bit / Ω 5: Display as REAL value	Bit-Field[4]	RW	0x01 (1_{dec})
80n0:21	Filter Settings	Setting the hardware filter in ADC 0: 2.5 Hz 1: 5 Hz 2: 10 Hz 3: 15 Hz 4: 25 Hz 5: 30 Hz 6: 50 Hz 7: 60 Hz 8: 100 Hz	UINT16	RW	0x02 (2_{dec})
80n0:31	Delay time	Delay time	UINT8	RW	0x00 (0_{dec})
80n0:3E	Steady state tolerance	If the last four values are no more than x / 1024 of end value apart, the "Steady state" bit is set to TRUE	UINT8	RW	0x00 (0_{dec})

Index 80n2SAI User Data (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n2:0	SAI User Data	Maximum subindex	UINT8	RO	0x24 (36_{dec})
80n2:0m (odd)	Range x 2/4W offset	one offset value per measuring range for user calibration	UINT32	RW	0x00000000 (0_{dec})
80n2:0m (even)	Range x 2/4W gain	one gain value per measuring range for user calibration 0x10000 corresponds to 1	UINT32	RW	0x10000 (65536_{dec})

5.12.5 Input data

Index 60n0 SAI Inputs (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	TC Inputs Ch.1	Maximum subindex	UINT8	RO	0x11 (17_{dec})
60n0:01	Underrange	Value below measuring range.	BOOLEAN	RO	0x00 (0_{dec})
60n0:02	Overrange	Measuring range exceeded.	BOOLEAN	RO	0x00 (0_{dec})
60n0:03	Extended Range	Extended measuring range is used.	BOOLEAN	RO	0x00 (0_{dec})
60n0:03	Data invalid	Input data are invalid.	BOOLEAN	RO	0x00 (0_{dec})
60n0:05	Range invalid	No valid range selected.	BOOLEAN	RO	0x00 (0_{dec})
60n0:06	Autorange disabled	The Autorange function is disabled.	BOOLEAN	RO	0x00 (0_{dec})
60n0:07	Error	The error bit is set if the value is invalid (wire breakage, overrange, underrange)	BOOLEAN	RO	0x00 (0_{dec})
60n0:09	Steady state	The bit is set if the last four values differ from the full-scale value by no more than the value of "Steady state tolerance" (index 0x80n0:3E) / 1024.	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	Analog input value as INT32	INT32	RO	0x0000 (0_{dec})

Index 60n1 SAI Range (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	TC Inputs Ch.1	Maximum subindex	UINT8	RO	0x09 (9_{dec})
60n1:05	Mode	Current measuring mode: 0: 4-wire measurement 1: 4-wire measurement, single-shot mode 2: 2-wire measurement 3: 2-wire measurement, single-shot mode	Bit-Field[4]	RO	0x00 (0_{dec})
60n1:09	Range	Current measuring range: -1: 10 m - 100 mΩ 0: 100 m - 1 Ω 1: 1 - 10 Ω 2: 10 - 100 Ω 3: 100 - 1k Ω 4: 1 k - 10k Ω 5: 10 k - 100 kΩ 6: 100 k - 1 MΩ 7: 1 M - 10 MΩ	UINT8	RO	0x00 (0_{dec})

Index 60n2 SAI Inputs Real (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	TC Inputs Ch.1	Maximum subindex	UINT8	RO	0x01 (1_{dec})
60n1:05	Value	Analog input value as REAL.	REAL	RO	0x00 (0_{dec})

5.12.6 Output data

Index 70n0 SAI Range (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
70n0:0	TC Outputs Ch.1	Maximum subindex	UINT8	RO	0x09 (9$_{dec}$)
70n0:01	Disable Autorange	Request to disable Autorange.	INT16	RO	0x0000 (0$_{dec}$)
70n0:04	Start Conversion	Measurement request	INT16	RO	0x0000 (0$_{dec}$)
70n0:05	Mode	Measuring mode request: 0: 4-wire measurement 1: 4-wire measurement, single-shot mode 2: 2-wire measurement 3: 2-wire measurement, single-shot mode	Bit-Field[4]	RO	0x0000 (0$_{dec}$)
70n0:09	Range	Measuring range request: -1: 10 m - 100 mΩ 0: 100 m - 1 Ω 1: 1 - 10 Ω 2: 10 - 100 Ω 3: 100 - 1 kΩ 4: 1 k - 10 kΩ 5: 10 k - 100 kΩ 6: 100 k - 1 MΩ 7: 1 M - 10 MΩ	INT8	RO	0x0000 (0$_{dec}$)

5.12.7 Configuration data (vendor-specific)

Index 80nF SAI Vendor data (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
80nF:0	SAI Vendor data	Maximum subindex	UINT8	RO	0x24 (36$_{dec}$)
80nF:0m (odd)	Range x 2/4W offset	one offset value per measuring range for manufacturer calibration	INT16	RO	0x00 (0$_{dec}$)
80nF:0m (even)	Range x 2/4W gain	one gain value per measuring range for manufacturer calibration 0x10000 corresponds to 1	UINT16	RO	0x10000 (65536$_{dec}$)

5.12.8 Information and diagnostic data

Index 80nE SAI Internal Data (for Ch.1 (n=0); Ch.2 (n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
800E:0	SAI Internal Data	Maximum subindex	UINT8	RO	0x03 (3$_{dec}$)
80nE:01	Raw value	ADC raw value	UINT32	RO	0x00000000 (0$_{dec}$)
80nE:03	Sampling Time	Duration of the last conversion in ms	UINT16	RO	0x0000 (0$_{dec}$)

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2$_{dec}$)
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16$_{dec}$)
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0002 (2$_{dec}$)

Index F008 Code word

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

Index F010Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Module list	UINT8	RO	0x02 (2_{dec})
F010:01	Subindex 001	Profile of the first module	UINT32	RO	0x168 (360_{dec})
F000:02	Subindex 002	Profile of the second module	UINT16	RO	0x168 (360_{dec})

5.12.9 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
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	0x014A1389 (21631881_{dec})

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL3314-0000

Index 1009 Hardware version

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

Index 100A Software version

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

Index 1018 Identity

Index (hex)	Name	Meaning	Data type	Flags	Default
1018:0	Identity	Information for identifying the slave	UINT8	RO	0x04 (4_{dec})
1018:01	Vendor ID	Vendor ID of the EtherCAT slave	UINT32	RO	0x00000002 (2_{dec})
1018:02	Product code	Product code of the EtherCAT slave	UINT32	RO	0x0CF23052 (217198674_{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	0x00100000 (1048576_{dec})
1018:04	Serial number	empty	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

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

Index 160n RxPDO-Map Range (Ch1 n=0; Ch2 n=1)

Index (hex)	Name	Meaning	Data type	Flags	Default
160n:0	RxPDO-Map Ch.1	PDO Mapping RxPDO n	UINT8	RO	0x05 (5_{dec})
160n:01	SubIndex 001	1. PDO Mapping entry (object 0x7000 (SAI Range), entry 0x01 (Disable Autorange))	UINT32	RO	0x7000:01, 1
160n:02	SubIndex 002	2. PDO Mapping entry (Fill Bits)	UINT32	RO	0x0000:00, 2
160n:03	SubIndex 003	3. PDO Mapping entry (object 0x7000 (SAI Range), entry 0x04 (Start conversion))	UINT32	RO	0x7000:04, 1
160n:04	SubIndex 004	4. PDO Mapping entry (object 0x7000 (SAI Range), entry 0x05 (Mode))	UINT32	RO	0x7000:05, 4
160n:05	SubIndex 005	5. PDO Mapping entry (object 0x7000 (SAI Range), entry 0x09 (Range))	UINT32	RO	0x7000:09, 8

Index 1A0n TxPDO-Map Inputs (Ch1 n=0; Ch2 n=3)

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0n:0	TxPDO-ParCh.1	PDO Parameter TxPDO 1	UINT8	RO	0x0C (12_{dec})
1A0n:01	SubIndex 001	1. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x01 (Underrange))	UINT32	RO	0x6000:01, 1
1A0n:02	SubIndex 002	2. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x02 (Overrange))	UINT32	RO	0x6000:02, 1
1A0n:03	SubIndex 003	3. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x03 (Extended Range))	UINT32	RO	0x6000:03, 1
1A0n:04	SubIndex 004	4. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x04 (Data Invalid))	UINT32	RO	0x6000:04, 1
1A0n:05	SubIndex 005	5. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x05 (Range Invalid))	UINT32	RO	0x6000:05, 1
1A0n:06	SubIndex 006	6. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x06 (Autorange disabled))	UINT32	RO	0x6000:06, 1
1A0n:07	SubIndex 007	7. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x07 (Error))	UINT32	RO	0x6000:07, 1
1A0n:08	SubIndex 008	8. PDO Mapping entry (Fill Bits)	UINT32	RO	0x0000:00, 1
1A0n:09	SubIndex 009	9. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x09 (Steady state))	UINT32	RO	0x6000:07, 1
1A0n:0A	SubIndex 010	10. PDO Mapping entry (Fill Bits)	UINT32	RO	0x0000:00, 5
1A0n:0B	SubIndex 011	11. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x0F (TxPDO State))	UINT32	RO	0x6000:0F, 1
1A0n:0C	SubIndex 012	12. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x10 (TxPDO Toogle))	UINT32	RO	0x6000:10, 1
1A0n:0D	SubIndex 013	13. PDO Mapping Entry (object 0x6000 (SAI Inputs), entry 0x11 (INT Value))	UINT32	RO	0x6000:11, 32

Index 1A0n TxPDO-Map Inputs Real (Ch1 n=2; Ch2 n=5)

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0n:0	TxPDO-ParCh.1	PDO Parameter TxPDO 1	UINT8	RO	0x01 (1_{dec})
1A0n:01	SubIndex 001	1. PDO Mapping Entry (object 0x6002 (SAI Inputs Real), entry 0x01 (Range))	UINT32	RO	0x6002:01, 32

Index 1A0n TxPDO-Map Range (Ch1 n=1; Ch2 n=4)

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0n:0	TxPDO-ParCh.1	PDO Parameter TxPDO 1	UINT8	RO	0x03 (3_{dec})
1A0n:01	SubIndex 001	1. PDO Mapping Entry (Fill Bits)	UINT32	RO	0x0000:00, 4
1A0n:02	SubIndex 002	2. PDO Mapping Entry (object 0x6001 (SAI Range), entry 0x05 (Mode))	UINT32	RO	0x6001:05, 4
1A0n:03	SubIndex 003	3. PDO Mapping Entry (object 0x6001 (SAI Range), entry 0x09 (Range))	UINT32	RO	0x6001:09, 8

Index 1C00 Sync manager type

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

Index 1C12 RxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C12:0	RxPDO assign	PDO Assign Outputs	UINT8	RW	0x02 (2_{dec})
1C12:01	Subindex 001	1. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1600 (5632_{dec})
1C12:02	Subindex 002	2. allocated RxPDO (contains the index of the associated RxPDO mapping object)	UINT16	RW	0x1601 (5633_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x06 (6_{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})
1C13:06	Subindex 006	6. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A05 (6661_{dec})

Index 1C32 SM output parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C32:0	SM output parameter	Synchronization parameters for the outputs	UINT8	RO	0x07 (7_{dec})
1C32:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 2 Event • 2: DC-Mode - Synchron with SYNC0 Event • 3: DC-Mode - Synchron with SYNC1 Event	UINT16	RW	0x0000 (0_{dec})
1C32:02	Cycle time	Cycle time (in ns): • 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})
1C32:03	Shift time	Time between SYNC0 event and output of the outputs (in ns, DC mode only)	UINT32	RW	0x00000000 (0_{dec})
1C32:04	Sync modes supported	Supported synchronization modes: • Bit 0 = 1: free run is supported • Bit 1 = 1: Synchron with SM 2 event is supported • Bit 2-3 = 01: DC mode is supported • Bit 4-5 = 10: Output shift with SYNC1 event (only DC mode) • Bit 14 = 1: dynamic times (measurement through writing of 0x1C32:08 [▸ 174])	UINT16	RO	0x8007 (32775_{dec})
1C32:05	Minimum cycle time	Minimum cycle time (in ns)	UINT32	RO	0x00000000 (0_{dec})
1C32:06	Calc and copy time	Minimum time between SYNC0 and SYNC1 event (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:08	Command	• 0: Measurement of the local cycle time is stopped • 1: Measurement of the local cycle time is started The entries 0x1C32:03 [▸ 174], 0x1C32:05 [▸ 174], 0x1C32:06 [▸ 174], 0x1C32:09 [▸ 174], 0x1C33:03 [▸ 175], 0x1C33:06 [▸ 174], 0x1C33:09 [▸ 175] are updated with the maximum measured values. For a subsequent measurement the measured values are reset	UINT16	RW	0x0000 (0_{dec})
1C32:09	Delay time	Time between SYNC1 event and output of the outputs (in ns, DC mode only)	UINT32	RO	0x00000000 (0_{dec})
1C32:0B	SM event missed counter	Number of missed SM events in OPERATIONAL (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:0C	Cycle exceeded counter	Number of occasions the cycle time was exceeded in OPERATIONAL (cycle was not completed in time or the next cycle began too early)	UINT16	RO	0x0000 (0_{dec})
1C32:0D	Shift too short counter	Number of occasions that the interval between SYNC0 and SYNC1 event was too short (DC mode only)	UINT16	RO	0x0000 (0_{dec})
1C32:20	Sync error	The synchronization was not correct in the last cycle (outputs were output too late; DC mode only)	BOOLEAN	RO	0x00 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x07 (7$_{dec}$)
1C33:01	Sync mode	Current synchronization mode: • 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	as 0x1C32:02 [▶ 174]	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: • 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 0x1C32:08 [▶ 174] or 0x1C33:08 [▶ 175])	UINT16	RO	0x8007 (32775$_{dec}$)
1C33:05	Minimum cycle time	as 0x1C32:05 [▶ 174]	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	as 0x1C32:08 [▶ 174]	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	as 0x1C32:11 [▶ 174]	UINT16	RO	0x0000 (0$_{dec}$)
1C33:0C	Cycle exceeded counter	as 0x1C32:12 [▶ 174]	UINT16	RO	0x0000 (0$_{dec}$)
1C33:0D	Shift too short counter	as 1C32:13 [▶ 174]	UINT16	RO	0x0000 (0$_{dec}$)
1C33:20	Sync error	as 0x1C32:32 [▶ 174]	BOOLEAN	RO	0x00 (0$_{dec}$)

6 Appendix

6.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

6.2 Firmware compatibility

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

Note

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

NOTICE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 178]. If a device is placed in BOOTSTRAP mode for a firmware update, it does not check when downloading whether the new firmware is suitable. This can result in damage to the device! Therefore, always make sure that the firmware is suitable for the hardware version!

EL3692			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
01 - 03	01	EL3692-0000-0016	2011/02
	02		2011/02
	03		2011/02
03 - 10	04	EL3692-0000-0017	2011/06
		EL3692-0000-0018	2011/11
	05	EL3692-0000-0019	2012/10
	06	EL3692-0000-0020	2014/08
11 - 12*	06*	EL3692-0000-0021	2017/07

EL3692-0020			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
09 – 12*	05	EL3692-0020-0016	2013/12
	06*	EL3692-0020-0020	2014/08
		EL3692-0020-0021	2017/07

EL3692-0030			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Date of release**
12*	06*	EL3692-0030-0021	2020/01

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.

6.3 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.

NOTICE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website.

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

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

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

Storage locations

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

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

NOTICE

Application-specific writing of the ESI-EEPROM

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

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_REV0016_SW01.efw

- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

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

NOTICE

Risk of damage to the device!

✓ Note the following when downloading new device files

a) Firmware downloads to an EtherCAT device must not be interrupted

b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.

c) The power supply must adequately dimensioned. The signal level must meet the specification.

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

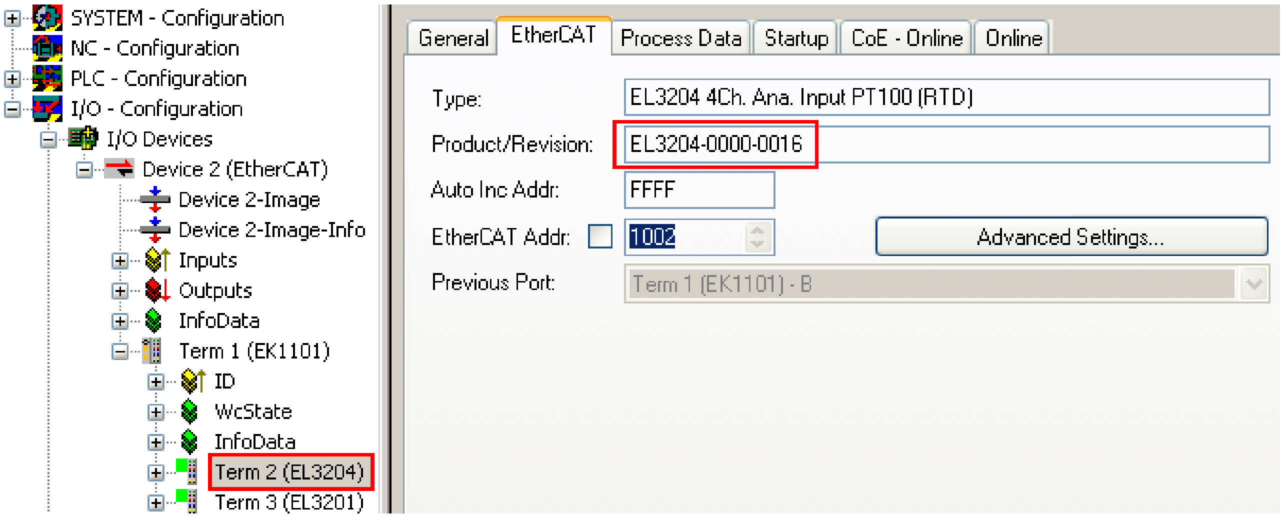
6.3.1 Device description ESI file/XML

NOTICE

Attention regarding update of the ESI description/EEPROM

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

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



Type: EL3204 4Ch. Ana. Input PT100 (RTD)

Product/Revision: EL3204-0000-0016

Auto Inc Addr: FFFF

EtherCAT Addr: □ 1002

Previous Port: Term 1 (EK1101) - B

Fig. 171: Device identifier consisting of name EL3204-0000 and revision -0016

The configured identifier must be compatible with the actual device description used as hardware, i.e. the description which the slave has loaded on start-up (in this case EL3204). Normally the configured revision must be the same or lower than that actually present in the terminal network.

For further information on this, please refer to the EtherCAT system documentation.

Update of XML/ESI description

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

Display of ESI slave identifier

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

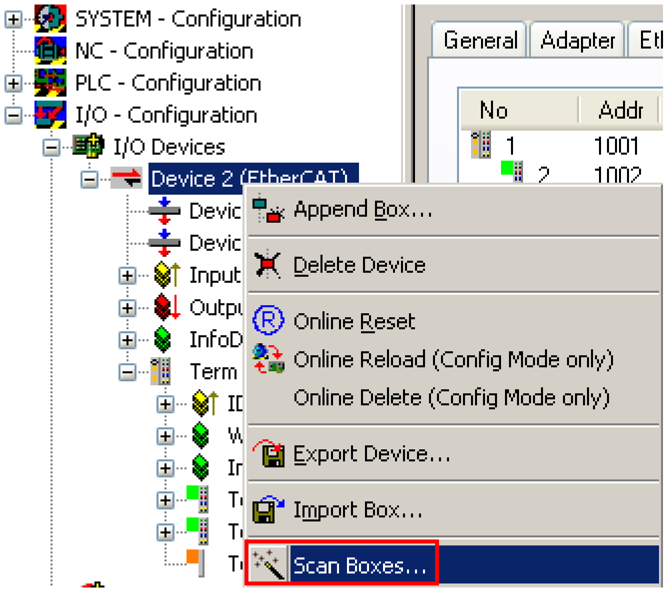
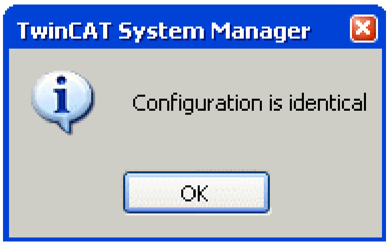


Fig. 172: Scan the subordinate field by right-clicking on the EtherCAT device

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



Configuration is identical

Fig. 173: Configuration is identical

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

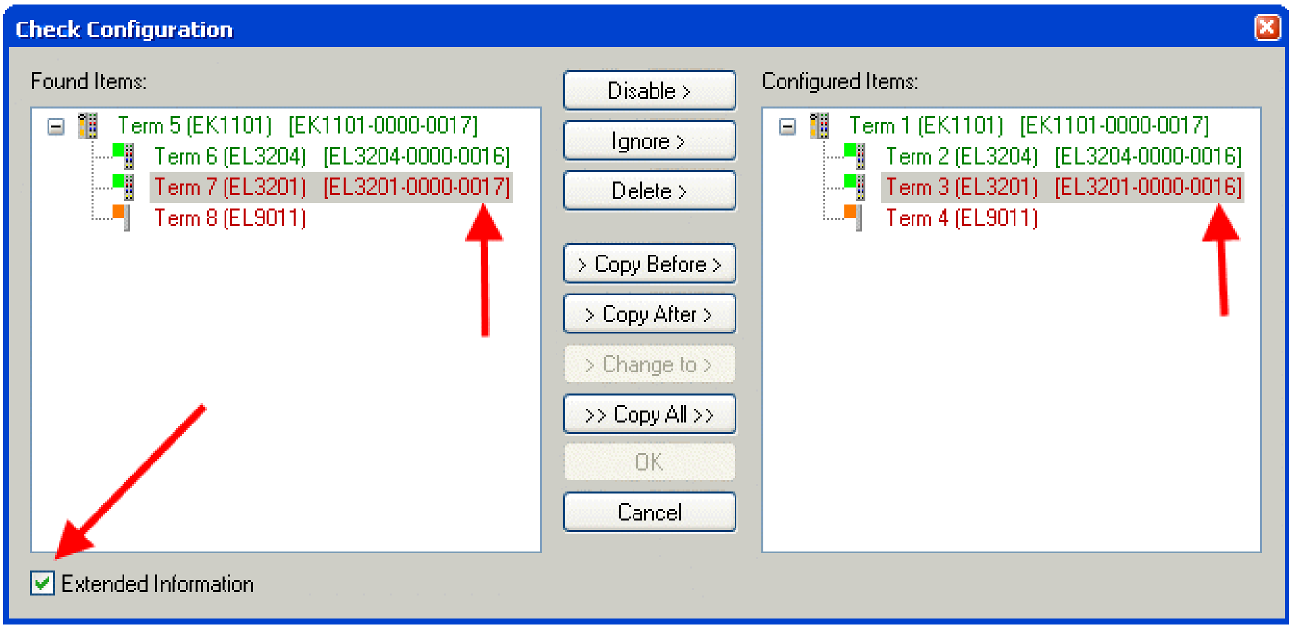

Check Configuration
Found Items:
Term 5 (EK1101) [EK1101-0000-0017]
Term 6 (EL3204) [EL3204-0000-0016]
Term 7 (EL3201) [EL3201-0000-0017]
Term 8 (EL9011)
Disable >
Ignore >
Delete >
> Copy Before >
> Copy After >
> Change to >
>> Copy All >>
OK
Cancel
Configured Items:
Term 1 (EK1101) [EK1101-0000-0017]
Term 2 (EL3204) [EL3204-0000-0016]
Term 3 (EL3201) [EL3201-0000-0016]
Term 4 (EL9011)
☑ Extended Information

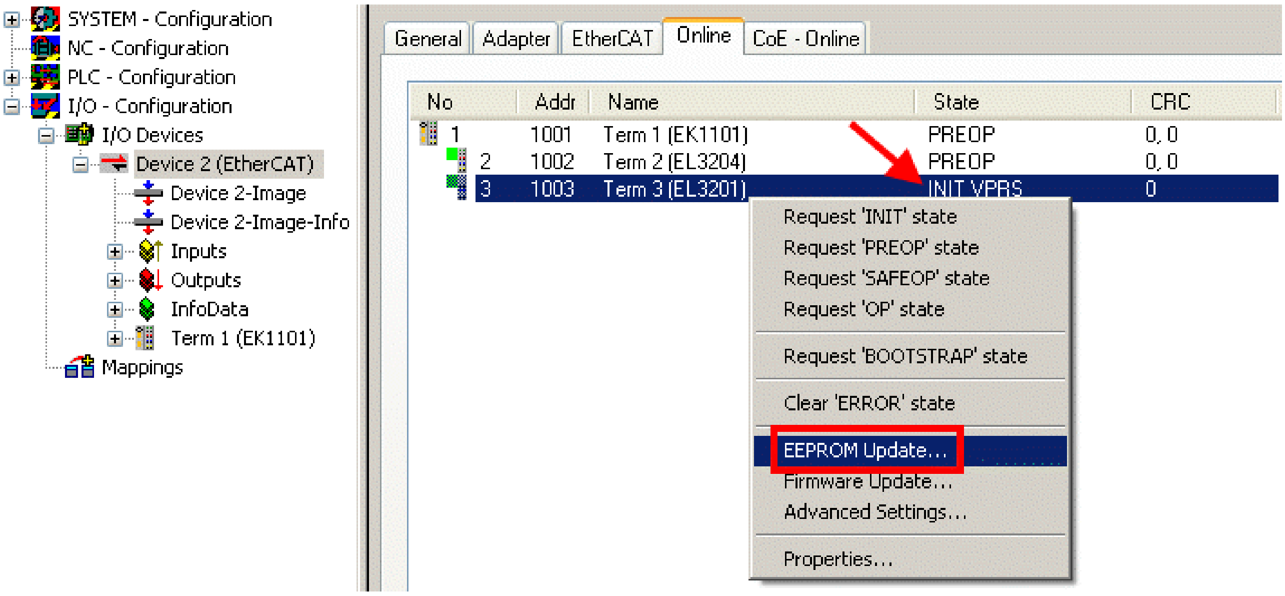
Fig. 174: Change dialog

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:

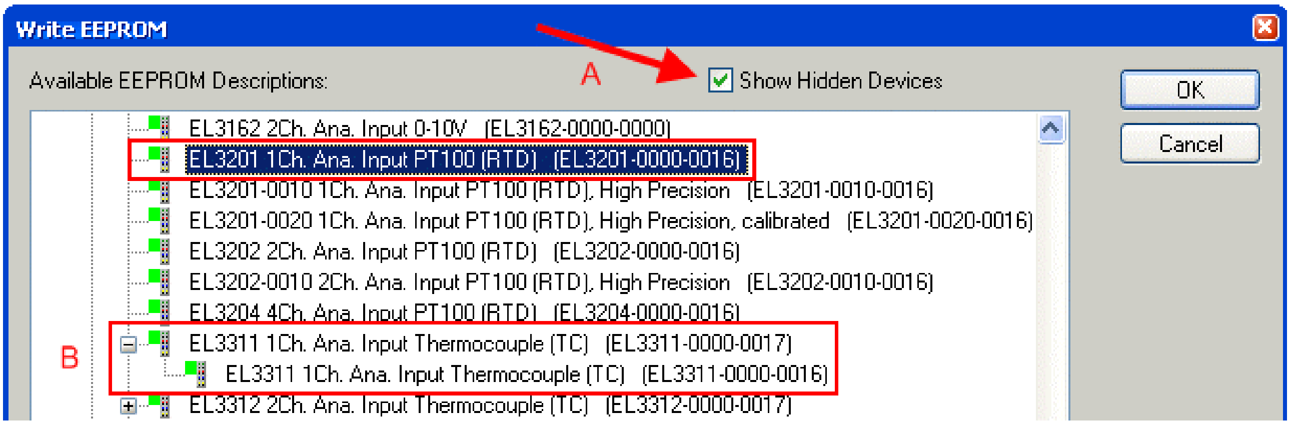
- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*



No	Addr	Name	State	CRC
1	1001	Term 1 (EK1101)	PREOP	0, 0
2	1002	Term 2 (EL3204)	PREOP	0, 0
3	1003	Term 3 (EL3201)	INIT VPRS	0

Fig. 175: 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.



EL3162 2Ch. Ana. Input 0-10V (EL3162-0000-0000)
EL3201 1Ch. Ana. Input PT100 (RTD) (EL3201-0000-0016)
EL3201-0010 1Ch. Ana. Input PT100 (RTD), High Precision (EL3201-0010-0016)
EL3201-0020 1Ch. Ana. Input PT100 (RTD), High Precision, calibrated (EL3201-0020-0016)
EL3202 2Ch. Ana. Input PT100 (RTD) (EL3202-0000-0016)
EL3202-0010 2Ch. Ana. Input PT100 (RTD), High Precision (EL3202-0010-0016)
EL3204 4Ch. Ana. Input PT100 (RTD) (EL3204-0000-0016)
EL3311 1Ch. Ana. Input Thermocouple (TC) (EL3311-0000-0017)
EL3311 1Ch. Ana. Input Thermocouple (TC) (EL3311-0000-0016)
EL3312 2Ch. Ana. Input Thermocouple (TC) (EL3312-0000-0017)

Fig. 176: Selecting the new ESI

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

The change only takes effect after a restart.

Most EtherCAT devices read a modified ESI description immediately or after startup from the INIT. Some communication settings such as distributed clocks are only read during power-on. The EtherCAT slave therefore has to be switched off briefly in order for the change to take effect.

6.3.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

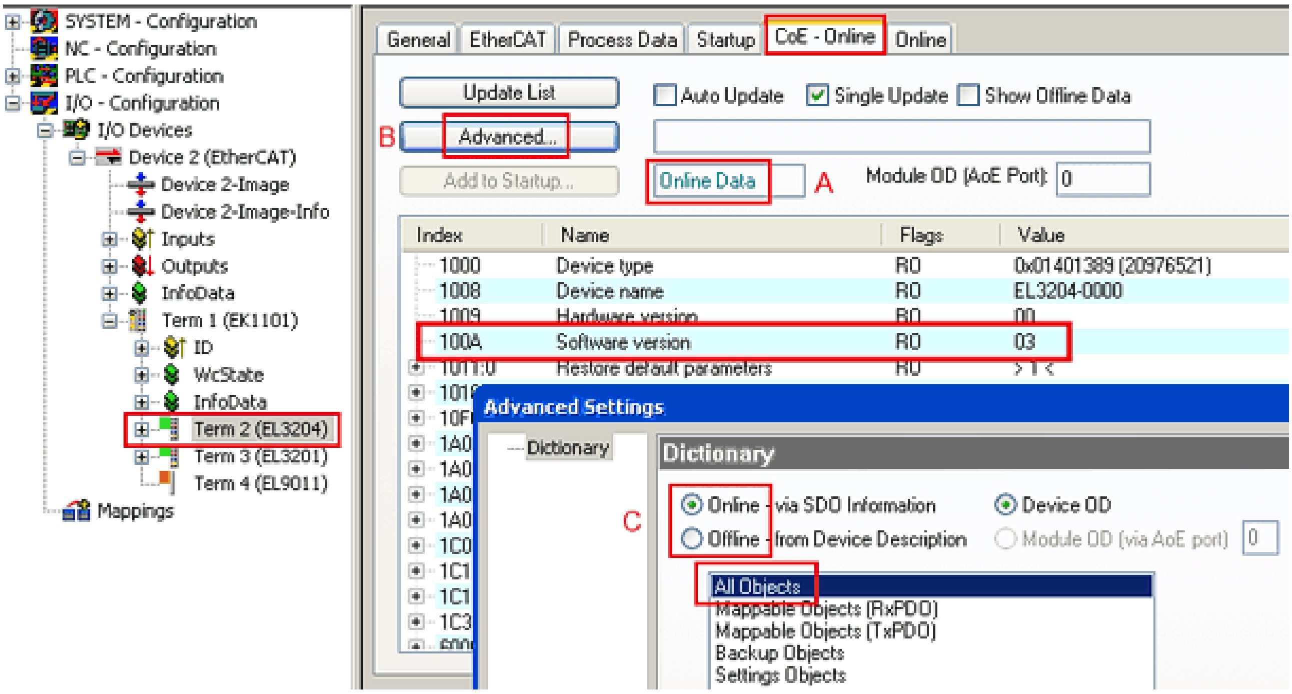
The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

CoE Online and Offline CoE

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.



☐ Auto Update ☑ Single Update ☐ Show Offline Data

Online Data

Module OD (AoE Port): 0

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	RO	> 1 <
1018			
10F			
1A0			
1A0			
1A0			
1A0			
1C0			
1C1			
1C1			
1C3			
600			

● Online - via SDO Information ○ Offline - from Device Description

● Device OD ○ Module OD (via AoE port) 0

All Objects
Mappable Objects (RxPDO)
Mappable Objects (TxPDO)
Backup Objects
Settings Objects

Fig. 177: Display of EL3204 firmware version

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

6.3.3 Updating controller firmware *.efw

CoE directory

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

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

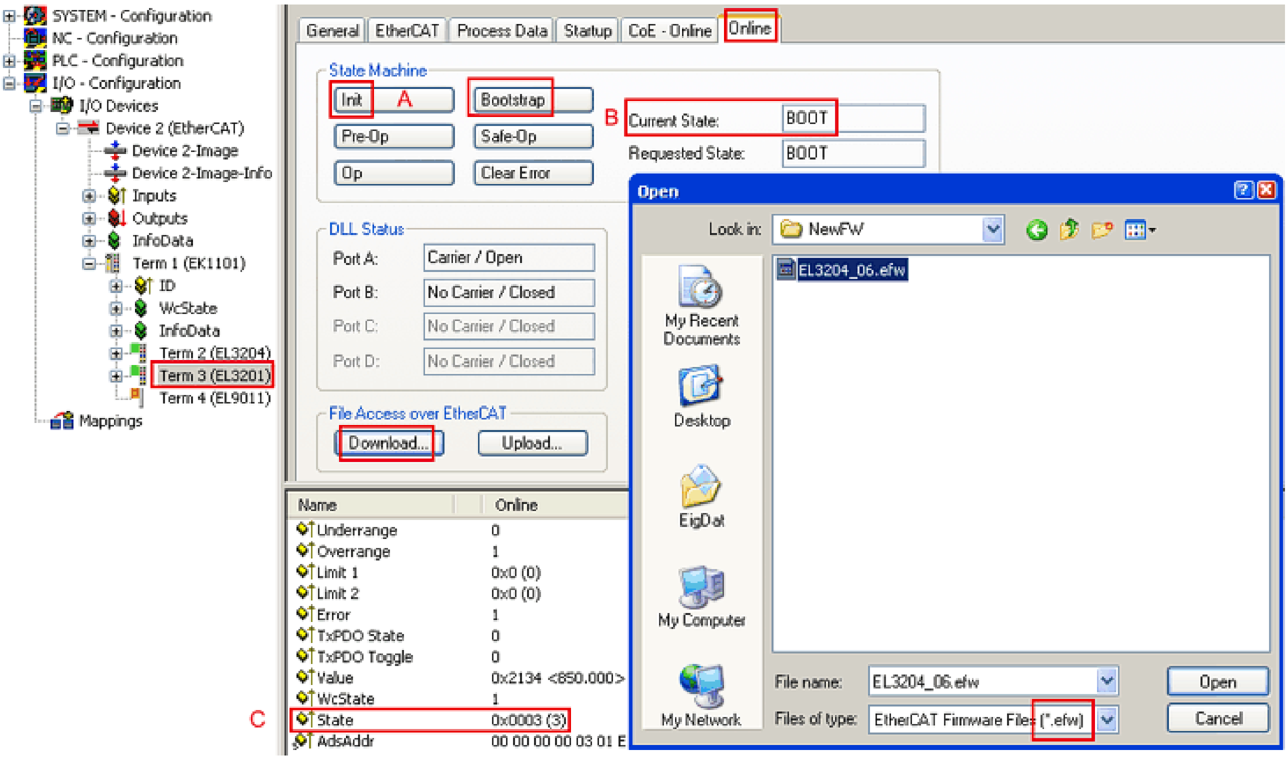
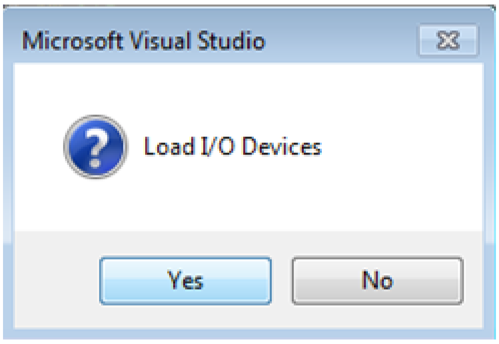


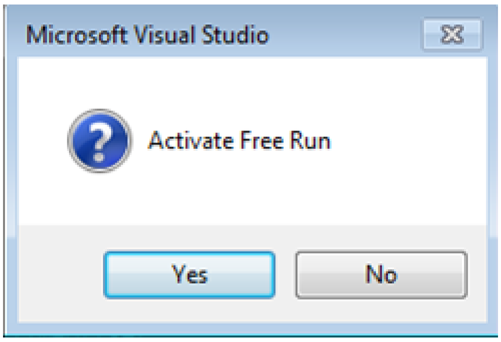
Fig. 178: Firmware Update

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.

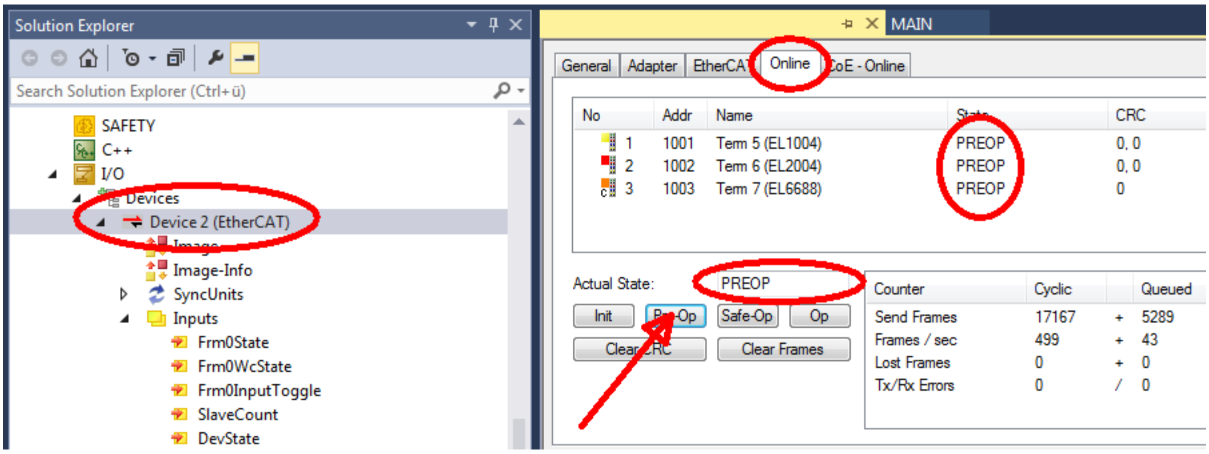

Microsoft Visual Studio

Load I/O Devices


Microsoft Visual Studio

Activate Free Run

- Switch EtherCAT Master to PreOP



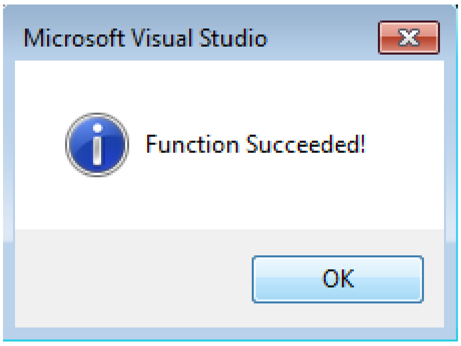
No	Addr	Name	State	CRC
1	1001	Term 5 (EL1004)	PREOP	0, 0
2	1002	Term 6 (EL2004)	PREOP	0, 0
3	1003	Term 7 (EL6688)	PREOP	0

Actual State: PREOP

Counter	Cyclic		Queued
Send Frames	17167	+	5289
Frames / sec	499	+	43
Lost Frames	0	+	0
Tx/Rx Errors	0	/	0

- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A password will not be necessary usually.



Microsoft Visual Studio

Function Succeeded!

- 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.

6.3.4 FPGA firmware *.rbf

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

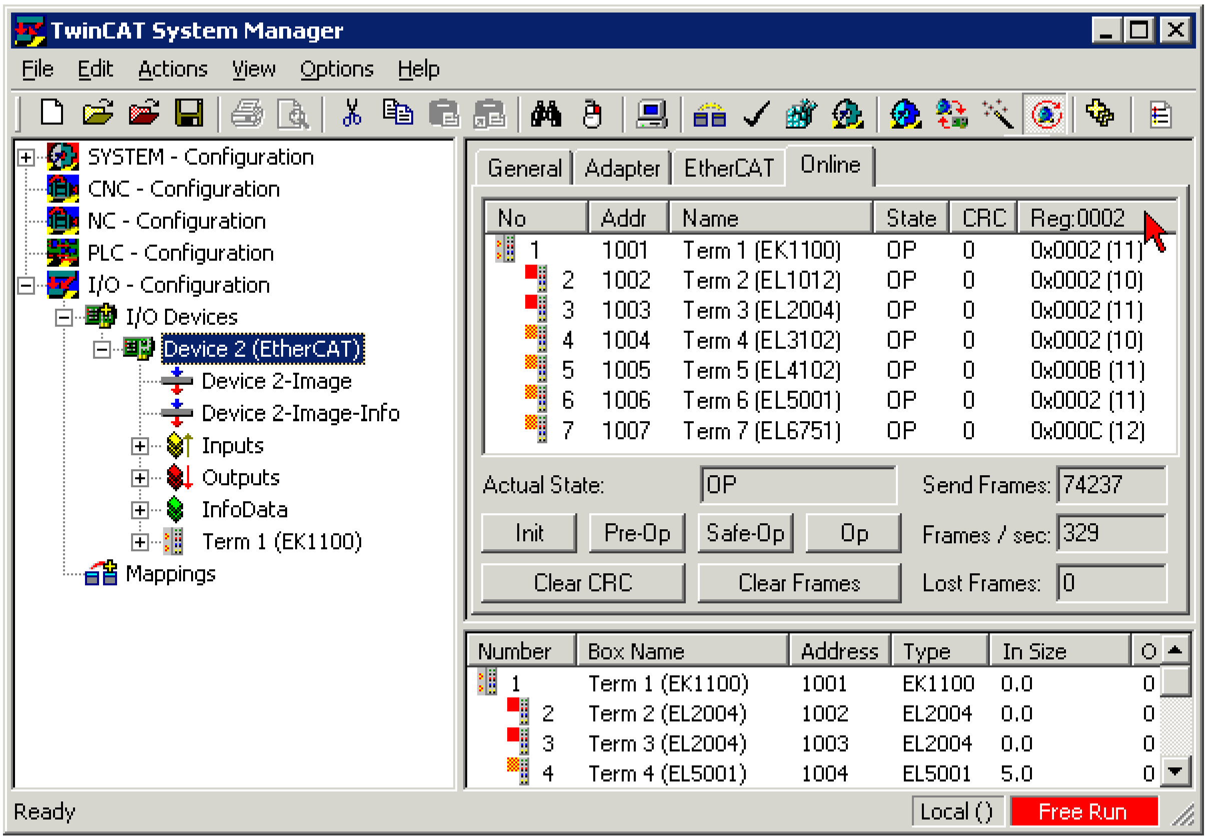
- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

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

Determining the version via the 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.



No	Addr	Name	State	CRC	Reg:0002
1	1001	Term 1 (EK1100)	OP	0	0x0002 (11)
2	1002	Term 2 (EL1012)	OP	0	0x0002 (10)
3	1003	Term 3 (EL2004)	OP	0	0x0002 (11)
4	1004	Term 4 (EL3102)	OP	0	0x0002 (10)
5	1005	Term 5 (EL4102)	OP	0	0x000B (11)
6	1006	Term 6 (EL5001)	OP	0	0x0002 (11)
7	1007	Term 7 (EL6751)	OP	0	0x000C (12)

Actual State: OP
Send Frames: 74237
Frames / sec: 329
Lost Frames: 0

Number	Box Name	Address	Type	In Size	O
1	Term 1 (EK1100)	1001	EK1100	0.0	0
2	Term 2 (EL2004)	1002	EL2004	0.0	0
3	Term 3 (EL2004)	1003	EL2004	0.0	0
4	Term 4 (EL5001)	1004	EL5001	5.0	0

Fig. 179: FPGA firmware version definition

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

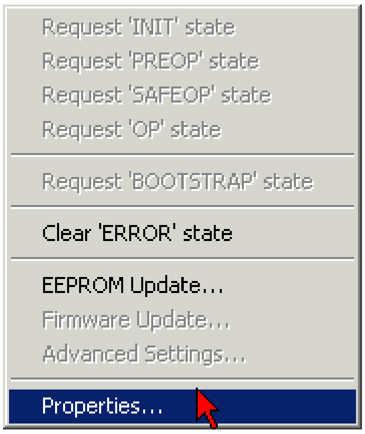
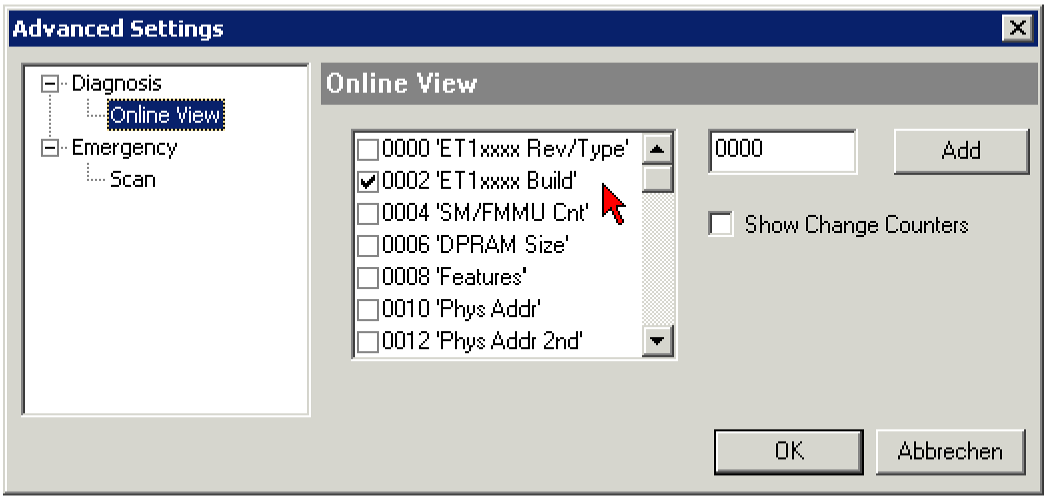


Fig. 180: Context menu *Properties*

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.



□ 0000 'ET1xxxx Rev/Type'
☑ 0002 'ET1xxxx Build'
□ 0004 'SM/FMMU Cnt'
□ 0006 'DPRAM Size'
□ 0008 'Features'
□ 0010 'Phys Addr'
□ 0012 'Phys Addr 2nd'

0000

□ Show Change Counters

Fig. 181: 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.

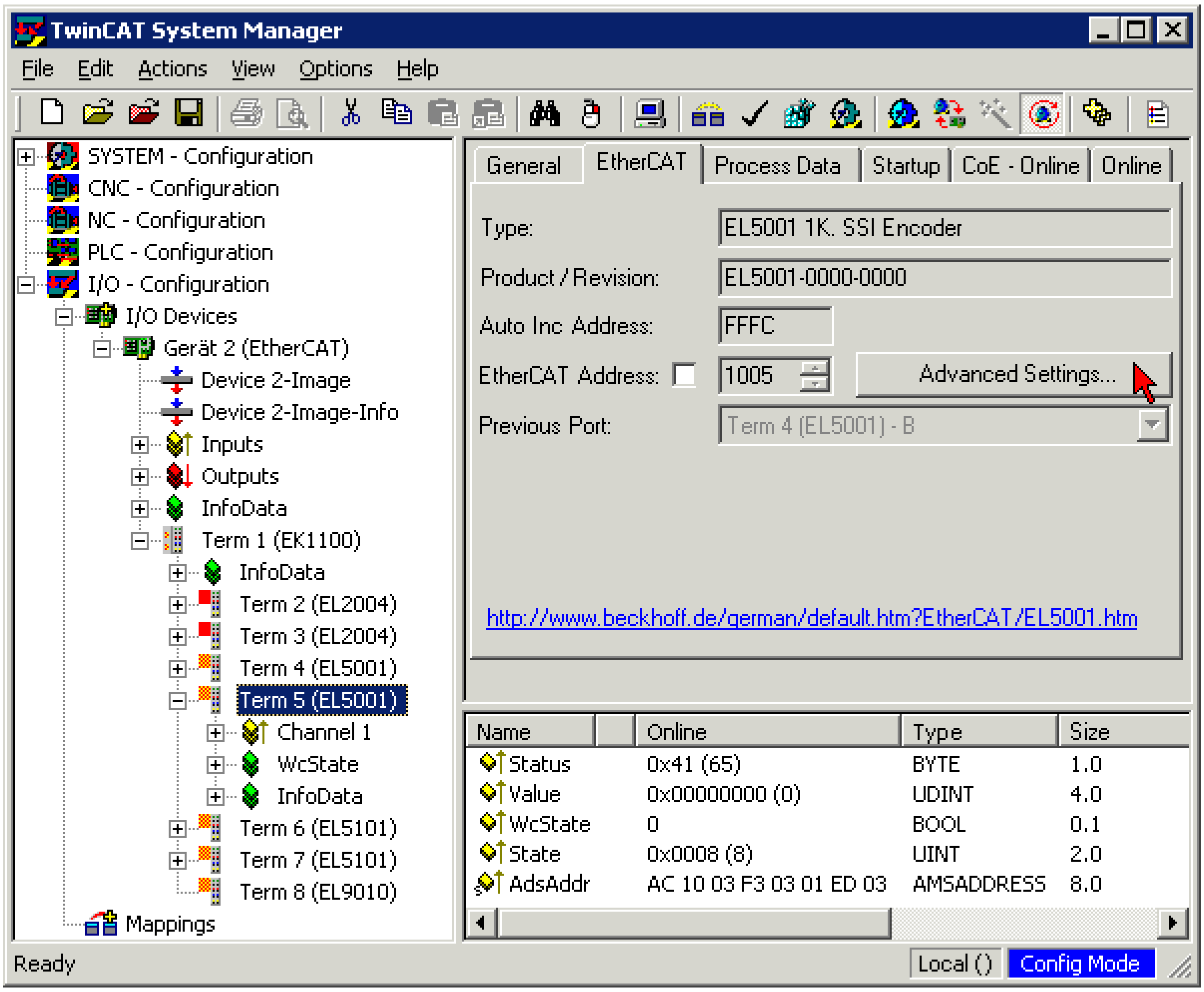
Older firmware versions can only be updated by the manufacturer!

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:



Type: EL5001 1K. SSI Encoder

Product / Revision: EL5001-0000-0000

Auto Inc Address: FFFC

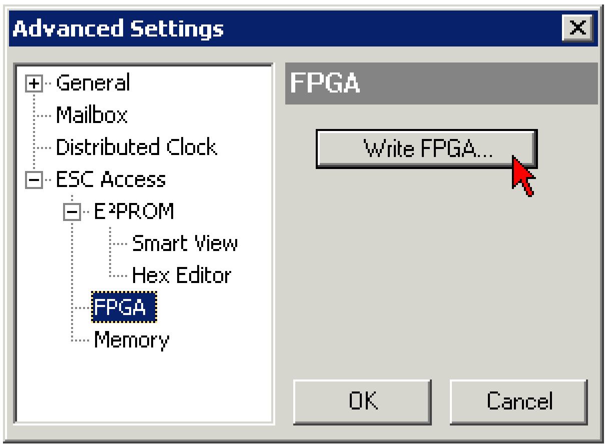
EtherCAT Address: □ 1005

Previous Port: Term 4 (EL5001) - B

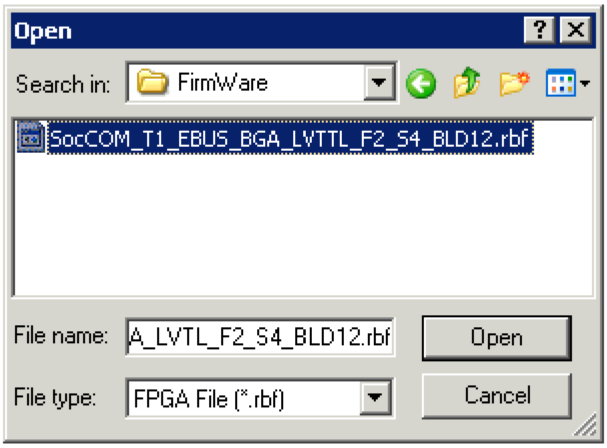
http://www.beckhoff.de/german/default.htm?EtherCAT/EL5001.htm

Name	Online	Type	Size
Status	0x41 (65)	BYTE	1.0
Value	0x00000000 (0)	UDINT	4.0
WcState	0	BOOL	0.1
State	0x0008 (8)	UINT	2.0
AdsAddr	AC 10 03 F3 03 01 ED 03	AMSADDRESS	8.0

- 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:



Open

Search in: FirmWare

SocCOM_T1_EBUS_BGA_LVTTL_F2_S4_BLD12.rbf

File name: A_LVTL_F2_S4_BLD12.rbf

File type: FPGA File (*.rbf)

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

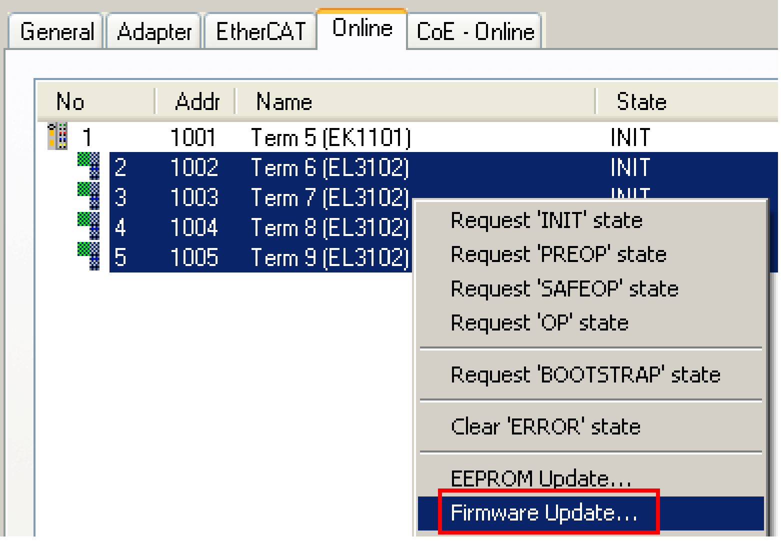
NOTICE

Risk of damage to the device!

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

6.3.5 Simultaneous updating of several EtherCAT devices

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



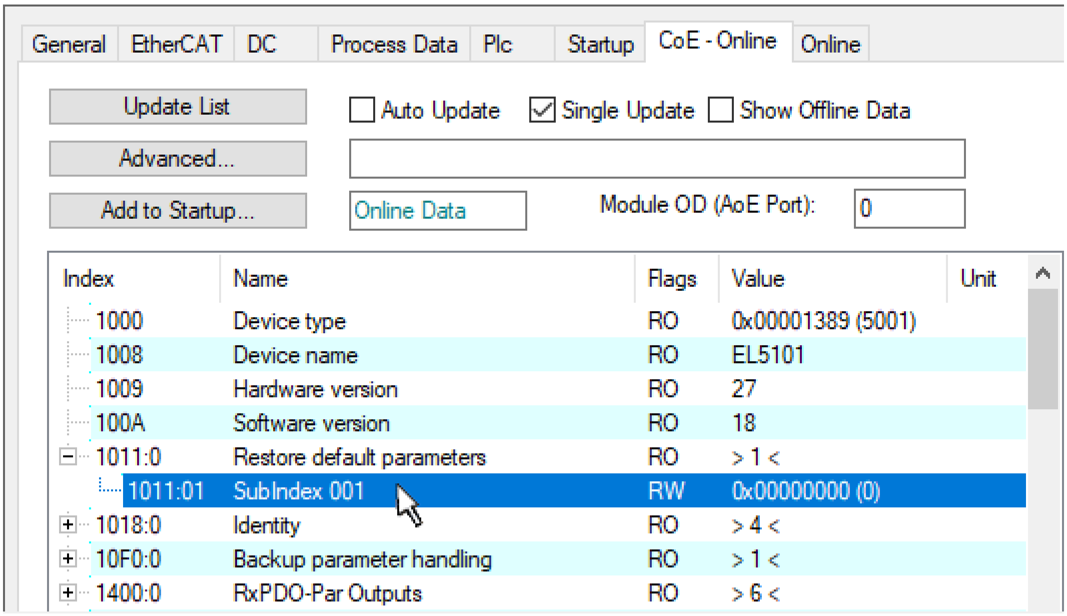
No	Addr	Name	State
1	1001	Term 5 (EK1101)	INIT
2	1002	Term 6 (EL3102)	INIT
3	1003	Term 7 (EL3102)	INIT
4	1004	Term 8 (EL3102)	
5	1005	Term 9 (EL3102)	

Fig. 182: Multiple selection and firmware update

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

6.4 Restoring the delivery state

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

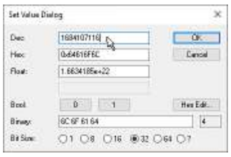


□ Auto Update ☑ Single Update □ Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value	Unit
1000	Device type	RO	0x00001389 (5001)	
1008	Device name	RO	EL5101	
1009	Hardware version	RO	27	
100A	Software version	RO	18	
1011:0	Restore default parameters	RO	> 1 <	
1011:01	SubIndex 001	RW	0x00000000 (0)	
1018:0	Identity	RO	> 4 <	
10F0:0	Backup parameter handling	RO	> 1 <	
1400:0	RxPDO-Par Outputs	RO	> 6 <	

Fig. 183: Selecting the *Restore default parameters* PDO



Dec: 1684107116

Hex: 0x64616F6C

Float: 1.6634185e+22

Binary: 6C 6F 61 64 4

Bit Size: ○ 1 ○ 8 ○ 16 ● 32 ○ 64 ○ ?

Fig. 184: Entering a restore value in the Set Value dialog

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

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

All backup objects are reset to the delivery state.

Alternative restore value

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

An incorrect entry for the restore value has no effect.

6.5 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Support

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

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

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

Service

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

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

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

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

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

More Information:
www.beckhoff.com/EL3692

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