BECKHOFF New Automation Technology

Documentation | EN

EL3632

2 Channel Analog Input Terminal for Condition Monitoring (IEPE)

2022-01-06 | Version: 3.0

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

© Beckhoff Automation GmbH & Co. KG, Germany.
The reproduction, distribution and utilization of this document as well as the communication of its contents to others without express authorization are prohibited.
Offenders will be held liable for the payment of damages. All rights reserved in the event of the grant of a patent, utility model or design.

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of instructions

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

⚠ DANGER
Serious risk of injury!
Failure to follow this safety instruction directly endangers the life and health of persons.

⚠ WARNING
Risk of injury!
Failure to follow this safety instruction endangers the life and health of persons.

⚠ CAUTION
Personal injuries!
Failure to follow this safety instruction can lead to injuries to persons.

NOTE
Damage to environment/equipment or data loss
Failure to follow this instruction can lead to environmental damage, equipment damage or data loss.

ℹ **Tip or pointer**

This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
3.0	• Chapter "Technical data" updated • Update structure
2.9	• Chapter "Technical data" updated • Chapter "Commissioning” updated • Update structure
2.8	• Chapter "Diag Messages" added • Update structure • Update revision status
2.7	• Update chapter "LEDs and connection”
2.6	• Update chapter "Technology" • Update structure • Update revision status
2.5	• Update chapter "Technical data" • Addenda chapter "Instructions for ESD protection" • Update chapter "Notices on Analog specification" • Addenda chapter "Installation instructions for enhanced mechanical load capacity"
2.4	• Update chapter "LEDs and connection" • Addenda chapter “Notices on analog specifications”
2.3	• Update chapter "Notes on the documentation" • Correction of Technical data • Update chapter "TwinCAT 2.1x" -> "TwinCAT Development Environment" and "TwinCAT Quick Start" • Update revision status
2.2	• Section “Oversampling terminals and TwinCAT Scope” added
2.1	• Update chapter "Object description" • Update chapter "Technology" • Update revision status
2.0	• Migration • Update structure • Update revision status
2.0	• Migration • Update structure • Update revision status
1.6	• Update chapter "Object description" • Update chapter "Firmware status" • Update structure
1.5	• Update chapter "Object description" • Update chapter "Technical data" • Update chapter "Firmware status" • Update structure
1.4	• Update Technical data
1.3	• Update chapter "Technology" • Update chapter "Firmware status"
1.2	• Addenda chapter "Technology"
1.1	• Addenda chapter "Technical data"
1.0	• Amendments • 1. Publication

Version	Comment
0.1	• Preliminary documentation for EL3632

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

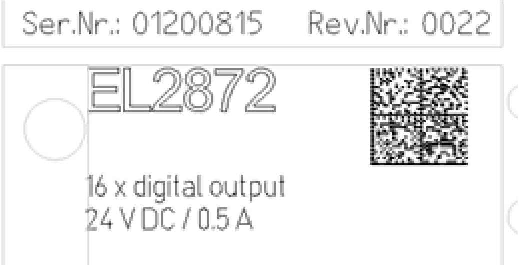


Fig. 1: EL2872 with revision 0022 and serial number 01200815

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.

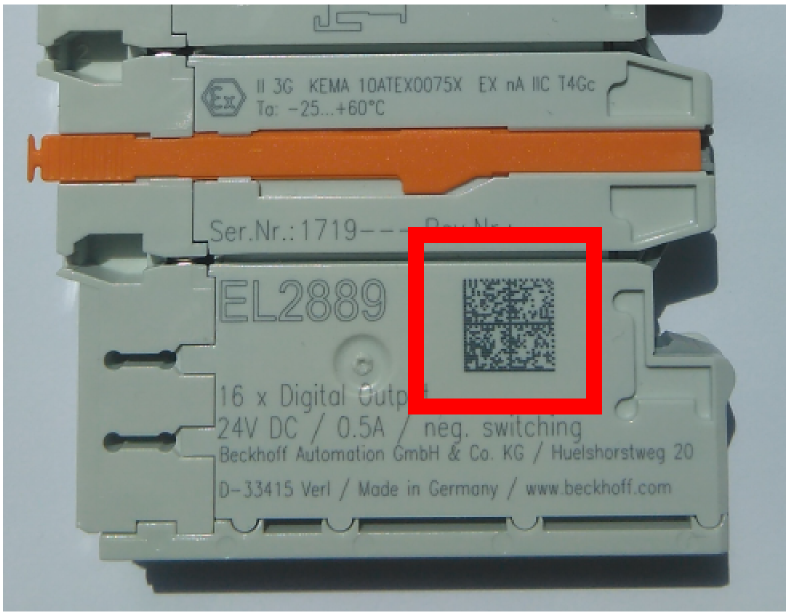

II 3G KEMA 10ATEX0075X EX nA IIC T4Gc
Ta: -25...+60°C
Ser.Nr.:1719---
EL2889
16 x Digital Output
24V DC / 0.5A / neg. switching
Beckhoff Automation GmbH & Co. KG / Huelshorstweg 20
D-33415 Verl / Made in Germany / www.beckhoff.com

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:

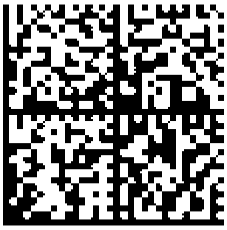


Fig. 3: Example DMC **1P**072222**S**BTNk4p562d7**1K**EL1809 **Q**1 **51S**678294

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.

NOTE

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

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.

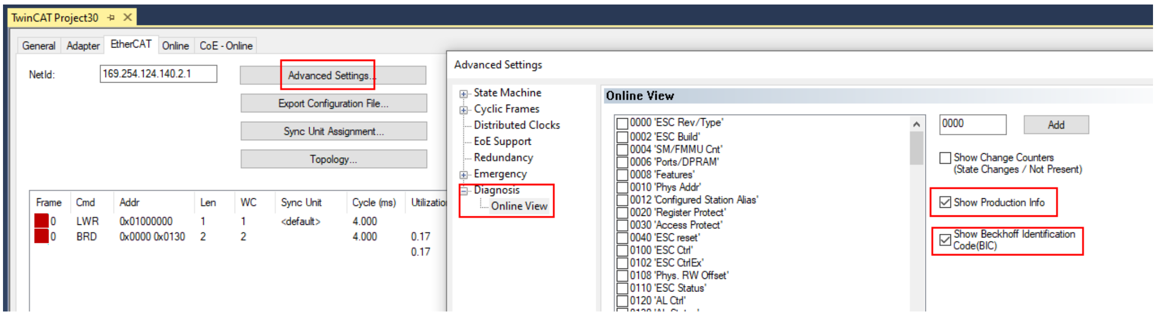
EtherCAT devices (IP20, IP67)

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, boxes) 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 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'

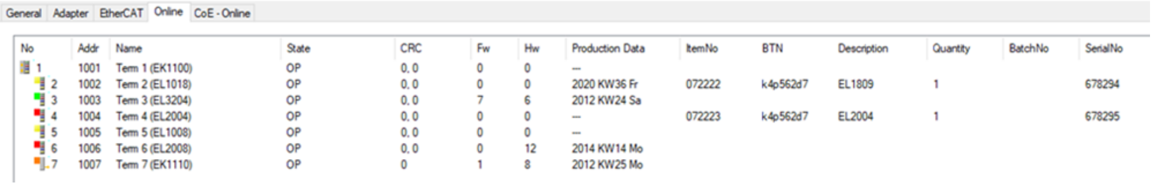
0000

□ Show Change Counters (State Changes / Not Present)

☑ Show Production Info

☑ Show Beckhoff Identification Code(BIC)

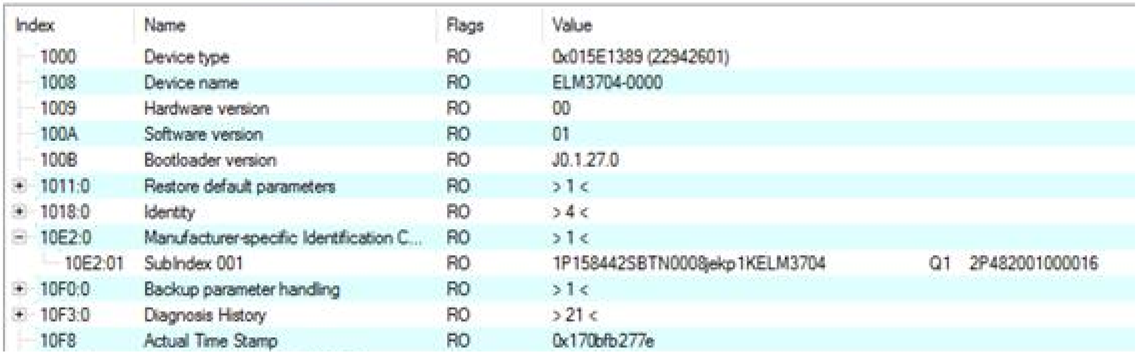
    - The BTN and its contents are then displayed:



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".
- 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 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.

- 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

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

2 Product description

2.1 Introduction

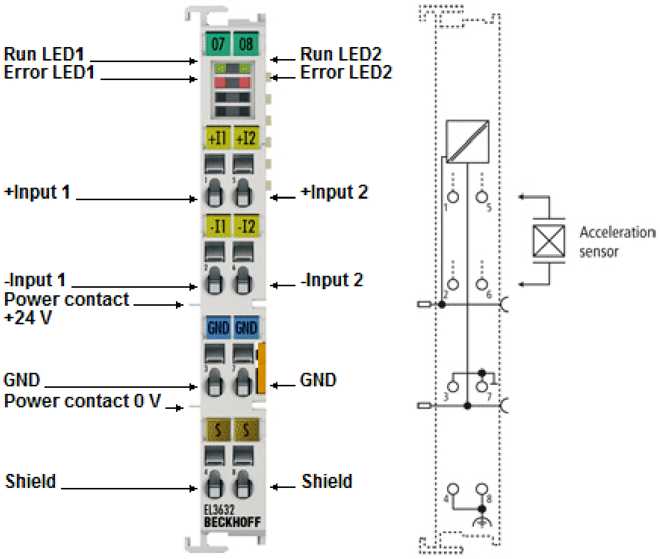


Fig. 4: EL3632

2-channel analog input terminal for Condition Monitoring (IEPE)

Accelerometers with IEPE interface can be directly connected to the EL3632 EtherCAT Terminal. The measuring signals are analyzed on the PC via the TwinCAT library. This enables all benefits of the PC platform, such as performance and flexibility, to be fully utilized. Alternatively, the evaluation can also be done using the user software. The terminal can be adapted to individual requirements through configurable filters and supply currents.

A galvanically isolated measurement configuration can be achieved using the EL9560. Through interfacing via EtherCAT and support of the distributed clocks function the measurement results – and any detected defects – can be precisely allocated to an axis position.

Quick links

- EtherCAT basic function principles
- IEPE - Technology, areas of application, function [▸ 18]
- Basic function principles EL3632 [▸ 126]

2.2 Technical data

Technical data	EL3632
Number of inputs	2
Measuring range	preset ±5 V up to 25 kHz, ±250 mV up to 10 Hz
Supply voltage	24 V_{DC} via power contacts
U_{EXCITE}	U_V -1 V typ.
Sensor state monitoring	yes, through monitoring of the bias voltage
Supply current I_{EXCITE}	typ. 2 mA/4 mA/8 mA (separately adjustable for both channels)
Input filter limit frequency	analog parameterisable 5th order low-pass filter up to 25 KHz typ. 0.05 Hz high-pass
Max. sampling rate	50 kSamples/s
Input	single ended
AC coupling	0.05 Hz typ.
Resolution	16 bit
Measuring error	0,5%
Sampling type	simultaneous
Ground reference	diffrential
Supply voltage for electronic	via the E-bus
Current consumption via E-bus	typ. 220 mA
Configuration	via TwinCAT System Manager
Weight	approx. 65 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 [▸ 40]	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 [▸ 43]
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Protection class	IP20
Installation position	variable
Marking / Approval*)	CE, EAC, UKCA cULus [▸ 39], ATEX [▸ 37]

*) 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

2.3 Technology

2.3.1 Vibration analysis

Vibration analysis refers to deriving of information from existing (mechanical) vibrations e.g. in machines or certain production processes. It can take place during Condition Monitoring, e.g. for drives, punching and pressing tools, in production processes such as balancing of rotating parts, or during other movements such as torsion of towers/wind turbines.

The table below illustrates typical acceleration values for natural and technical processes.

Typical acceleration values

Machine or event	Typical g-value
Commercial aircraft (take-off)	≈ 0,5
Formula 1 car (start)	≈ 1 – 1,5
Commercial aircraft (turning, max.)	≈ 2,5
Pendulum at 90° amplitude	≈ 2
Space Shuttle during journey into orbit	max. 3 (exact)
Space Shuttle during re-entry into the Earth's atmosphere	max. 1.6
Typical roller coaster ride (max.)	4 (6)
Formula 1 car (cornering, max.)	≈ 4 – 5
Circular looping (base)	≥ 6
Combat aircraft/aerobatics (max.)	≈ 10 (13,8)
Ejector seat	15 – 20
Car back-rest breaks at	≈ 20
Head-on car collision	up to ≈ 50
Car passenger compartment during crash	max. 120
Survived by a human	≈ 180
Hard fist stroke	up to ≈ 100
Raindrop hitting the eye	up to ≈ 150
Ball pen hitting hard floor from 1 m height	order of magnitude 1000
Hard disk falling on hard floor from 1 m height (without deformation of the floor)	10,000 or more
Laboratory centrifuge	≈ 10.000
Ultra centrifuge	≈ 100.000
Rifle bullet during firing	≈ 100.000
Spike during ejection from a nettle cell	5.410.000
Nuclear bomb explosion (bomb case)	up to ≈ 10^{11}
Neutron star surface	≈ $2 \cdot 10^{11}$

2.3.2 Application of condition monitoring

Condition monitoring can be used to glean information on the state of rotating/moving parts through measurement of vibrations at machines/drives/gears and subsequent analysis with suitable mathematical tools (e.g. TwinCAT library, FFT, custom user programs).

The existing vibration is recorded continuously or at longer, regular time intervals and finally compared with a setpoint value/initial value (Fig. *Sample of ball bearing damage and subsequent analysis*).
In this way any damage can be detected at an early stage. Instead of changing components preventively on a regular basis or waiting for sudden damage and subsequent expensive downtime and possible consequential damage, repairs and downtimes become plannable. Needless failures, consequential damage or prematurely and costly replacement of intact parts can be avoided.

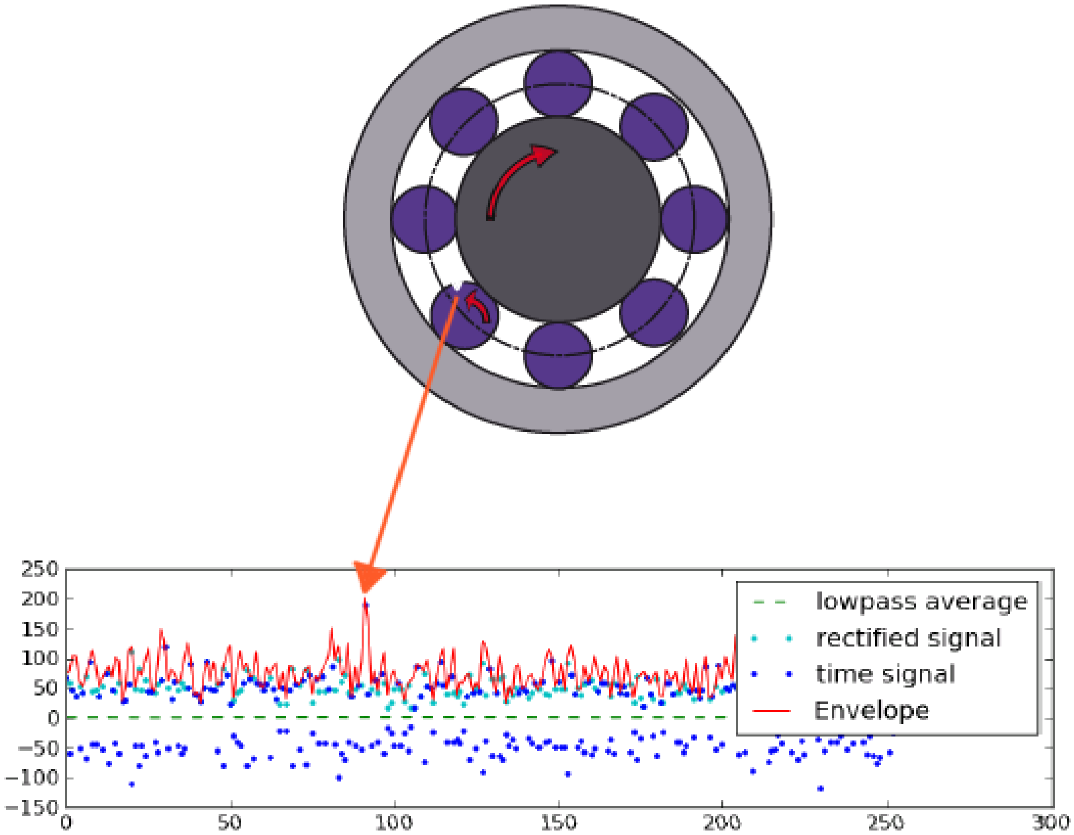


Fig. 5: Example of ball bearing damage and subsequent analysis

In the example above the so-called envelope enables analysis of shock pulses resulting from unevenness in the roller bearing. The defective element can be identified based on the periodicity (envelope spectrum).

2.3.3 Output signals of IEPE sensors

Vibrations can be recorded with IEPE (Integrated Electronics Piezo Electric) sensors, for example. The advantage of this technology is an integrated amplifier as impedance transformer, so that only a simple two-wire connection (coax) is required for the low-impedance output signal. IEPE sensors are typically supplied with 2…20 mA constant current. In inactive state they produce a constant DC bias voltage (zero voltage/ U_{bias}) typically 7…14 V. Depending on the acceleration of the sensor, an analog AC voltage generated proportionally to the movement is added to the sensor's U_{bias} ; e.g. a 50 Hz sinusoidal deflection with an amplitude of 1 g (= 9.81 m/s^2) produces a 50 Hz sinusoidal output voltage of AC +/-50 mV + U_{bias} in the case of a sensor with a sensitivity of 50 mV/g (Fig. *Output signal of an IEPE sensor (sample)*). The maximum output signal of a sensor is usually AC +/-5 V (+ U_{bias}).

Increasing cable length results in increasing cable capacitance (typically 100 pF/m), so that the control capability of the integrated amplifier drops with increasing signal frequency. This can be partly compensated by increasing the supply current (Fig. *Control capability of the IEPE impedance transformer depending on cable capacitance and supply current*).

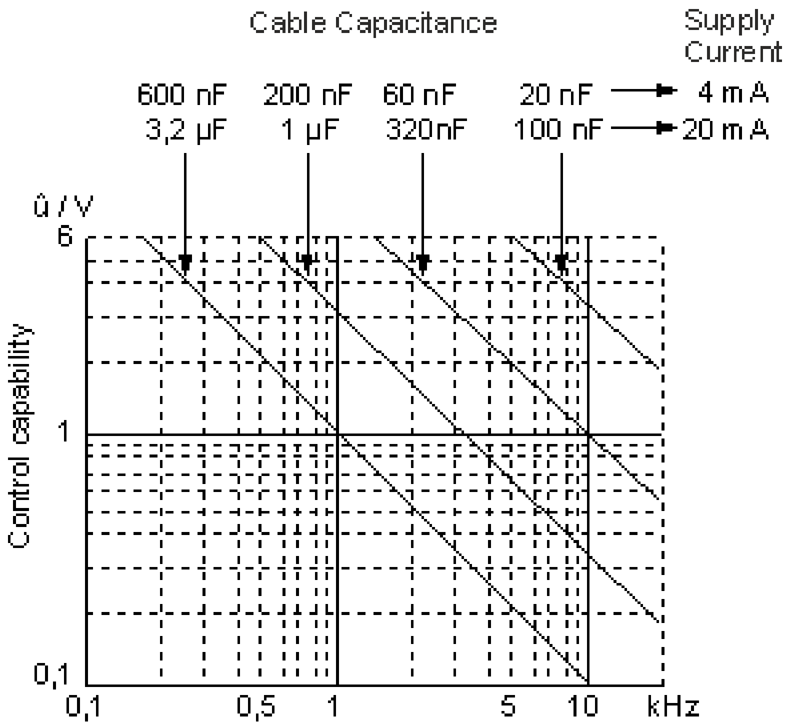


Fig. 6: Control capability of the IEPE impedance transformer depending on cable capacitance and supply current.

The basic properties of IEPE acceleration sensors are described by various parameters such as sensitivity (e.g. 50 mV/g), measuring range (e.g. +/-100 g), +/-3 dB frequency range (under 1 Hz to several kHz), current consumption (2…20 mA), bias voltage etc. The figure *Frequency response of an acceleration sensor* shows an example of a frequency response (amplitude of the output signal in relation to the frequency).

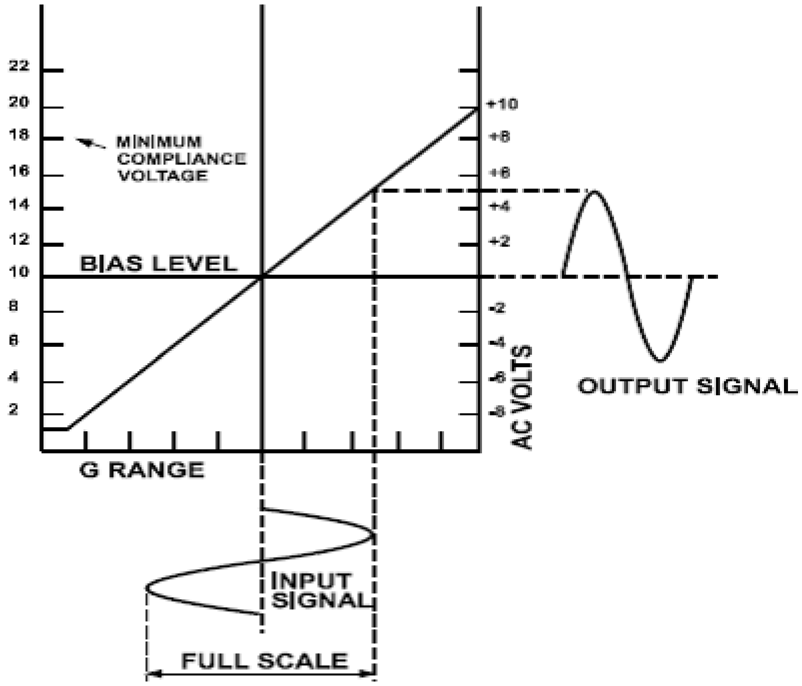


Fig. 7: Output signal of an IEPE sensor (sample)

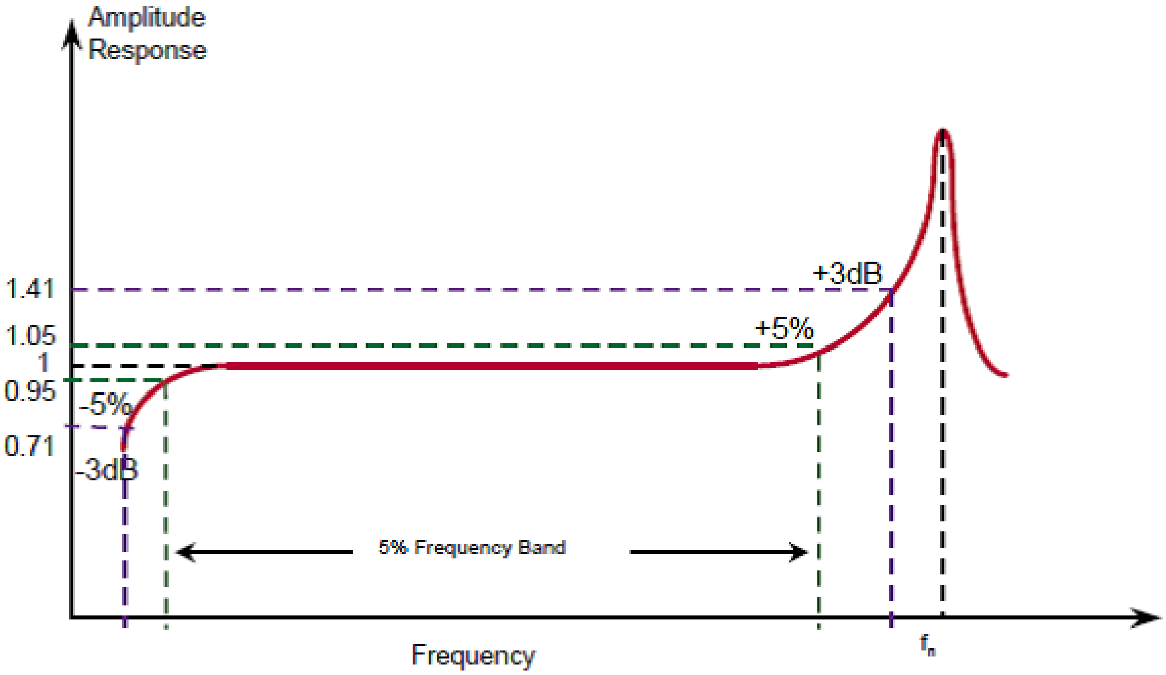


Fig. 8: Frequency response of an acceleration sensor

Other proprietary names for this electrical technique include DeltaTron®, Isotron™, ICP™, Piezotron™ or CCLD

Note: At their core IEPE sensors usually have quartz crystals, which experience small charge changes due to the motion. Measuring and transporting these over several meters requires complex cable installations and measuring instruments for charge amplification. For cost-effective and robust application in industrial environments the solution is to integrate a simple charge amplifier as impedance transformer in the sensor. This enables it to generate the above-mentioned voltage signal and transport the vibration information over larger distance (several tens of meters).

2.3.4 Basic principles of IEPE technology

IEPE ("Integrated Electronics Piezo-Electric") is the standardized name for an analog electrical interface between piezoelectric sensors and electronic analysis equipment. Different manufacturers have developed their own brand names, such as ICP®, CCLD®, Isotron®, DeltaTron®, Piezotron®.

Application

Piezoelectric sensors are usually based on a quartz, in which an electrical charge is shifted under mechanical load. The charge can be detected as a voltage if the measurement is made with sufficiently high impedance. The measurement is a preferably static process, which must take place within a much smaller time interval than 10 seconds, since otherwise the charge difference is dissipated through external or internal derivatives. Such a sensor is therefore less suitable for static long-term loads, such as weighing a silo. Such sensors tend to be used for all kinds of predominantly higher frequency vibration measurements (unbalance detection, sound signals via microphones up to ultrasound, mechanical vibrations, foundation monitoring, etc.).

Over the decades, two electrical forms of interfaces to the evaluation unit have developed:

- Direct charge output
- IEPE output

Charge output

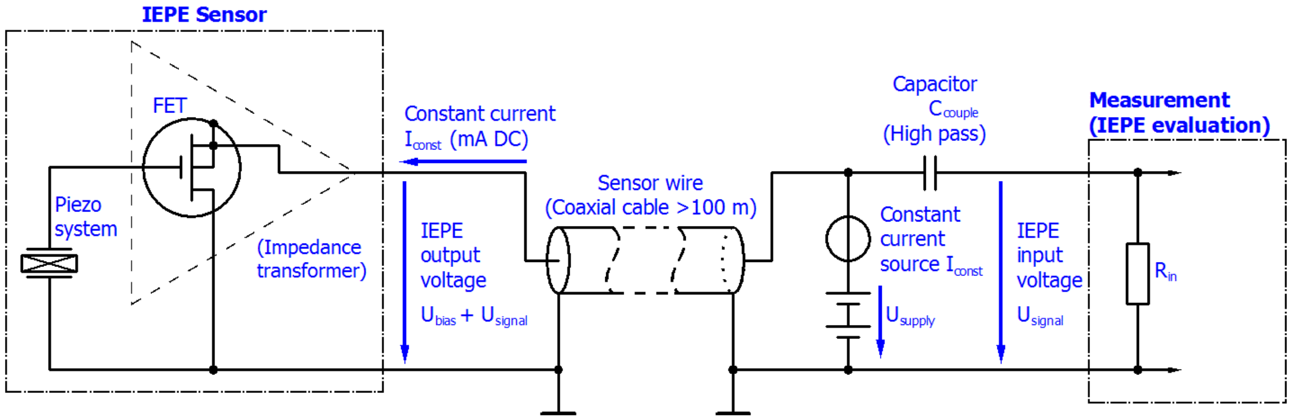
The sensor's output signal is provided by a very low change of charge (usually in the range of a little femto to pico coloumb) and tapped by a (possibly short) 2-wire cable. Thus a so called charge amplifier is an essential component of the measurement electronics.

Advantages: the sensor can be exposed to high temperatures over 150 °C; no power supply is required.

Disadvantages: very sensitive to external influences on the cable (line length, possible movement of the cable, type of cable and shielding, electromagnetic fields, etc.); elaborate receiving electronics and cable due to high source impedance.

IEPE output

Since the charging output interface is not generally accepted for industrial applications, a more robust transfer method was sought at an early stage. Thus, for IEPE integrates a field effect transistor (FET) directly at the output of the sensor.


IEPE Sensor
FET
Piezo system
(Impedance transformer)
Constant current I_{const} (mA DC)
IEPE output voltage
$U_{bias} + U_{signal}$
Sensor wire (Coaxial cable >100 m)
Capacitor C_{couple} (High pass)
Constant current source I_{const}
U_{supply}
IEPE input voltage
U_{signal}
Measurement (IEPE evaluation)
R_{in}

If this sensor is supplied with constant current between 2 and 8 mA via the two-pole cable, the resulting bias voltage is usually approx. 8 to 15 V. If the piezo system is now directly or indirectly (e.g. by a diaphragm) loaded by a measure quantity e.g. force in form of pressure or acceleration, the FET changes its channel resistance caused by change of charge amount on its gate; according to this, a change of the gate source voltage. Because of the supply of I_{const} by a constant current source, the bias voltage changes correspondingly by the mechanical load of about few volts. Although the evaluation unit now usually has to provide the constant current supply but is able to deduce the measurement quantity by the back-measured voltage.

Advantages: robust system, which is suitable for operation in industrial conditions.

Disadvantages: upper temperature limit for the sensor 150 to 200 °C, smaller dynamic range.

Notes on constant current

- The higher the feed current, the more the vibration sensor heats up. This may cause a disadvantageous effect on e.g. the basic accuracy of the sensor. For this, note the information provided by the sensor manufacturer.
- The higher the feed current, the higher the maximum transferable signal frequency, since charge inflows and outflows can be balanced more quickly on the cable.
- The higher the supply current, the higher the resulting bias voltage. As a result, the transfer may become more robust against electromagnetic influence, but on the other hand that for large amplitudes the measurement quantity may enter the upper saturation faster.

Notes on the IEPE measuring device

- In some IEPE measuring devices the supply current can be switched off (0 mA), so that they can be used for voltage measurements, cf. for example Beckhoff ELM3604
- Since normally only AC signals are of interest in the vibration range, IEPE evaluations have an electrical high-pass with a cutoff frequency of approximately 10 Hz on the input side. Depending on the application e.g. the capturing of slow tower vibrations, the limit of this high-pass may be relevant, cf. for example the configurable high-pass of the ELM3604, which can be switched off.
- The bias voltage is suitable for detection of wire breakage or short circuit cases respectively, e.g. see also ELM3604 diagnostics options.

Notes on the IEPE sensor

- If IEPE sensors are mounted on high-voltage or frequency converter-controlled motors, an electrically insulated installation or an insulated sensor may be recommendable. In other cases, interference effects on the IEPE measurement have already been observed. This purely functional consideration must be weighed up by the plant installer against normative and electrical safety requirements.

2.3.5 Beckhoff EL3632

A suitable measuring transducer is the EL3632 with two independent channels, each with an integrated and separately configurable current source, integrated high-pass filter, separately parameterisable analog filter for signal processing and parameterisable sampling rate/oversampling factor. The EL3632 does not pre-analyses the vibration data but transports the continuous amplitude information to the EtherCAT master. The master analyses the measured data on the existing higher-level PC via the TwinCAT library or custom software.

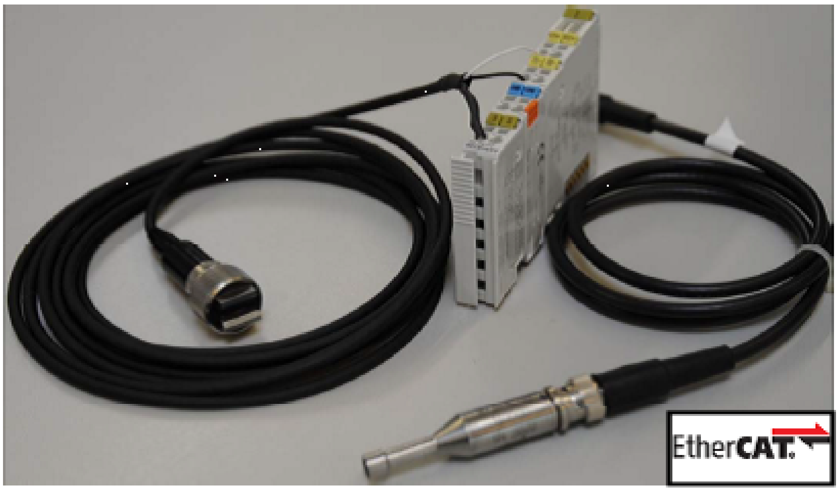


Fig. 9: EL3632 with periphery

2.4 Start

For commissioning:

- mount the EL3632 as described in the chapter Mounting and wiring [▶ 36]
- configure the EL3632 in TwinCAT as described in the chapter Commissioning [▶ 52].

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

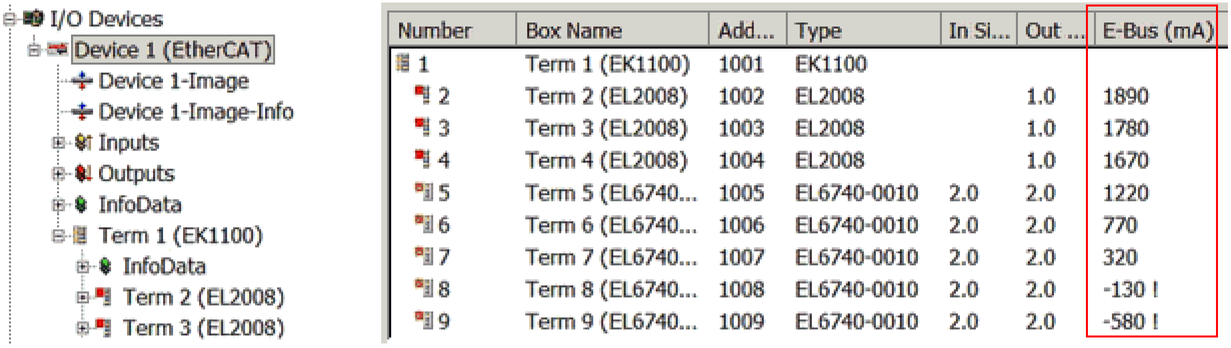
It is recommended to use the appropriate Beckhoff components e.g.
- cable sets ZK1090-9191-xxxx respectively
- RJ45 connector, field assembly ZS1090-0005
- EtherCAT cable, field assembly ZB9010, ZB9020

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.



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. 10: System manager current calculation

NOTE

Malfunction possible!

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

3.3 General notes for setting the watchdog

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

The EtherCAT slave controller (ESC) features two watchdogs:

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

SM watchdog (SyncManager Watchdog)

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

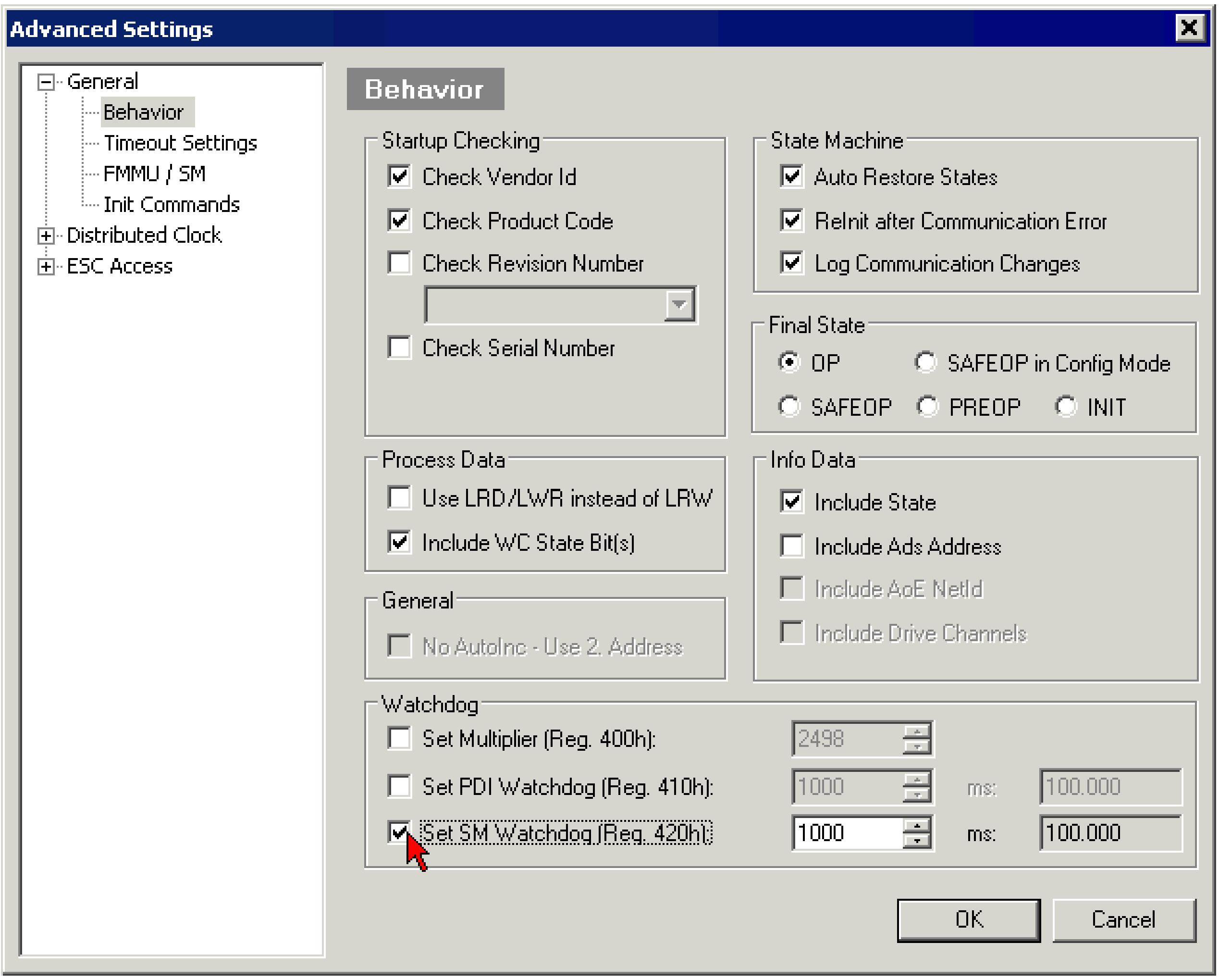
The SyncManager watchdog monitors correct and timely process data communication with the ESC from the EtherCAT side.

PDI watchdog (Process Data Watchdog)

If no PDI communication with the EtherCAT slave controller (ESC) takes place for longer than the set and activated PDI watchdog time, this watchdog is triggered.
PDI (Process Data Interface) is the internal interface between the ESC and local processors in the EtherCAT slave, for example. The PDI watchdog can be used to monitor this communication for failure.

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

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



Advanced Settings

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

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. 11: EtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

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

Multiplier

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

1/25 MHz * (watchdog multiplier + 2) = 100 µs (for default setting of 2498 for the multiplier)

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

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

Example "Set SM watchdog"

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

Calculation

Multiplier = 2498 → watchdog base time = 1 / 25 MHz * (2498 + 2) = 0.0001 seconds = 100 µs
SM watchdog = 10000 → 10000 * 100 µs = 1 second watchdog monitoring time

⚠ CAUTION

Undefined state possible!

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

⚠ CAUTION

Damage of devices and undefined state possible!

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

3.4 EtherCAT State Machine

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

A distinction is made between the following states:

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

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

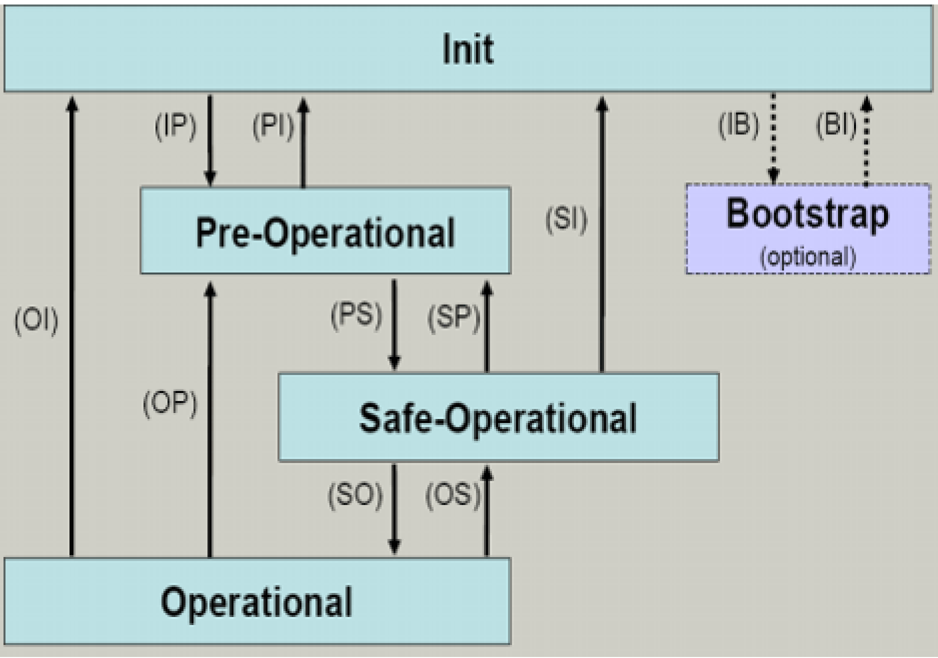


Fig. 12: States of the EtherCAT State Machine

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 [▶ 27] 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$_{dec}$)
- SubIndex: 0x00…0xFF (0...255$_{dec}$)

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.

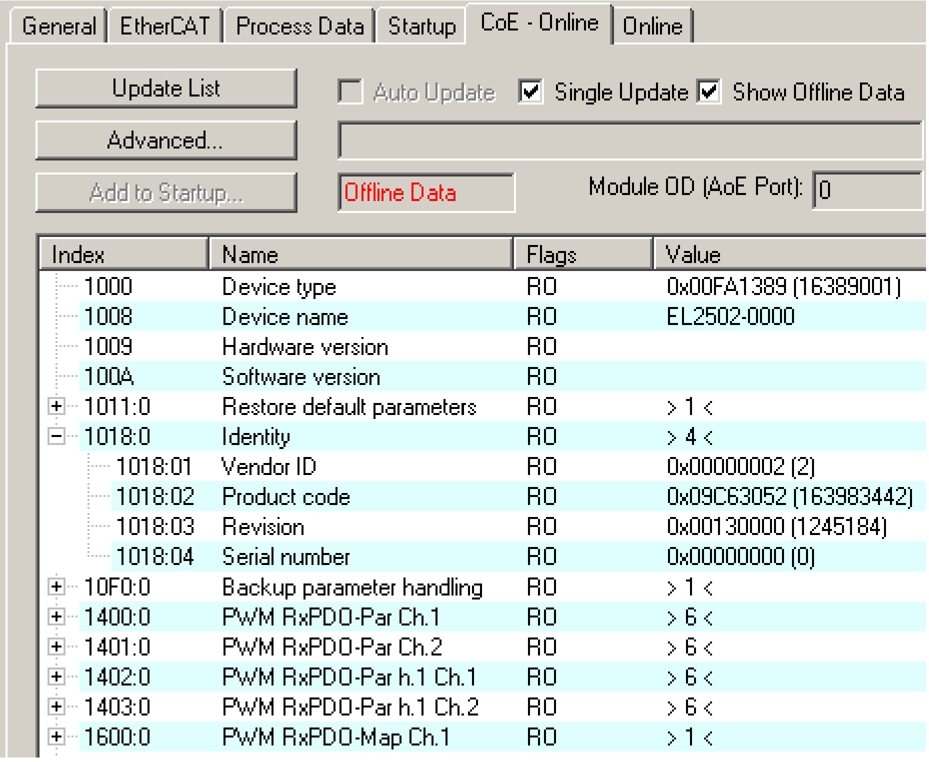
Other important ranges are:

- 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. 13: “CoE Online” tab

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

Data management and function “NoCoeStorage”

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

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

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.

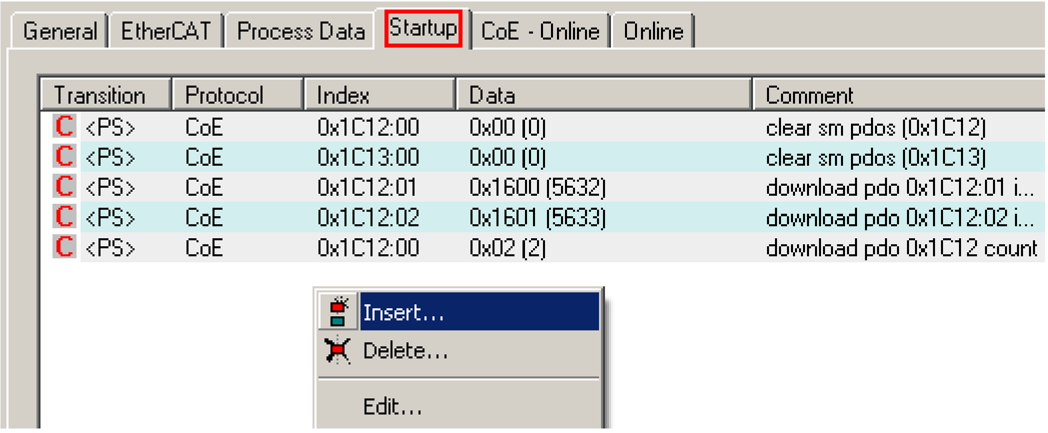
Startup list

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. 14: Startup list in the TwinCAT System Manager

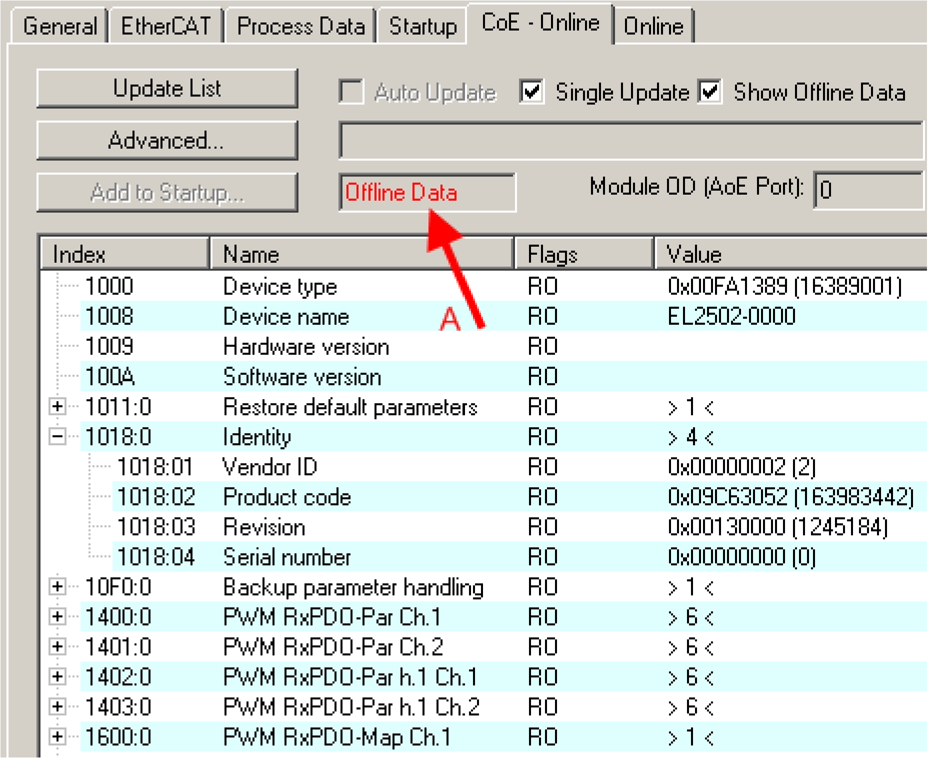
The Startup list may already contain values that were configured by the System Manager based on the ESI specifications. Additional application-specific entries can be created.

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

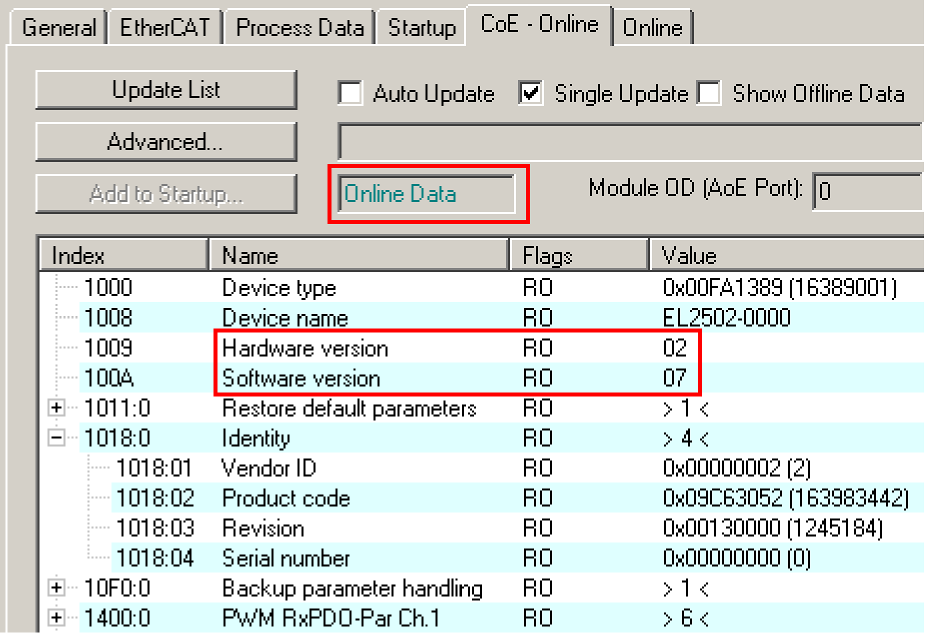
Offline Data

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. 15: 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. 16: 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 Mounting and wiring

4.1 Instructions for ESD protection

NOTE

Destruction of the devices by electrostatic discharge possible!

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

- Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly.
- Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.).
- Surroundings (working place, packaging and personnel) should 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.

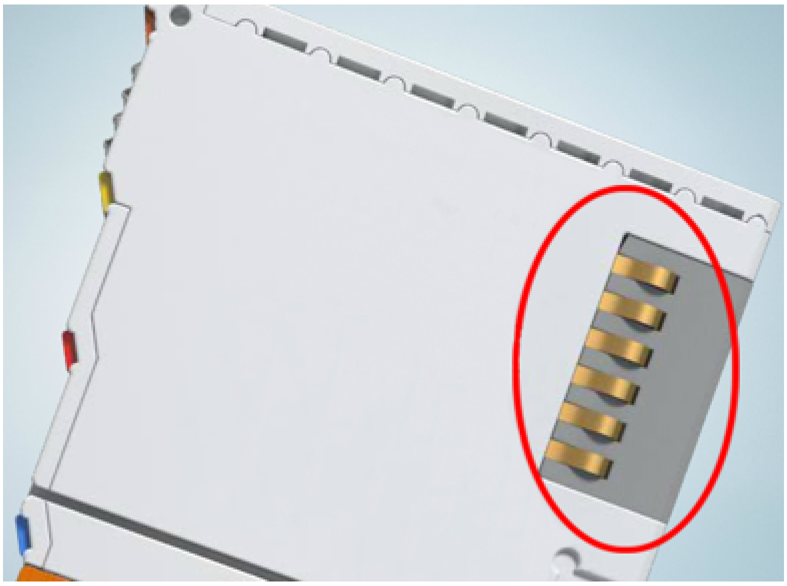


Fig. 17: Spring contacts of the Beckhoff I/O components

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

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 on the Beckhoff homepage www.beckhoff.com!

4.3 UL notice

c UL us	**Application** Beckhoff EtherCAT modules are intended for use with Beckhoff's UL Listed EtherCAT System only.
c UL us	**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).
c UL us	**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!

Assembly

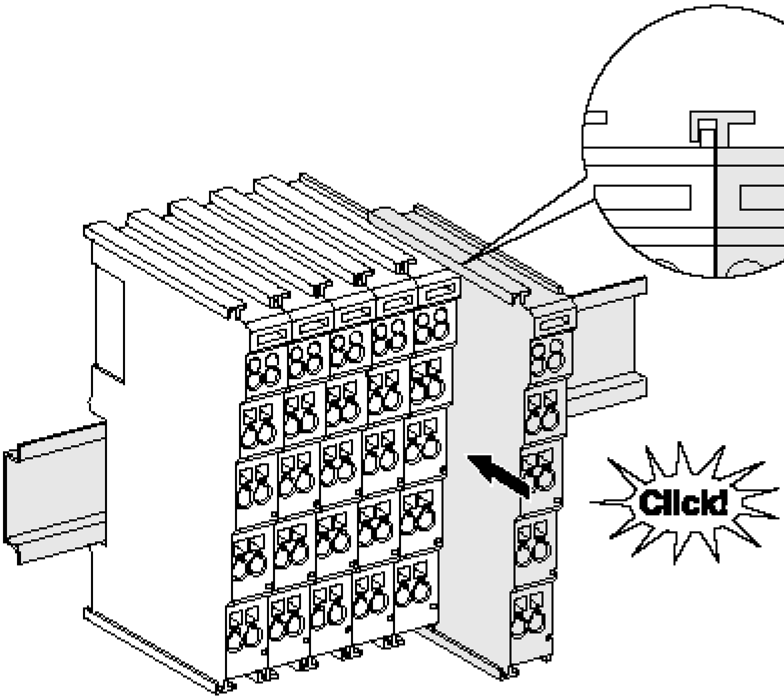


Fig. 18: Attaching on mounting rail

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

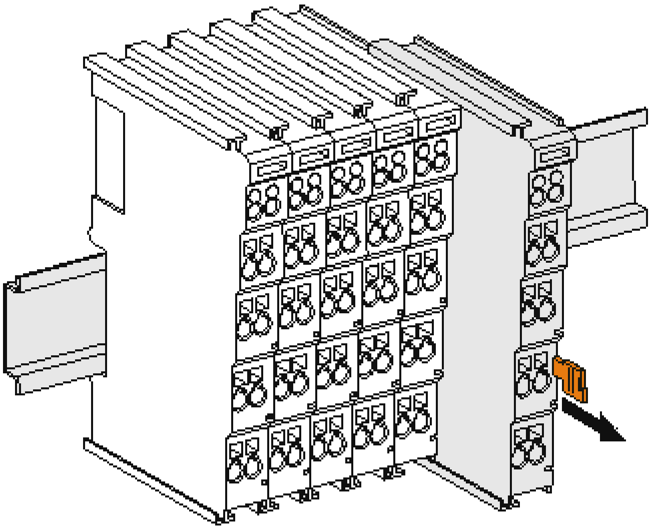


Fig. 19: Disassembling of terminal

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.

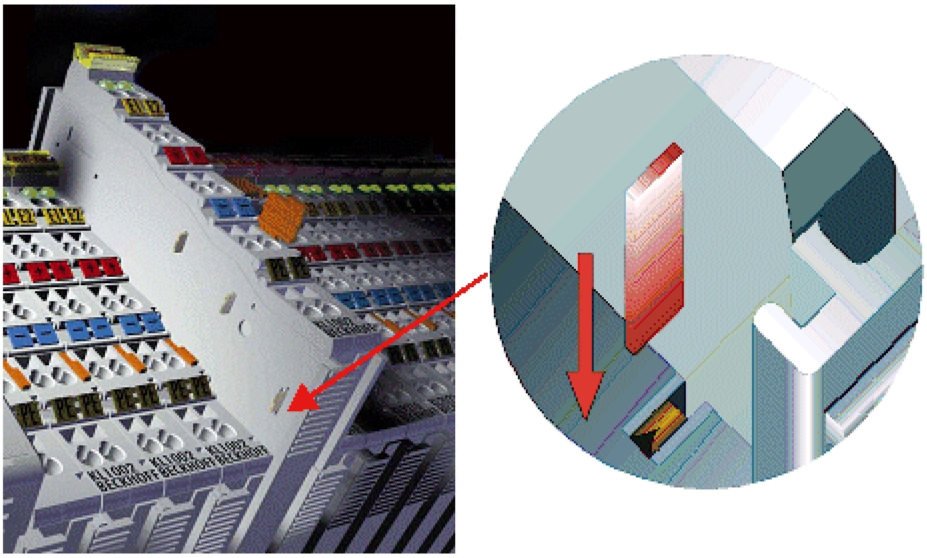


Fig. 20: Power contact on left side

NOTE

Possible damage of the device

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

⚠ WARNING

Risk of electric shock!

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

4.5 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 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)

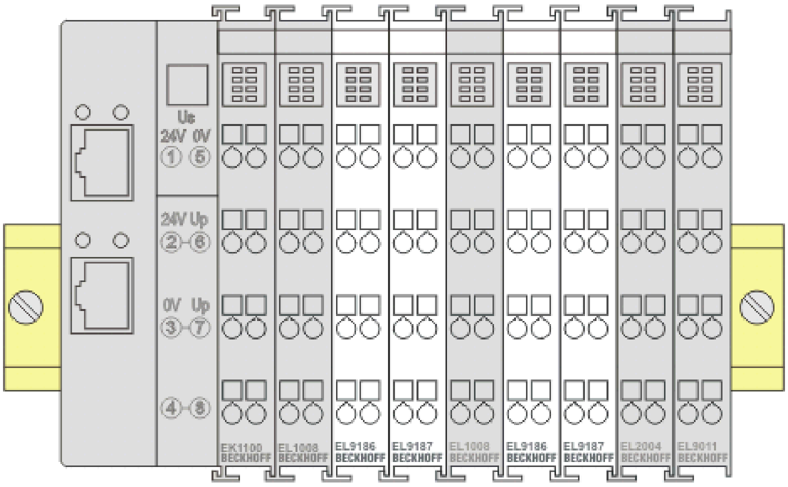


Fig. 21: Correct positioning

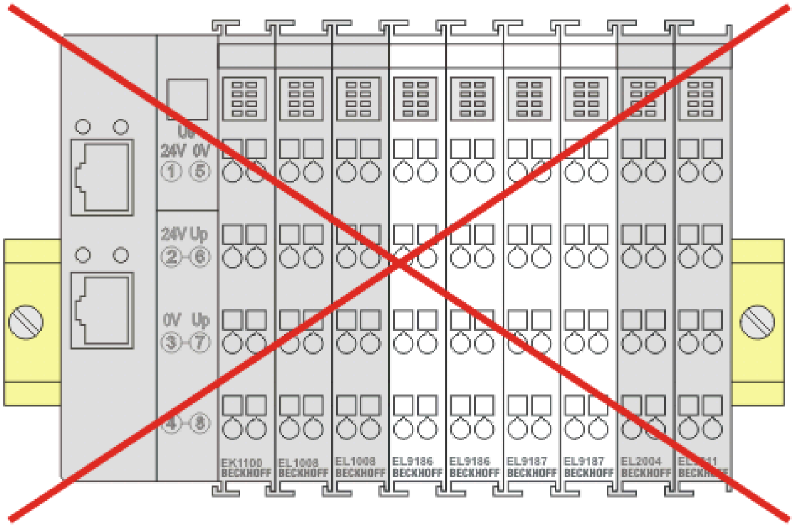


Fig. 22: Incorrect positioning

4.7 Connection

4.7.1 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)

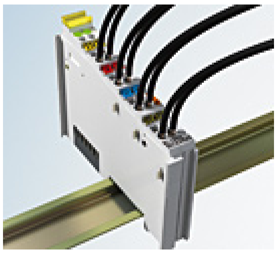


Fig. 23: Standard wiring

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)

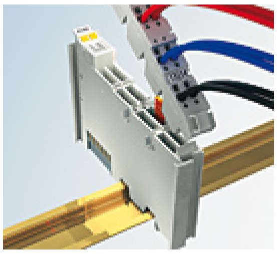


Fig. 24: Pluggable wiring

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)

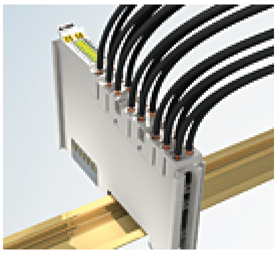


Fig. 25: High Density Terminals

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 [▸ 47]!

4.7.2 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

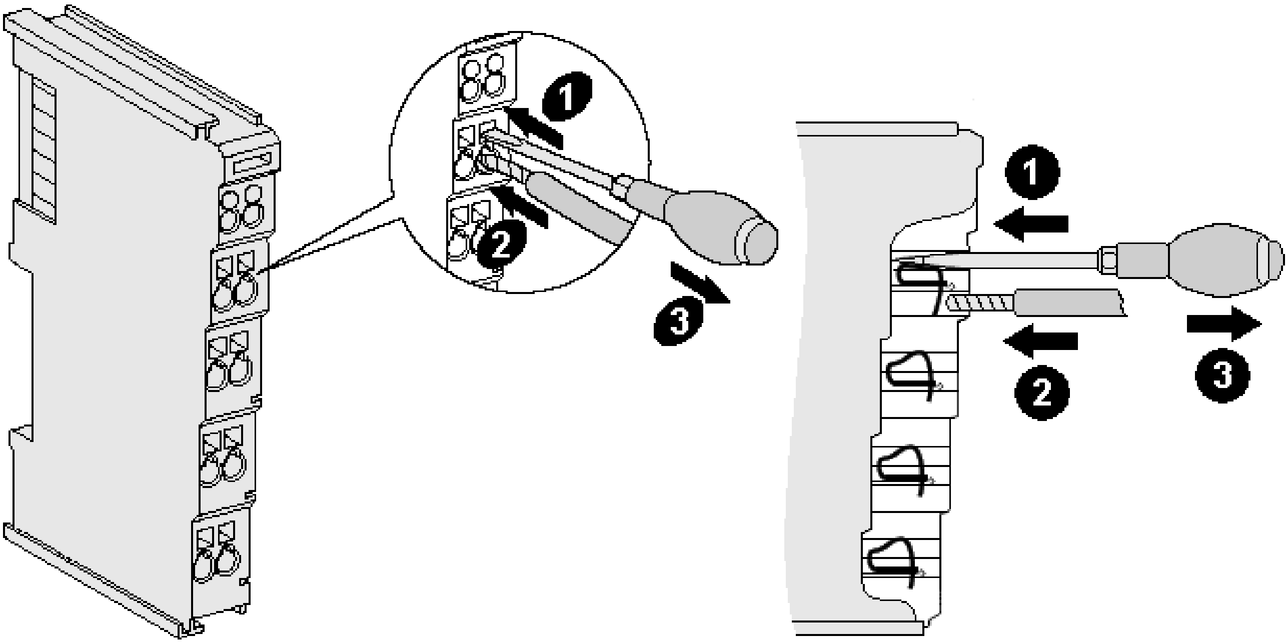


Fig. 26: Connecting a cable on a terminal point

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 [▶ 46]) 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 [▸ 46])
Wire stripping length	8 ... 9 mm

4.7.3 Shielding

Shielding

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

4.8 LEDs and connection

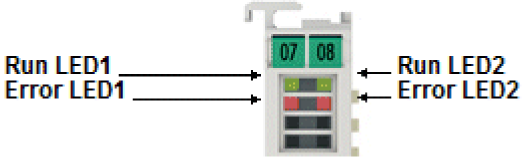


Fig. 27: LEDs

LEDs

LED	Color	Meaning	
RUN	green	This LED indicates the terminal's operating state:	
		off	State of the EtherCAT State Machine [▸ 29]: **INIT** = initialization of the terminal
		flashing	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		single flash	State of the EtherCAT State Machine: **SAFEOP** = verification of the Sync Manager [▸ 102] 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
Error 1/2	red	Open circuit error	

Connection EL3632

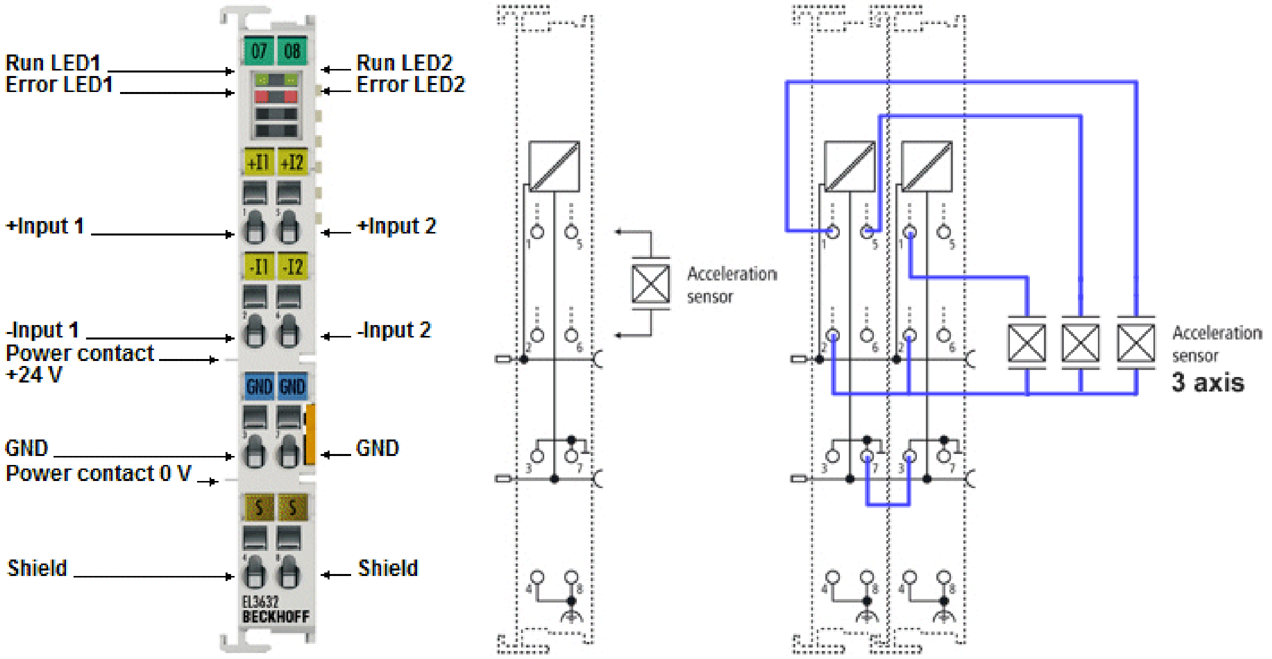


Fig. 28: Connection

Note: Terminal point 2 and 6 (- Input of the channels) are internally linked. Thus, a multichannel IEPE sensor can be connected, that supports one GND returning wire only. If a three channel sensor (3 axis, triaxis) will be connected via two terminals, each GND and respectively each -Input have to bridged.

Terminal point	No.	Comment
+ Input 1	1	+ Input 1
- Input 1	2	- Input 1
GND	3	Ground (internally connected with terminal point 7)
Shield	4	Shield (internally connected to terminal point 8)
+ Input 2	5	+ Input 2
- Input 2	6	- Input 2
GND	7	Ground (internally connected with terminal point 3)
Shield	8	Shield (internally connected to terminal point 4)

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

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

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

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

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

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

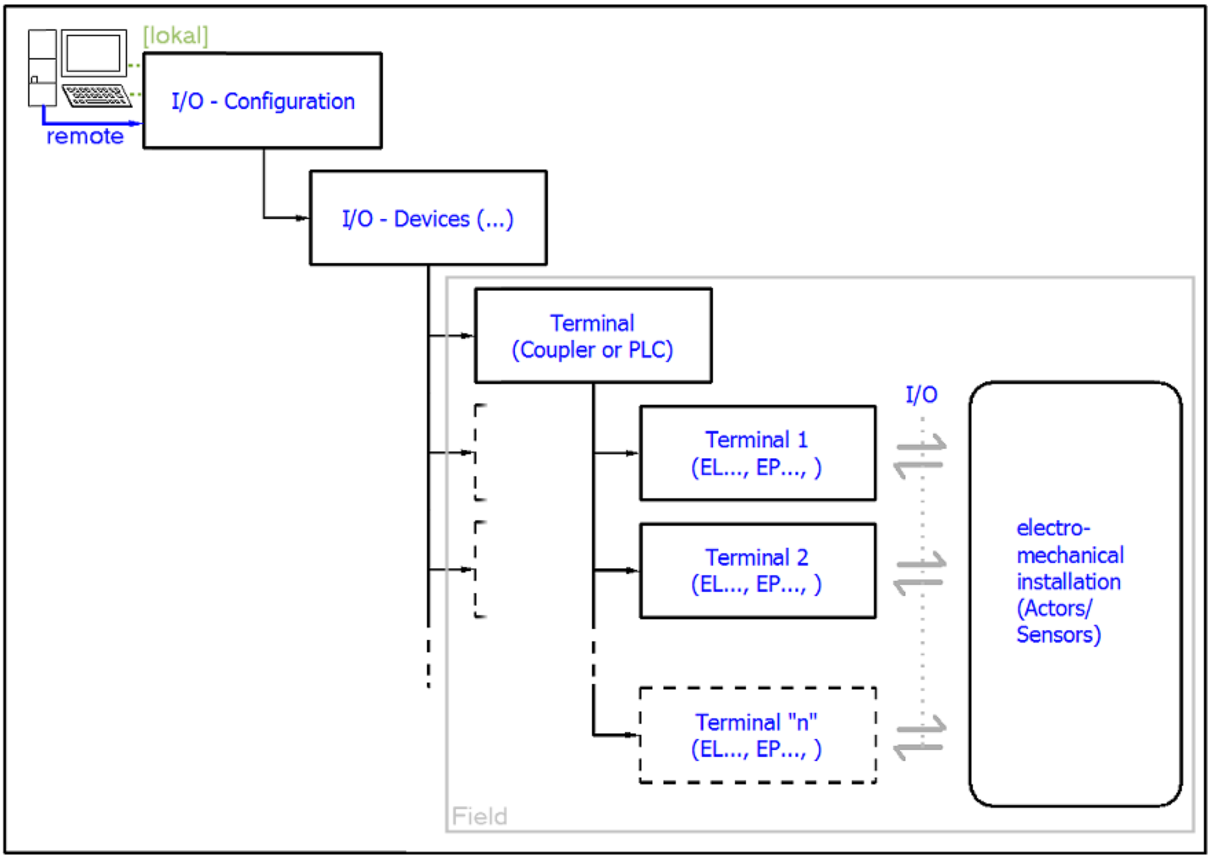


Fig. 29: Relationship between user side (commissioning) and installation

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

Sample configuration (actual configuration)

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

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

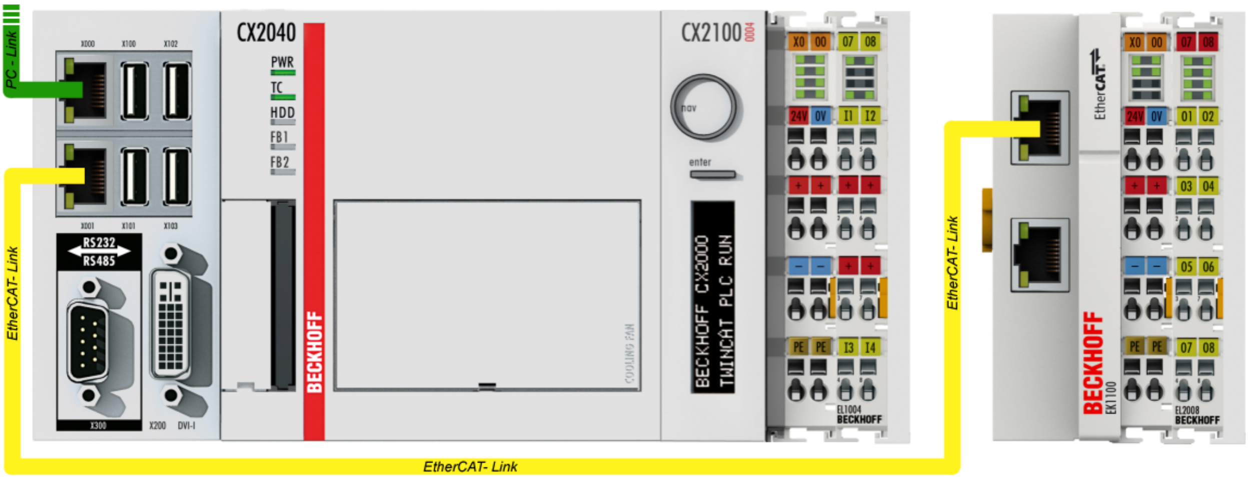


Fig. 30: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

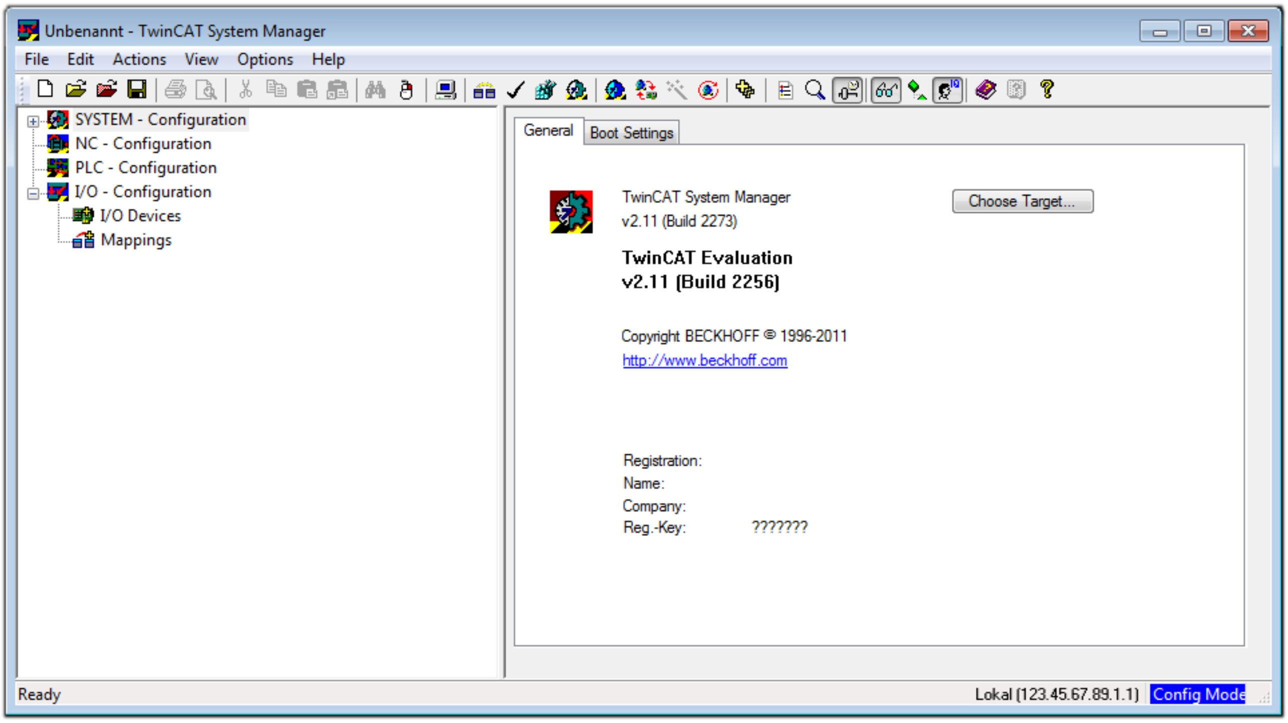
Note that all combinations of a configuration are possible; for example, the EL1004 terminal could also be connected after the coupler, or the EL2008 terminal could additionally be connected to the CX2040 on the right, in which case the EK1100 coupler wouldn’t be necessary.

5.1.1 TwinCAT 2

Startup

TwinCAT 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

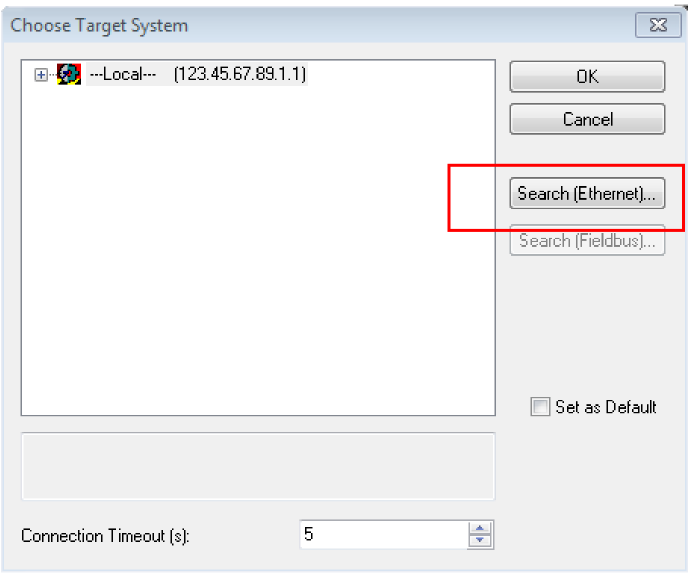
Registration:
Name:
Company:
Reg.-Key: ???????

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 thereby the next step is "Insert Device [▶ 57]".

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



---Local--- (123.45.67.89.1.1)

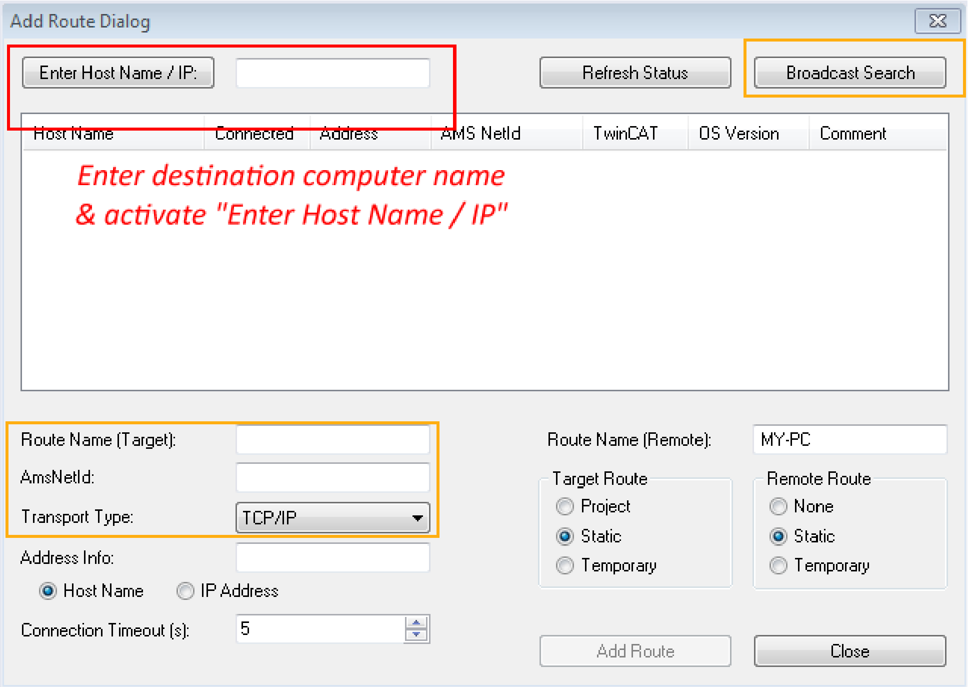
☐ Set as Default

Connection Timeout (s): 5

Fig. 32: Selection of the target system

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

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



Host Name	Connected	Address	AMS NetId	TwinCAT	OS Version	Comment

Route Name (Target):

AmsNetId:

Transport Type: TCP/IP

Address Info:

● Host Name ○ IP Address

Connection Timeout (s): 5

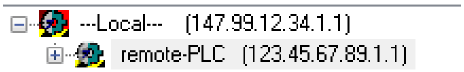
Route Name (Remote): MY-PC

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 password may have to be entered):



---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 menu “Actions” → “Set/Reset TwinCAT to Config Mode…” (Shift + F4).

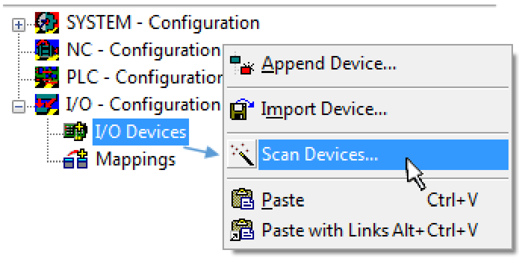
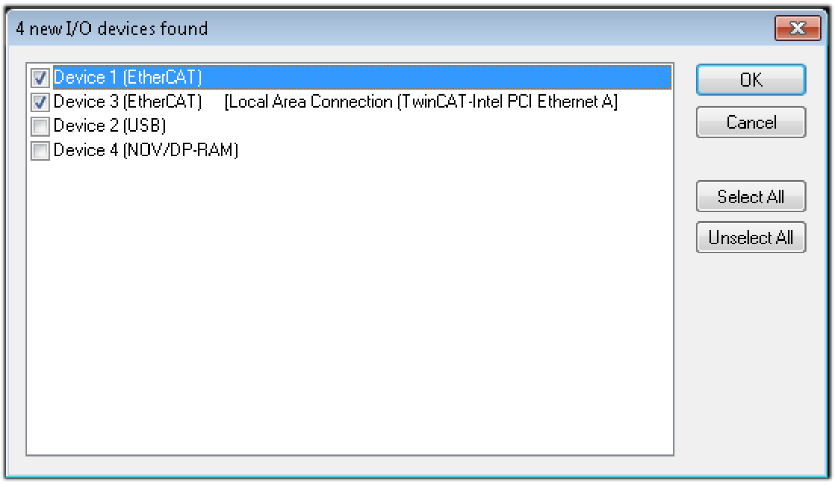


Fig. 34: Select “Scan Devices...”

Confirm the warning message, which follows, and select “EtherCAT” 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 the devices to be integrated

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

Based on the sample configuration [▶ 53] described at the beginning of this section, the result is as follows:

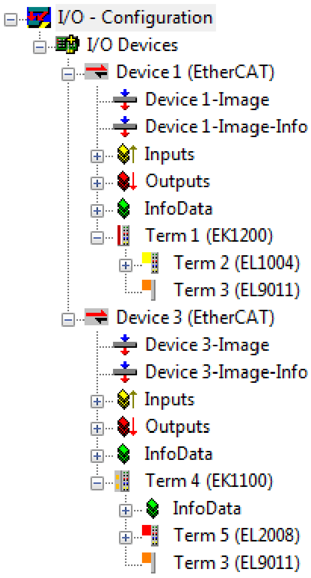


Fig. 36: Mapping of the configuration in the TwinCAT 2 System Manager

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

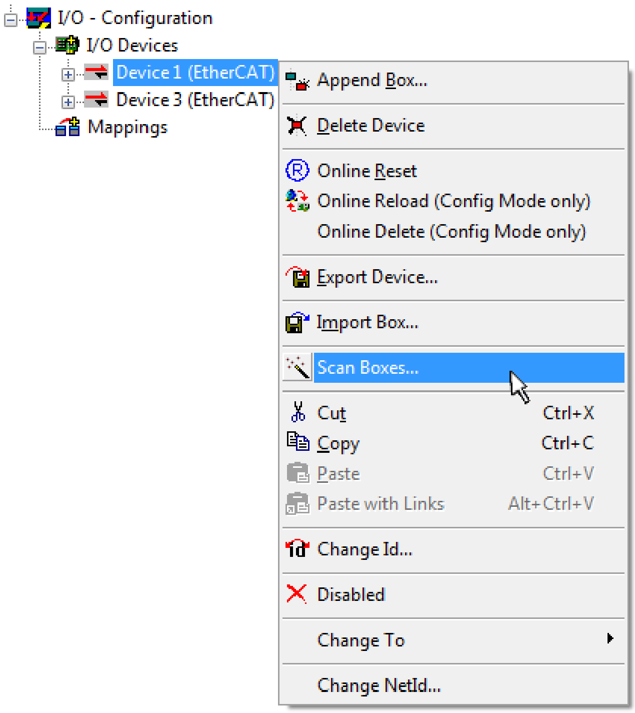


Fig. 37: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

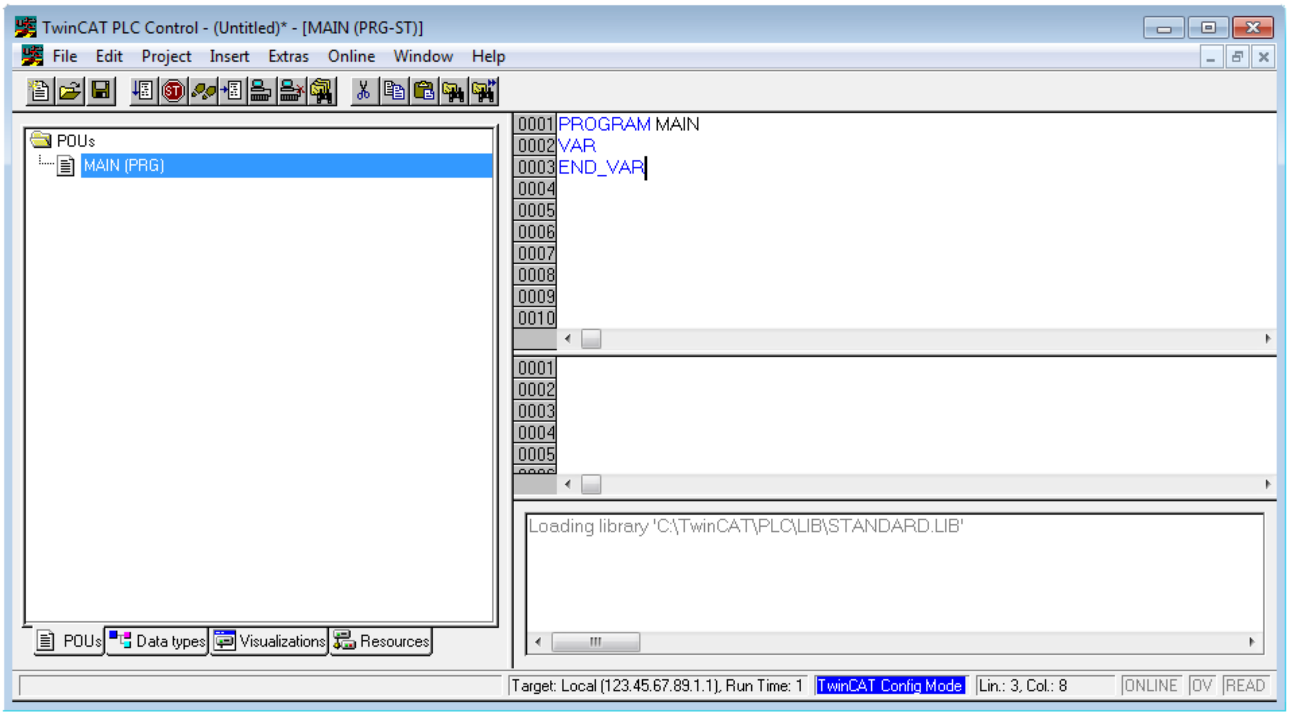
TwinCAT PLC Control is the development environment for the creation of the controller in different program environments: TwinCAT PLC Control supports all languages described in IEC 61131-3. There are two text-based languages and three graphical languages.

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

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

The following section refers to Structured Text (ST).

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



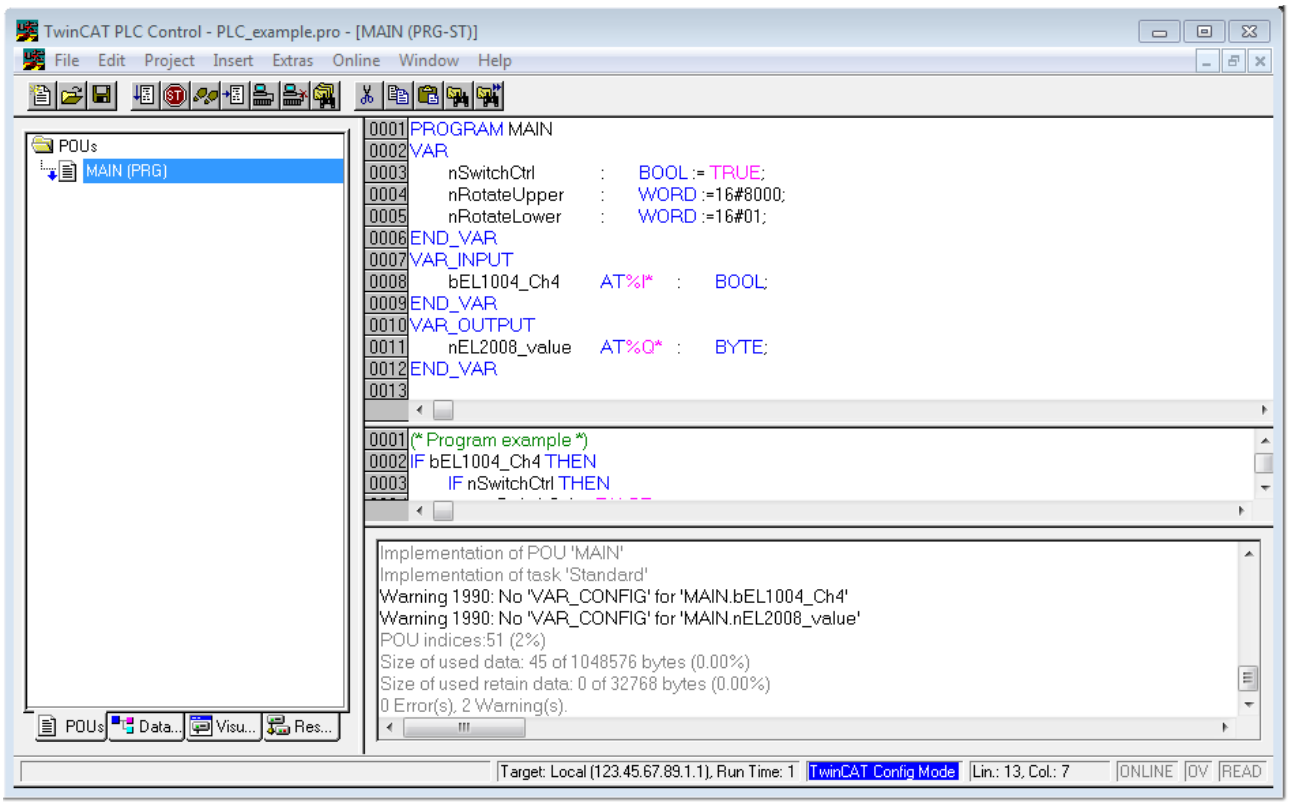
```
PROGRAM MAIN
VAR
END_VAR
```

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

Sample variables and a sample 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
```

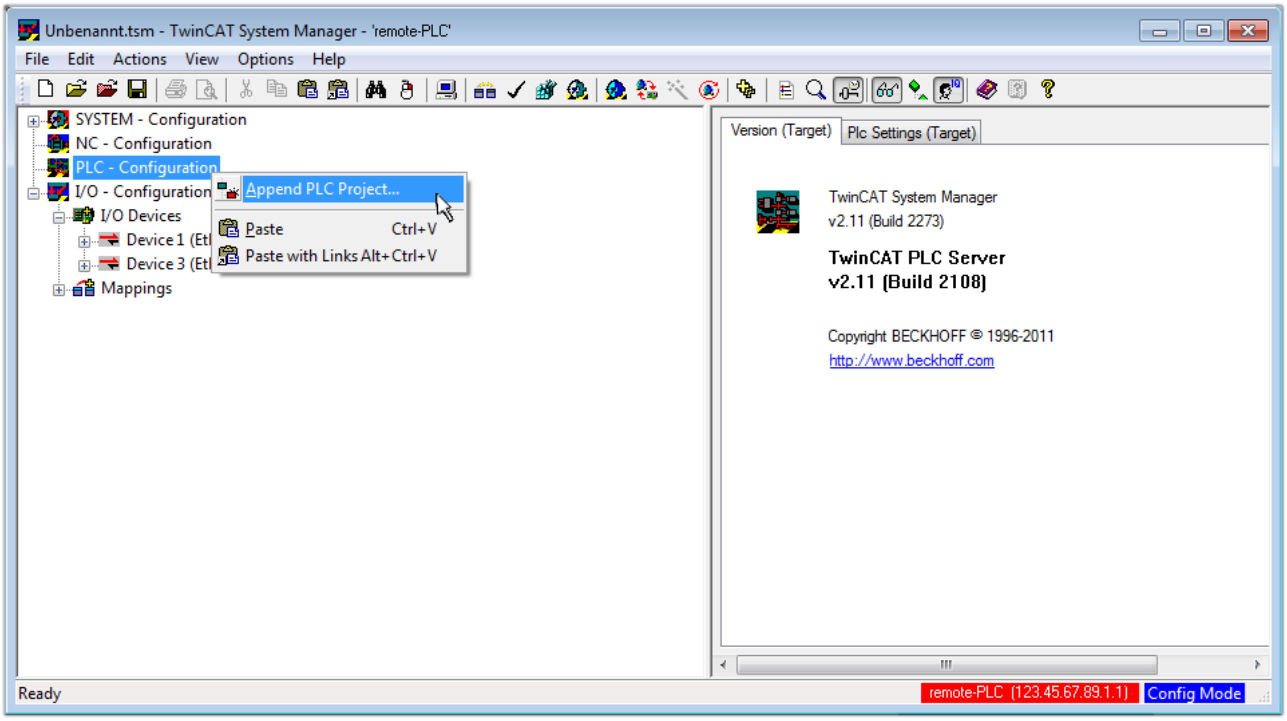
```
Implementation of POU 'MAIN'
Implementation of task 'Standard'
Warning 1990: No 'VAR_CONFIG' for 'MAIN.bEL1004_Ch4'
Warning 1990: No 'VAR_CONFIG' for 'MAIN.nEL2008_value'
POU indices:51 (2%)
Size of used data: 45 of 1048576 bytes (0.00%)
Size of used retain data: 0 of 32768 bytes (0.00%)
0 Error(s), 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: Sample program with variables after a compile process (without variable integration)

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

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



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:

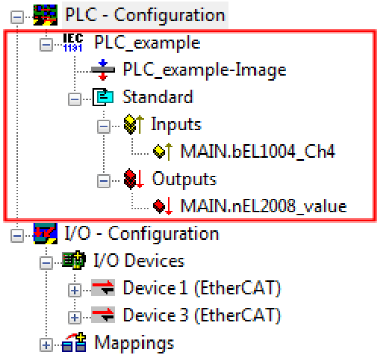
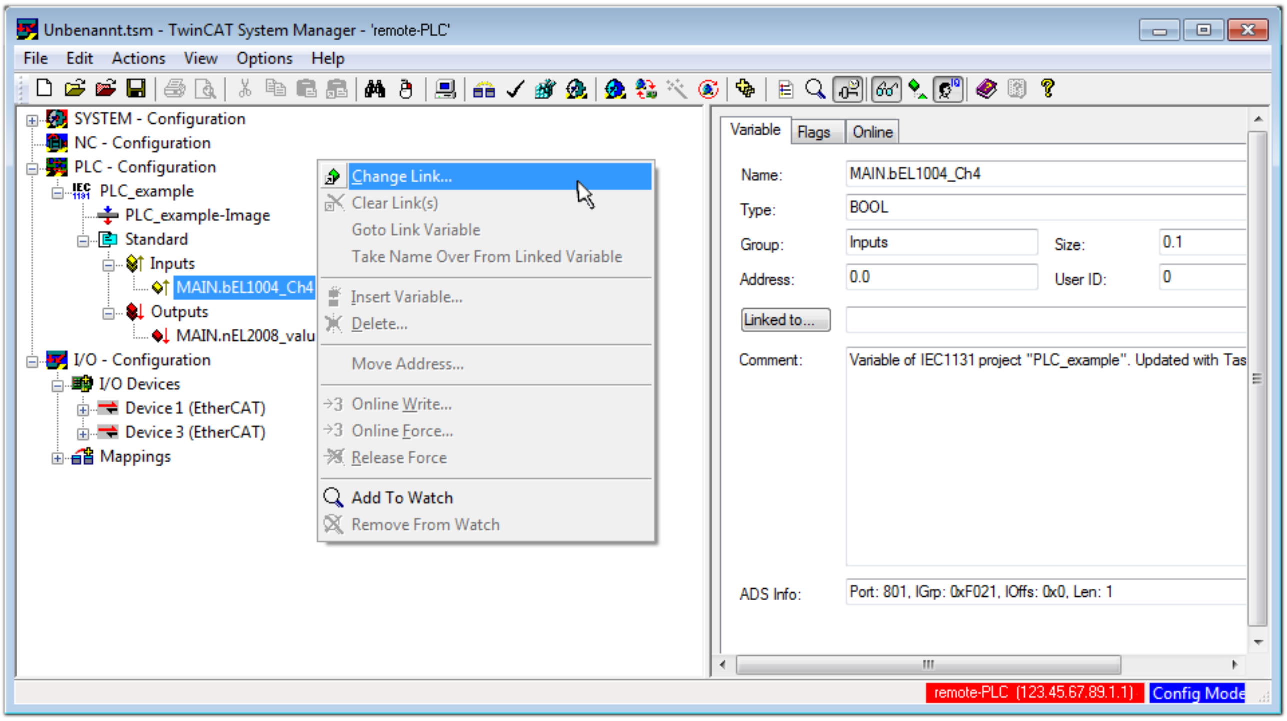


Fig. 41: PLC project integrated in the PLC configuration of the System Manager

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

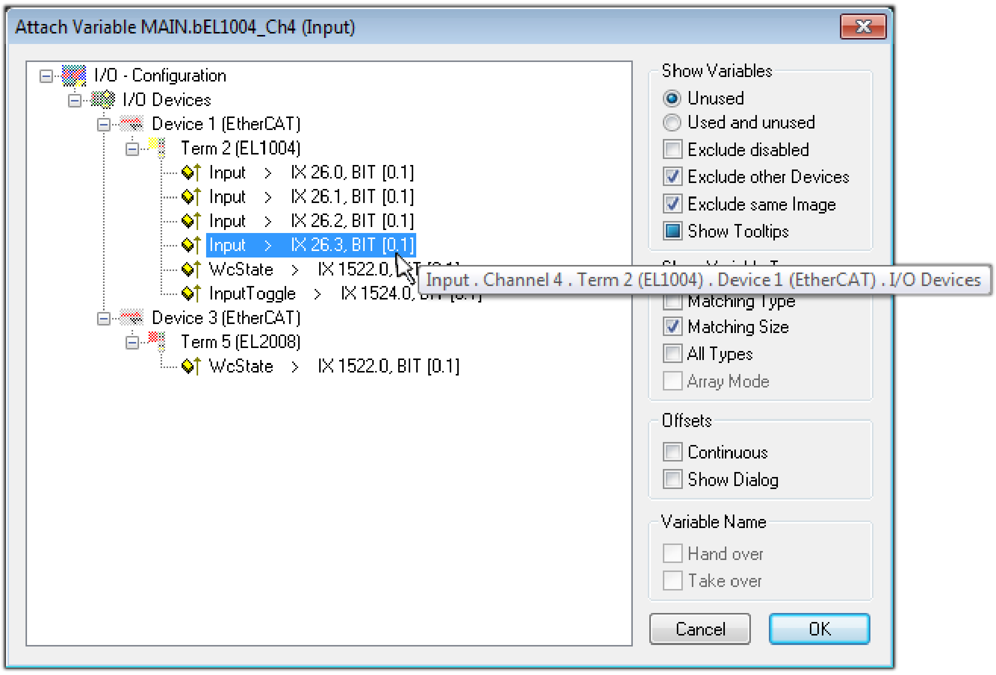
Address: 0.0 User ID: 0

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 variable “bEL1004_Ch4” of type BOOL 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

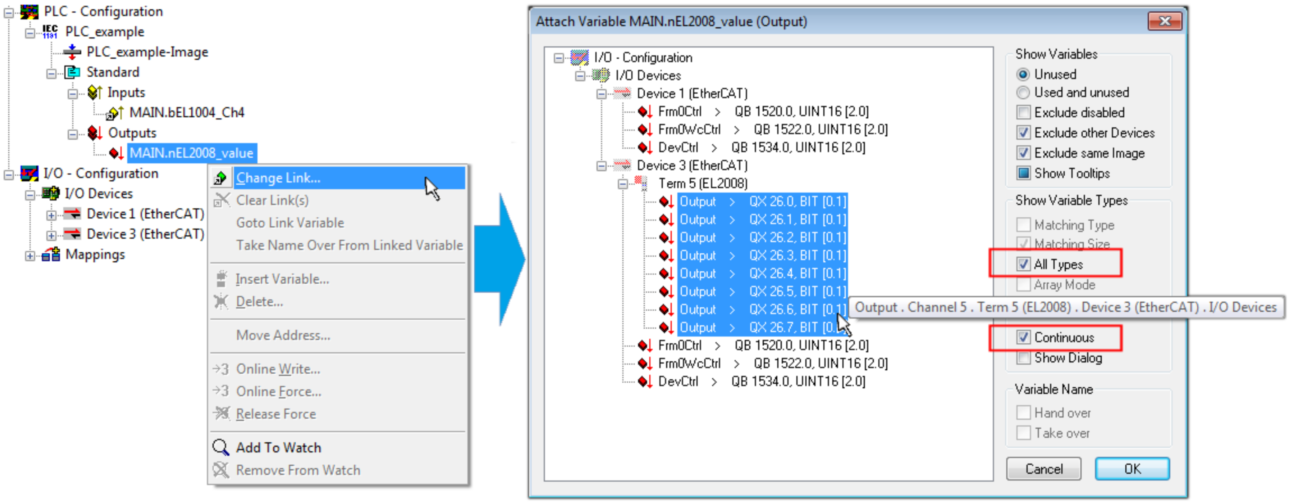
Show Variable Types
□ Matching Type
☑ Matching Size
□ All Types
□ Array Mode

Offsets
□ Continuous
□ Show Dialog

Variable Name
□ Hand over
□ Take over

Fig. 43: Selecting PDO of type BOOL

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



```
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)
      Frm0Ctrl  >  QB 1520.0, UINT16 [2.0]
      Frm0WcCtrl  >  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]
      Frm0Ctrl  >  QB 1520.0, UINT16 [2.0]
      Frm0WcCtrl  >  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

Offsets
☑ 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 variable “nEL2008_value” sequentially to all eight selected output bits of the EL2008 terminal. In this way it is possible to subsequently address all eight outputs of the terminal in the program with a byte

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

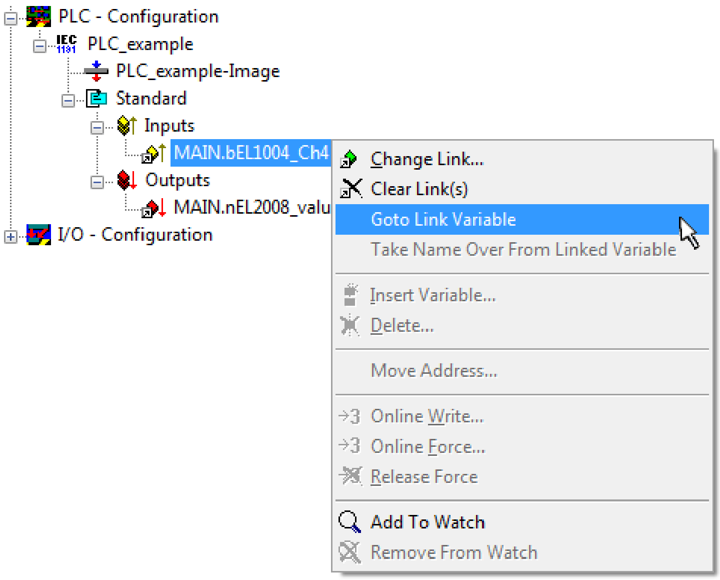
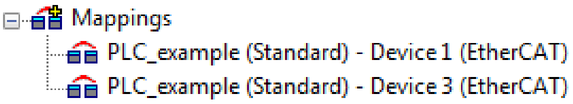


Fig. 45: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

The process of assigning variables to the PDO is completed via the menu selection “Actions” → “Generate Mappings”, key Ctrl+M or by clicking on the symbol in the menu.

This can be visualized in the configuration:


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

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

Activation of the configuration

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

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

Starting the controller

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

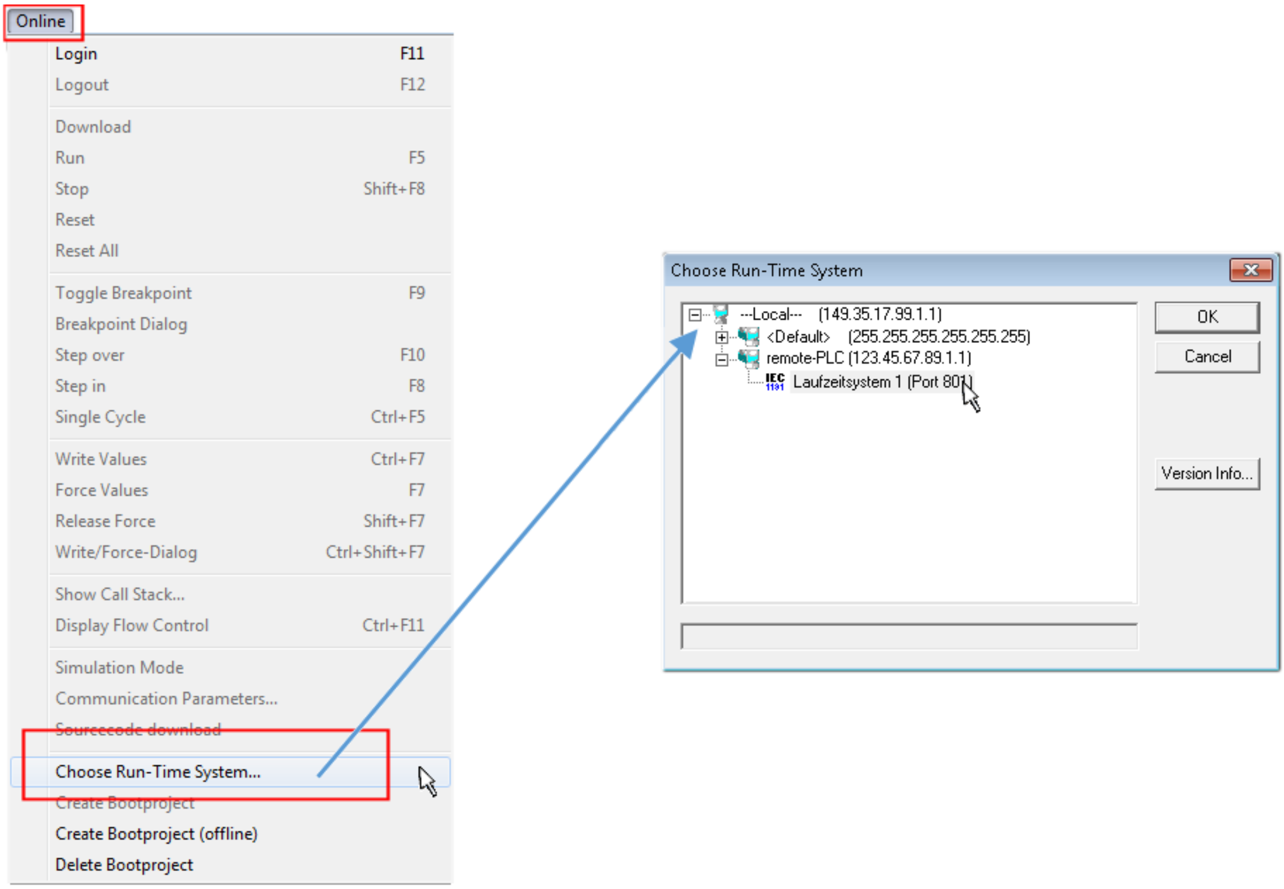
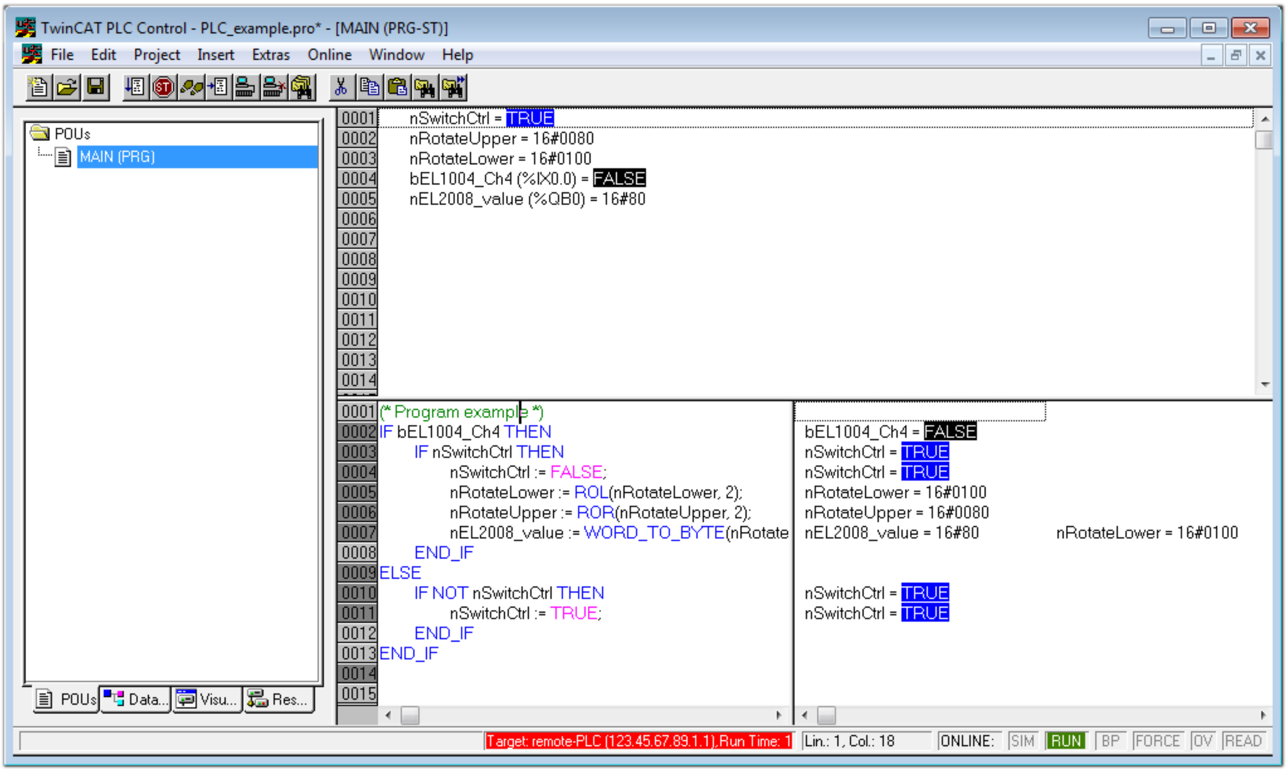


Fig. 46: Choose target system (remote)

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


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

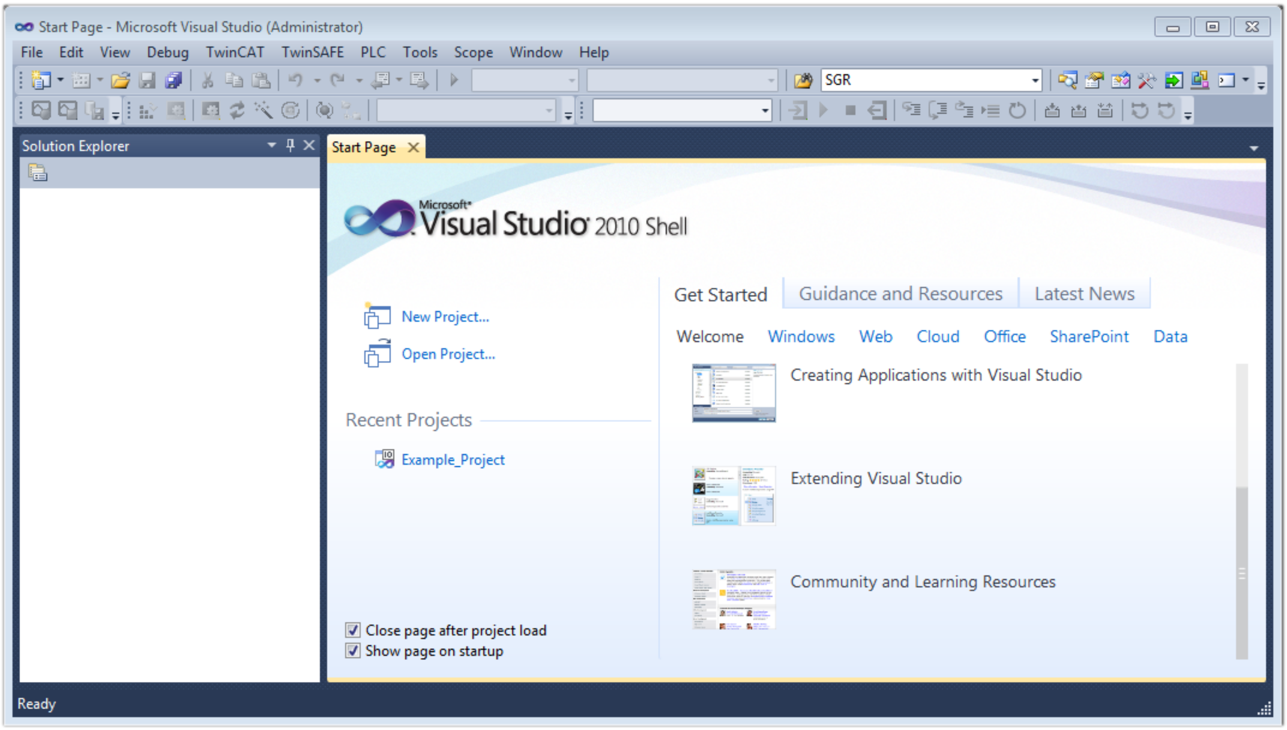
The PLC can now be started via “Online” → “Run”, F5 key or .

5.1.2 TwinCAT 3

Startup

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

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

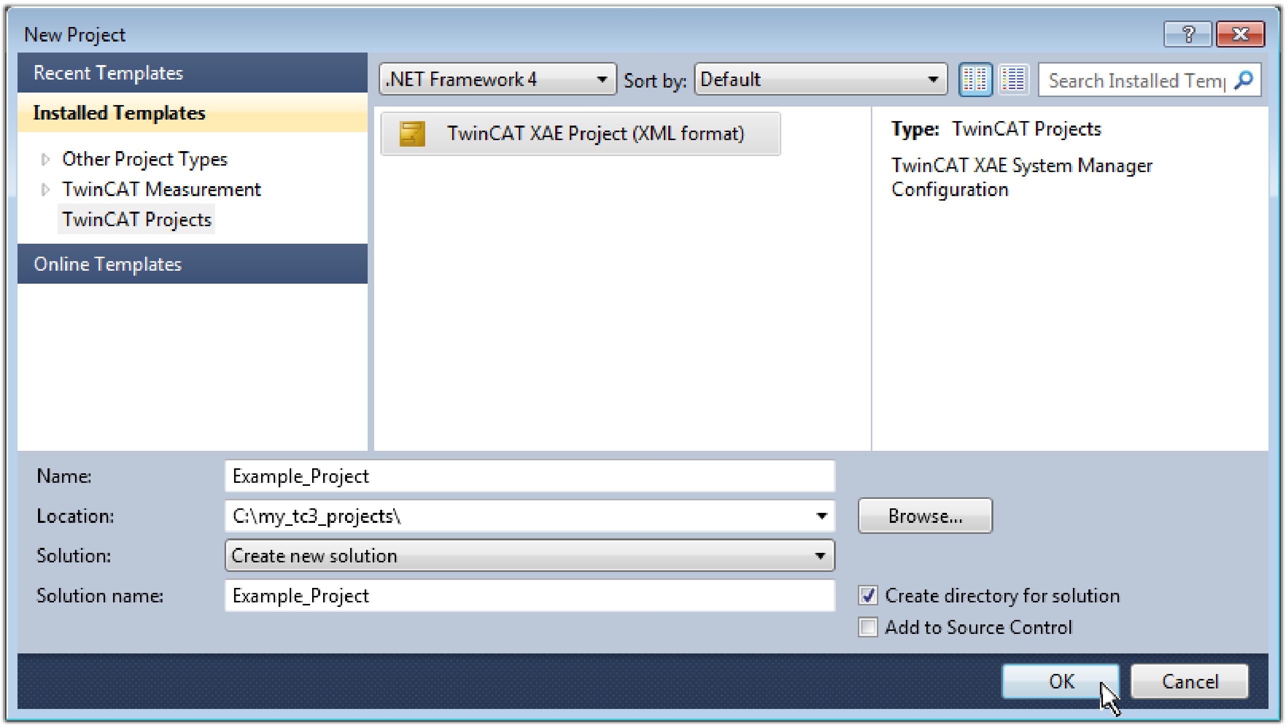

Start Page - Microsoft Visual Studio (Administrator)
File Edit View Debug TwinCAT TwinSAFE PLC Tools Scope Window Help
SGR
Solution Explorer
Start Page
Microsoft® Visual Studio® 2010 Shell
New Project...
Open Project...
Recent Projects
Example_Project
Get Started
Guidance and Resources
Latest News
Welcome 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

☑ Close page after project load

☑ Show page on startup

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):



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 project

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

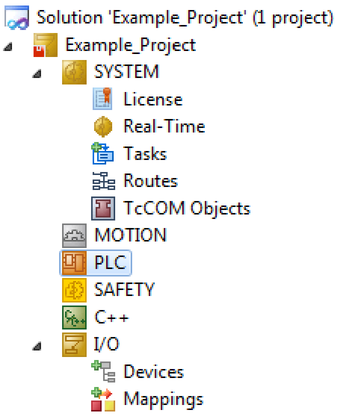
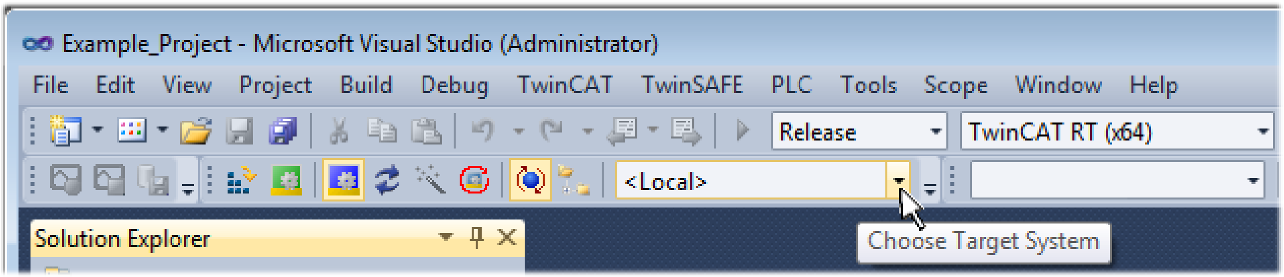


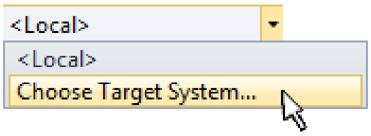
Fig. 50: New TwinCAT3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thereby the next step is "Insert Device [▶ 68]".

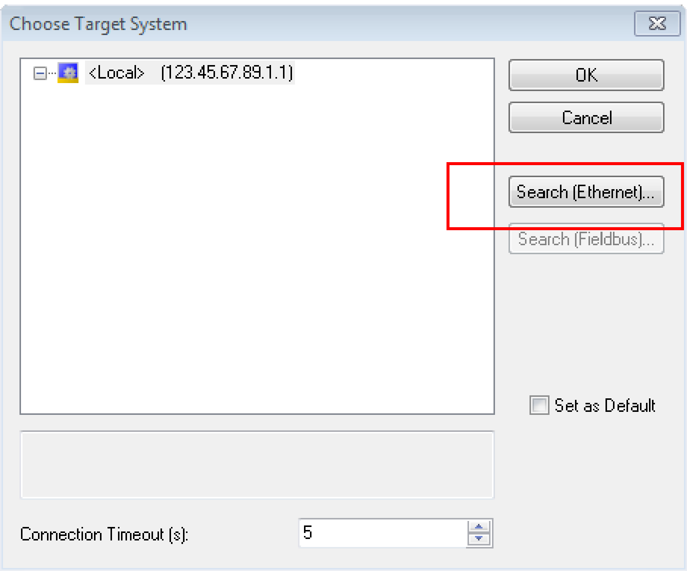
If the intention is to address the TwinCAT runtime environment installed on a PLC as development environment remotely from another system, the target system must be made known first. Via the symbol in the menu bar:



expand the pull-down menu:



and open the following window:

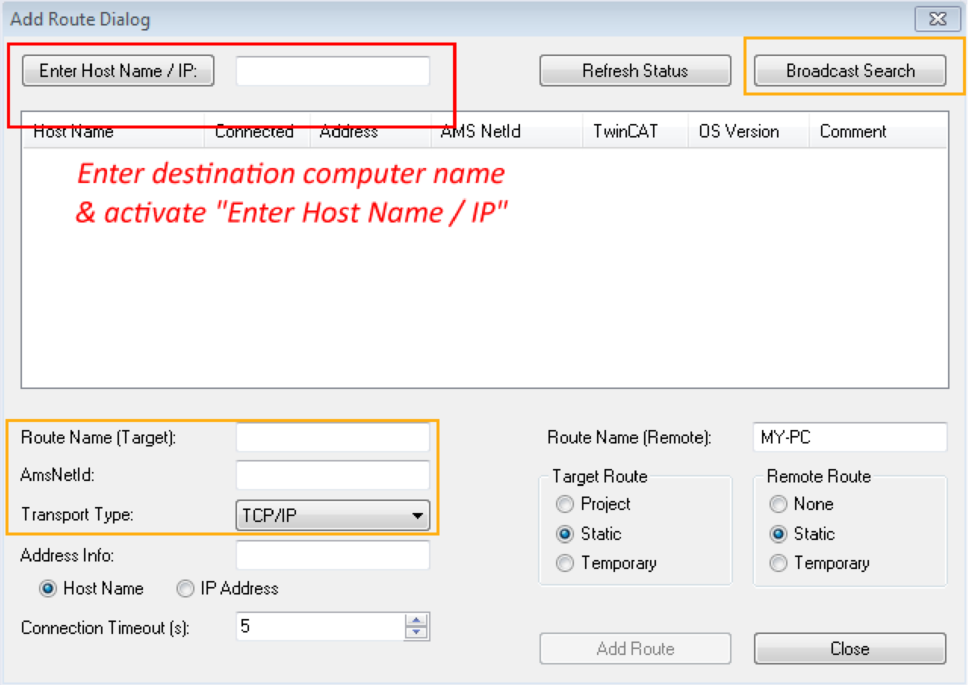


Connection Timeout (s): 5

Fig. 51: Selection dialog: Choose the target system

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

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



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

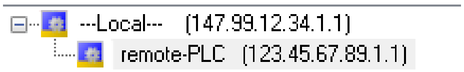
Route Name (Remote): MY-PC

Target Route: ○ Project ● Static ○ Temporary

Remote Route: ○ None ● Static ○ Temporary

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

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



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

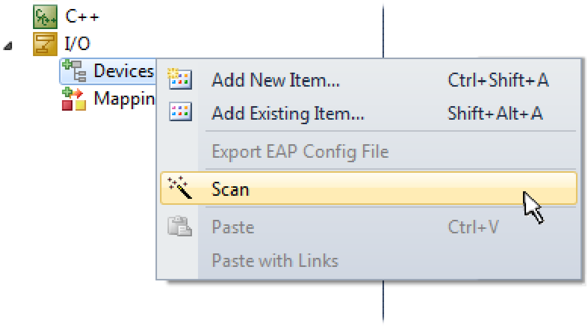
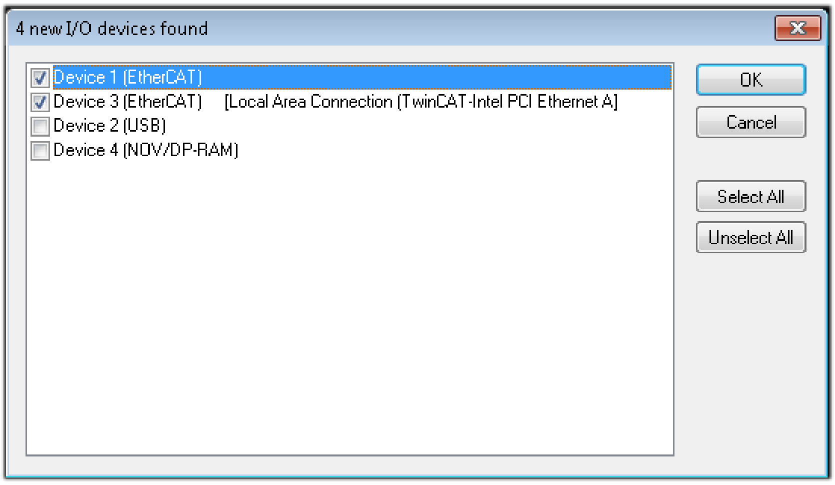


Fig. 53: Select “Scan”

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



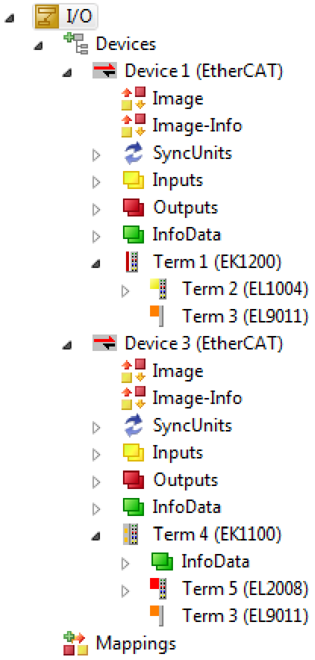
4 new I/O devices found

☑ 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 the devices to be integrated

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

Based on the sample configuration [▸ 53] 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 TwinCAT3 environment

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

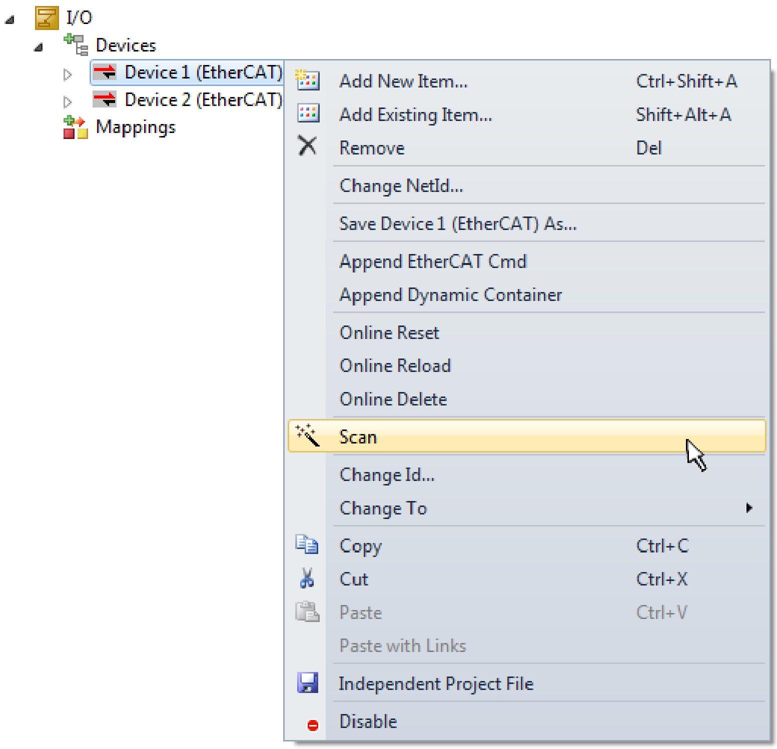


Fig. 56: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

The following section refers to Structured Text (ST).

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

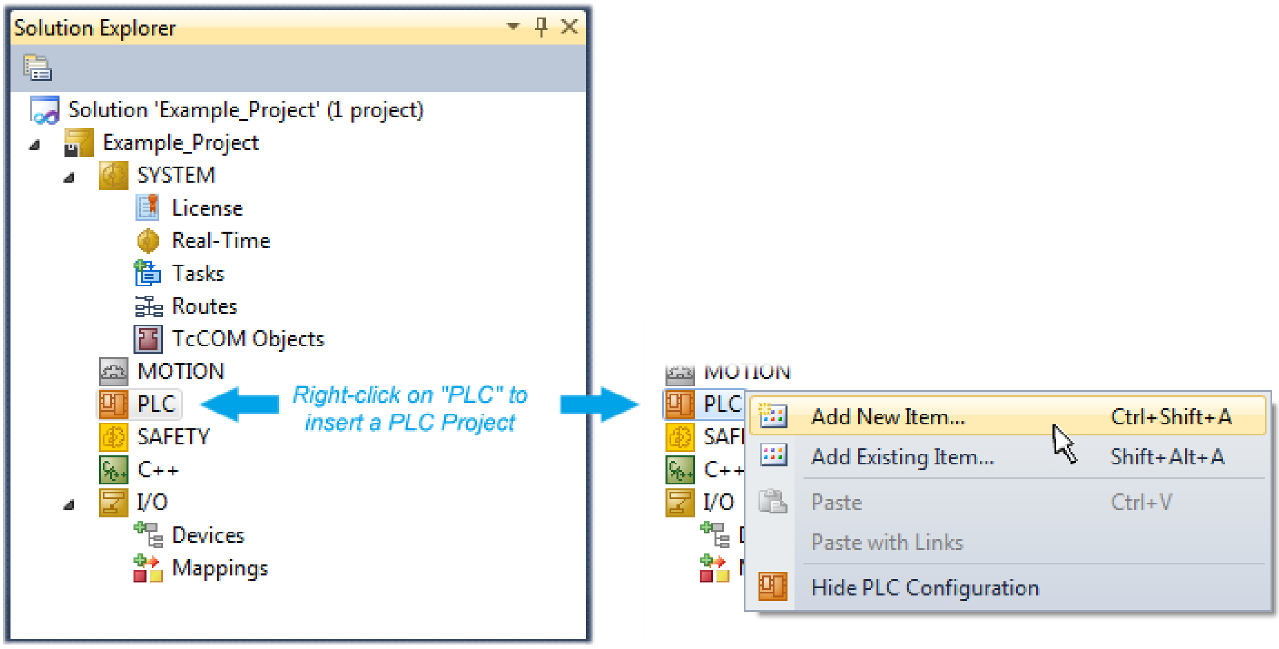
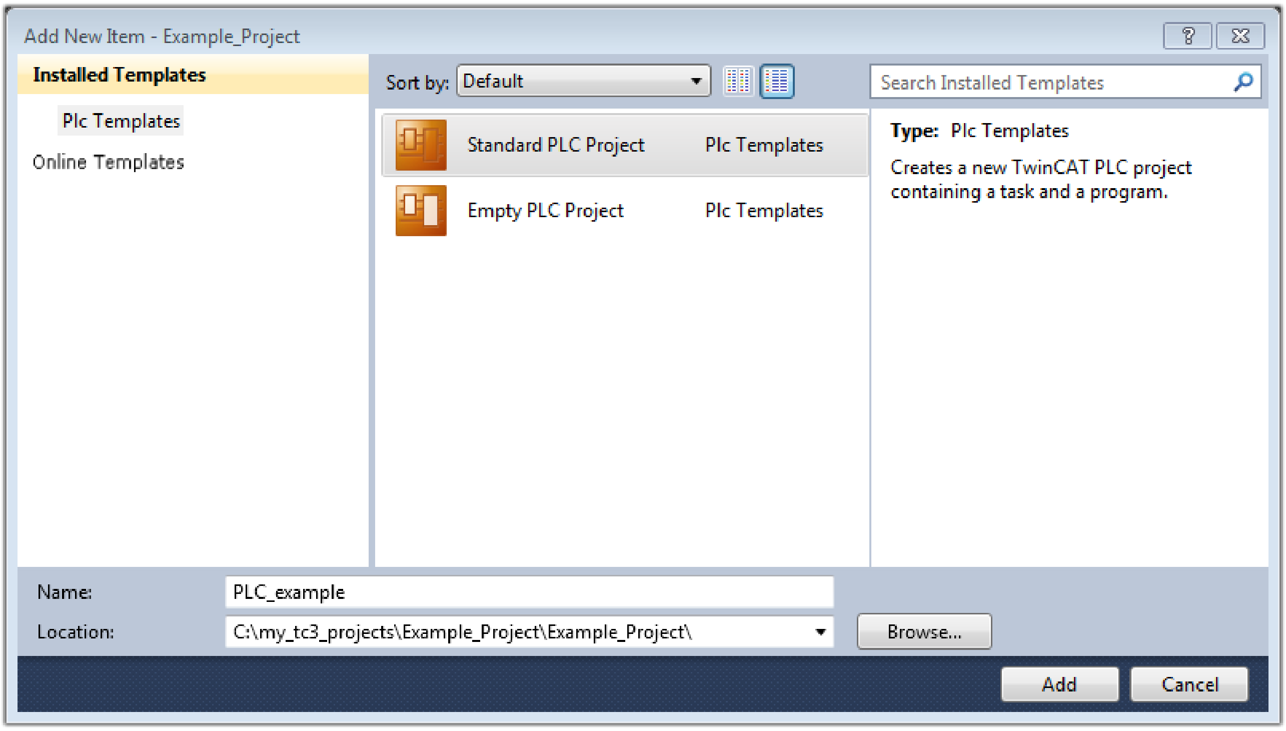


Fig. 57: Adding the programming environment in “PLC”

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
Browse...
Add
Cancel

Type: Plc Templates

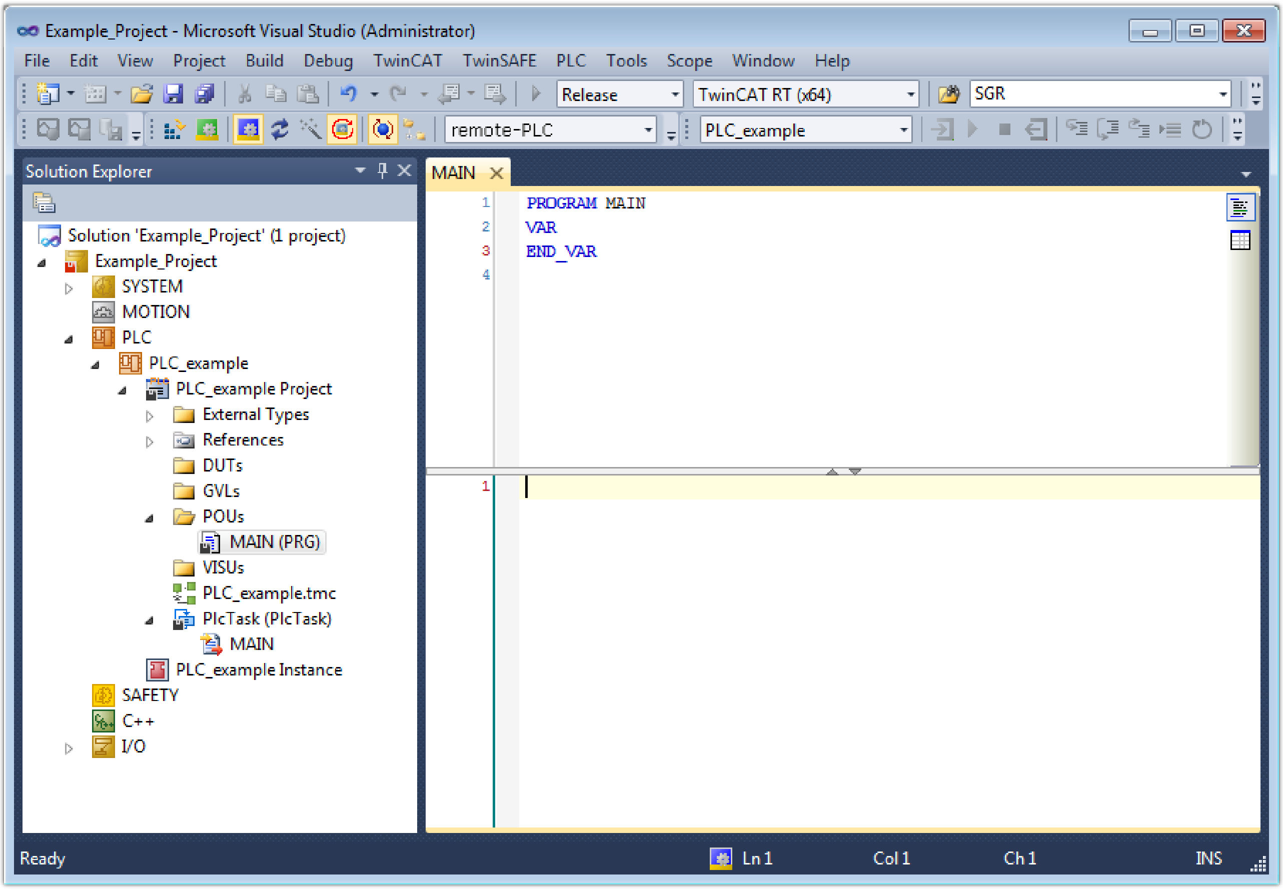
Creates a new TwinCAT PLC project containing a task and a program.

Name: PLC_example

Location: C:\my_tc3_projects\Example_Project\Example_Project\

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

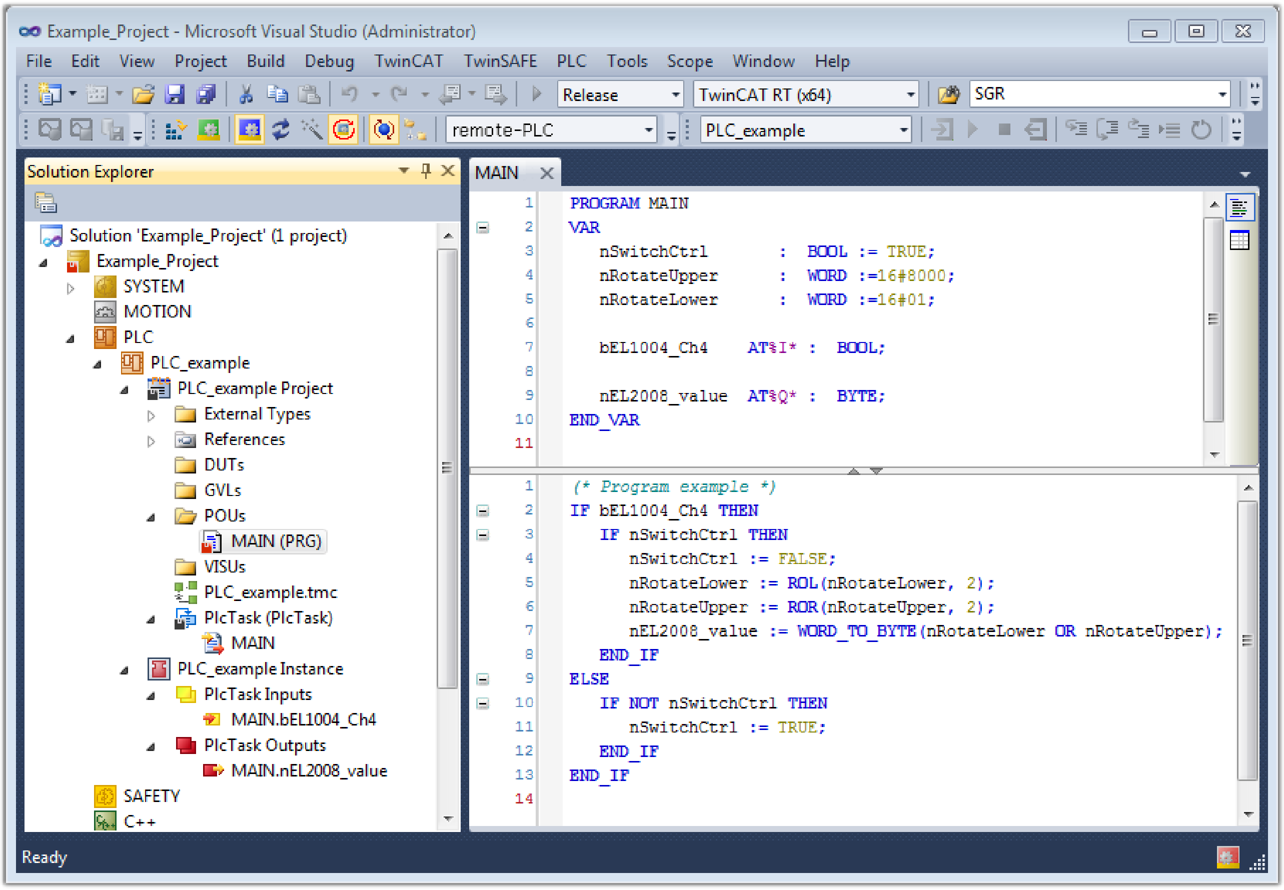
The “Main” program, which already exists by selecting “Standard PLC project”, can be opened by double-clicking on “PLC_example_project” in “POUs”. The following user interface is shown for an initial project:



```
PROGRAM MAIN
VAR
END_VAR
```

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

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



```
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: Sample program with variables after a compile process (without variable integration)

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

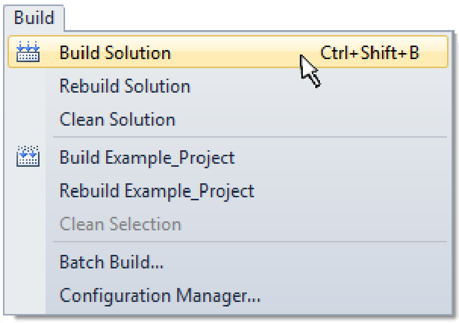
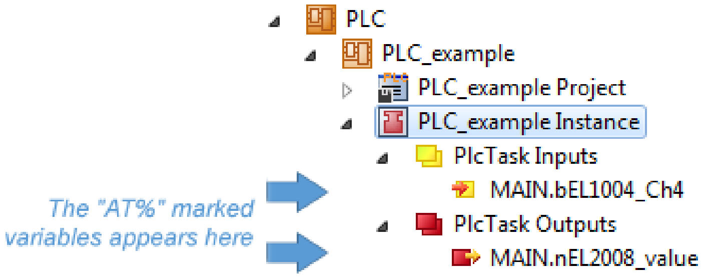


Fig. 61: Start program compilation

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



Assigning variables

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

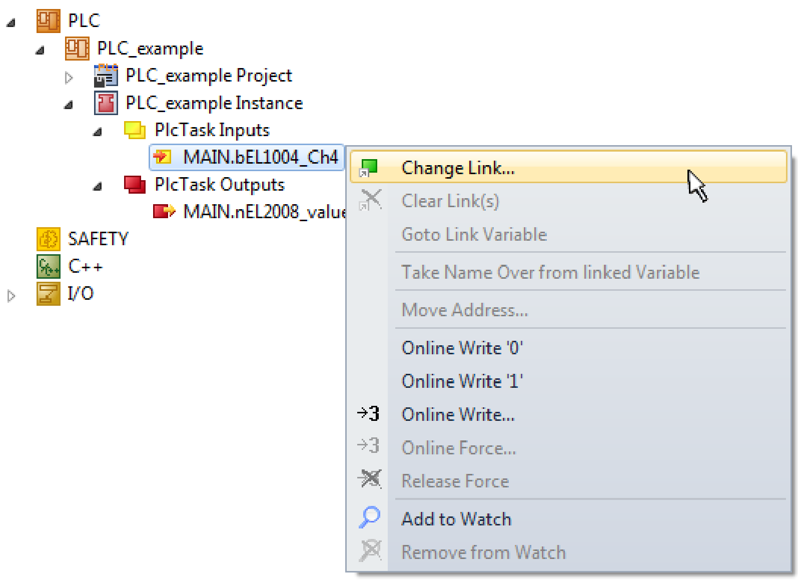
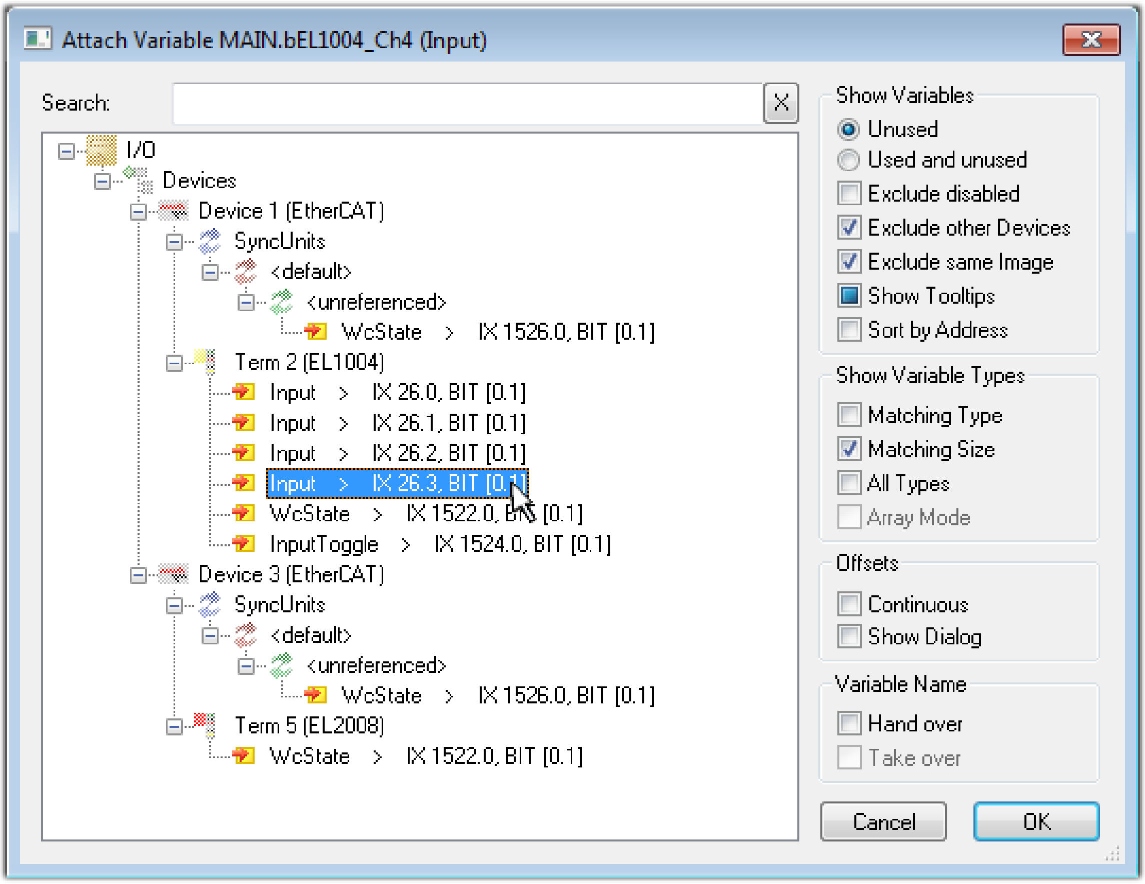


Fig. 62: Creating the links between PLC variables and process objects

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


Attach Variable MAIN.bEL1004_Ch4 (Input)

Search:

```
I/O
  Devices
    Device 1 (EtherCAT)
      SyncUnits
        <default>
          <unreferenced>
            WcState  >  IX 1526.0, BIT [0.1]
      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)
      SyncUnits
        <default>
          <unreferenced>
            WcState  >  IX 1526.0, BIT [0.1]
      Term 5 (EL2008)
        WcState  >  IX 1522.0, BIT [0.1]
```

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

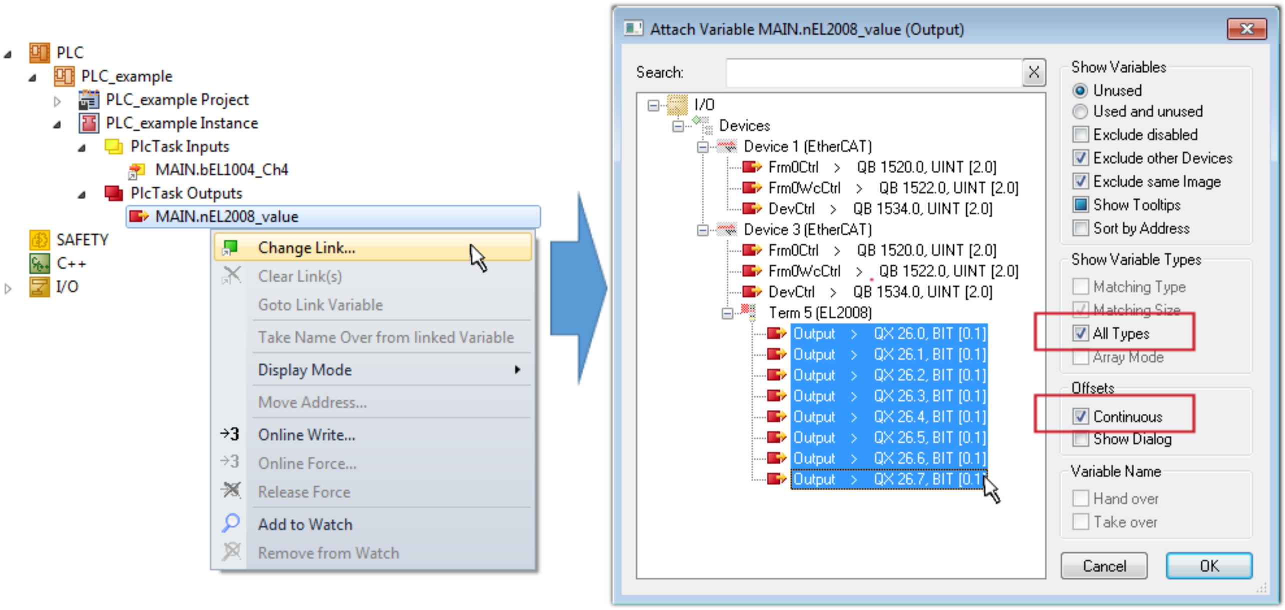
Show Variable Types
□ Matching Type
☑ Matching Size
□ All Types
□ Array Mode

Offsets
□ Continuous
□ Show Dialog

Variable Name
□ Hand over
□ Take over

Fig. 63: Selecting PDO of type BOOL

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



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]
```

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

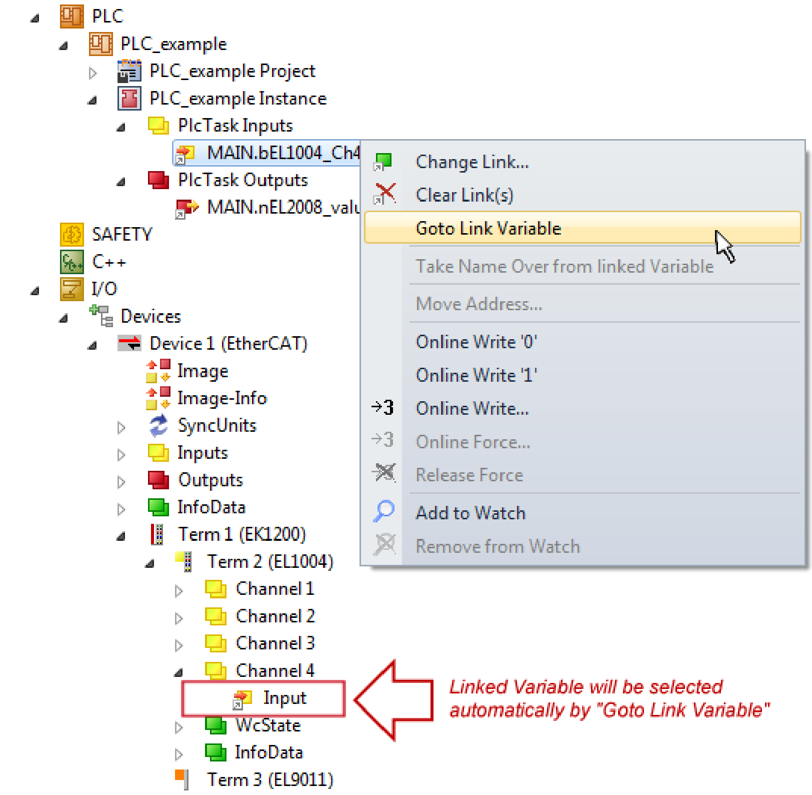
Offsets
☑ Continuous
□ Show Dialog

Variable Name
□ Hand over
□ Take over

Cancel OK

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



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

Fig. 65: Application of a “Goto Link” variable, using “MAIN.bEL1004_Ch4” as a sample

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

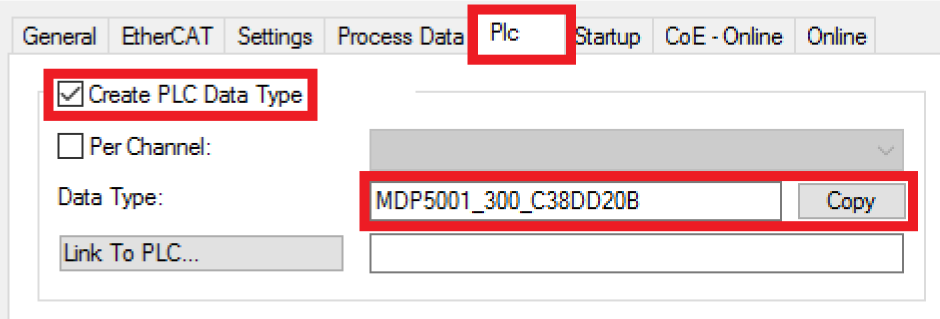
Note on the type of variable assignment

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

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

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

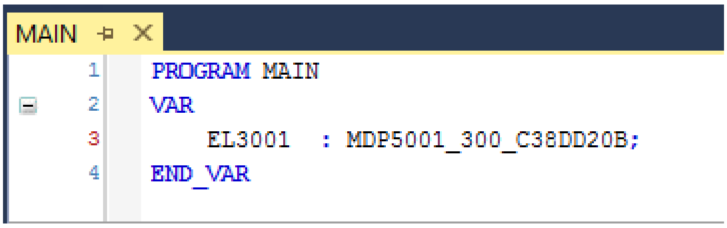
1. First the required process data must be selected in the “Process data” tab in TwinCAT.
2. After that, the PLC data type must be generated in the tab “PLC” via the check box.
3. The data type in the “Data Type” field can then be copied using the “Copy” button.



Data Type: MDP5001_300_C38DD20B

Fig. 66: Creating a PLC data type

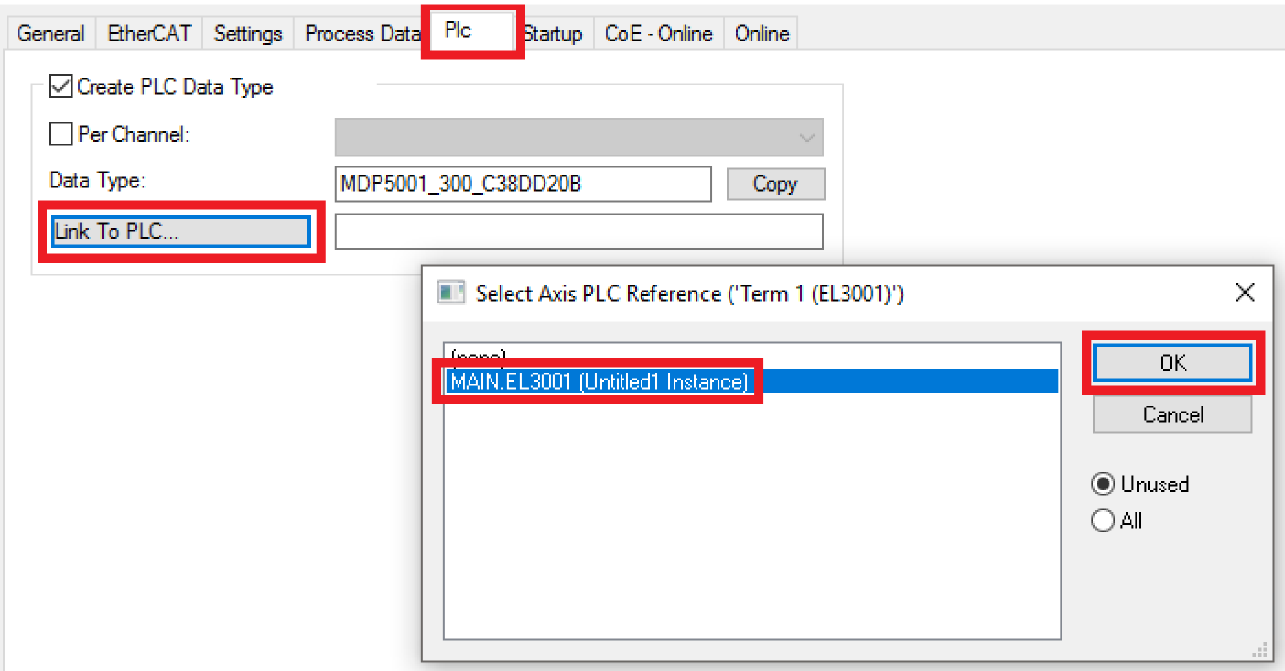
4. An instance of the data structure of the copied data type must then be created in the PLC.



```
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

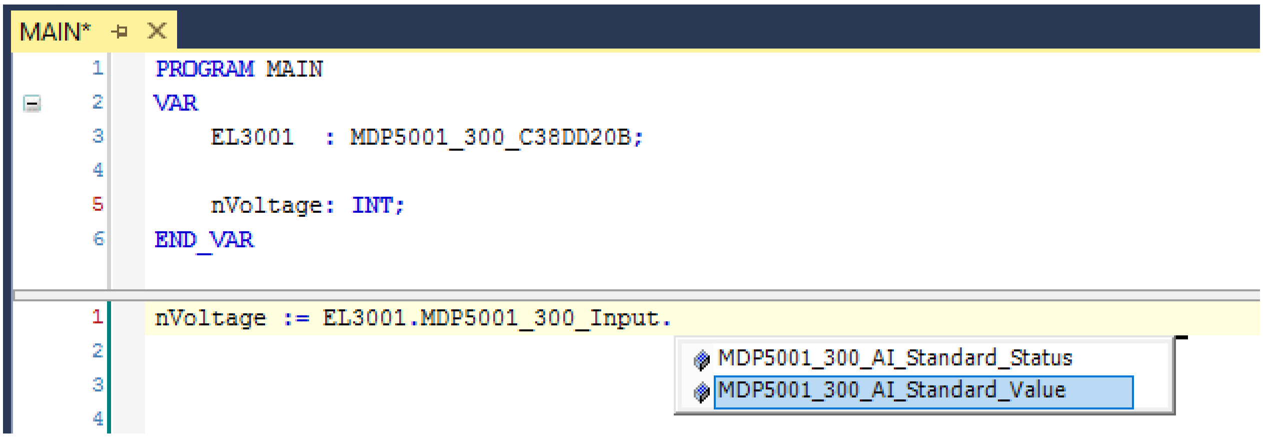
Select Axis PLC Reference ('Term 1 (EL3001)')

(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.



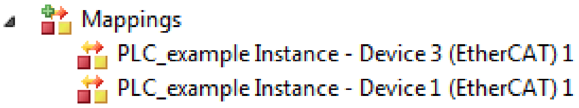
```
PROGRAM MAIN
VAR
    EL3001  : MDP5001_300_C38DD20B;

    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 settings of the development environment to the runtime system. Confirm the messages “Old configurations are overwritten!” and “Restart TwinCAT system in Run mode” with “OK”. The corresponding assignments can be seen in the project folder explorer:

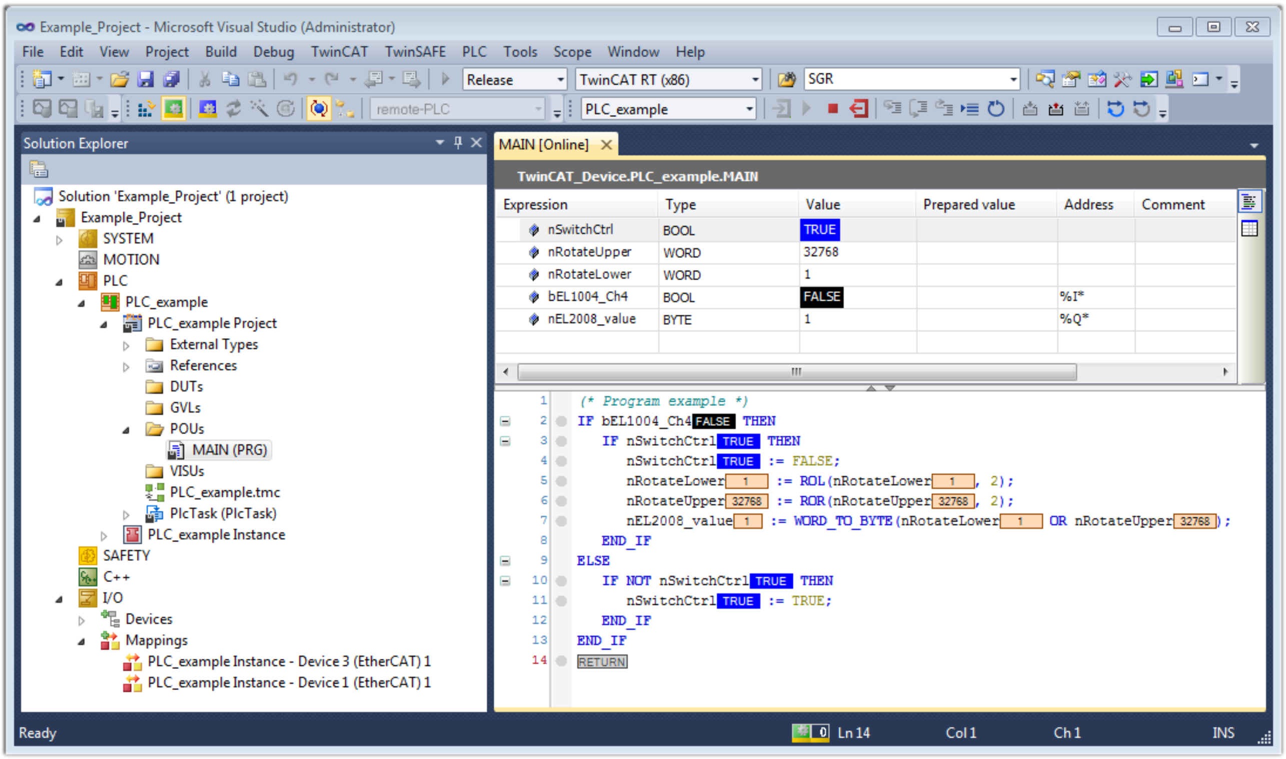


Mappings
PLC_example Instance - Device 3 (EtherCAT) 1
PLC_example Instance - Device 1 (EtherCAT) 1

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



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 development environment (VS shell): logged-in, after program startup

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

5.2 TwinCAT Development Environment

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

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

Details:

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

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

Additional features:

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

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

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

5.2.1 Installation of the TwinCAT real-time driver

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

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

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

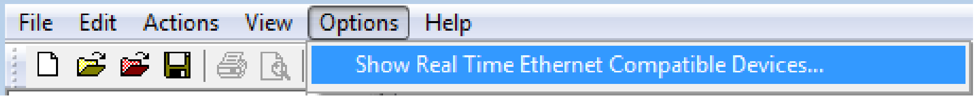


Fig. 71: System Manager “Options” (TwinCAT 2)

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

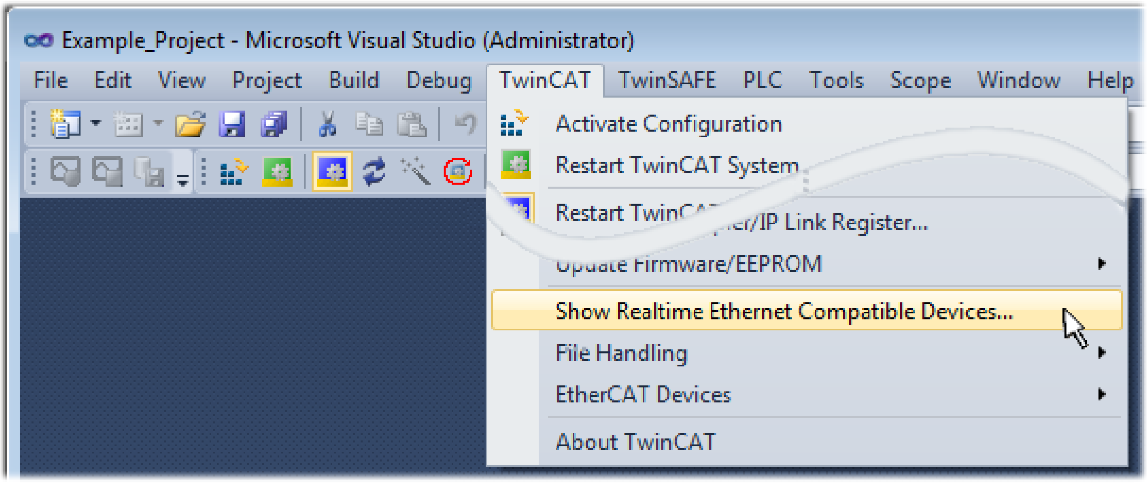
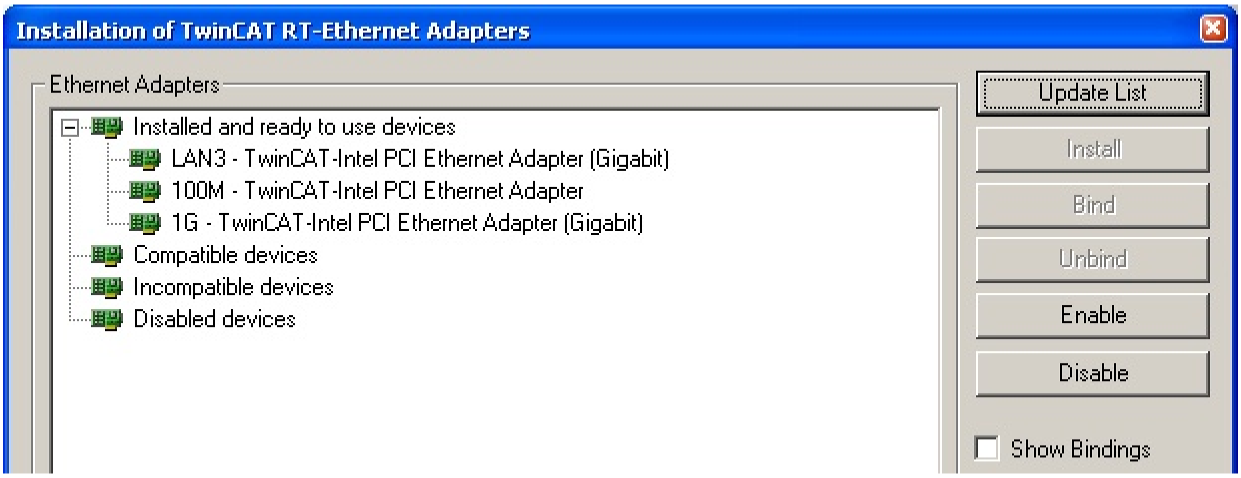


Fig. 72: Call up under VS Shell (TwinCAT 3)

The following dialog appears:



Installation of TwinCAT RT-Ethernet Adapters

Ethernet Adapters

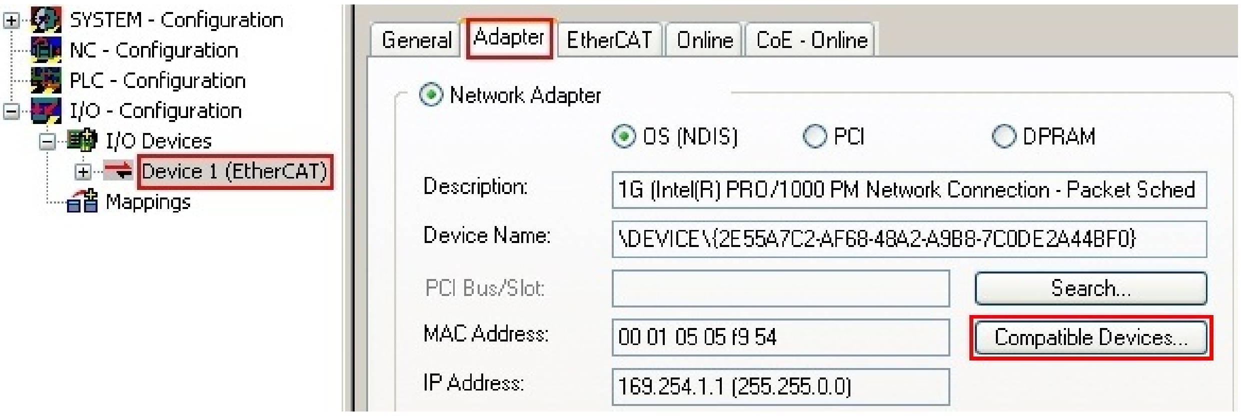
- Installed and ready to use devices
  - LAN3 - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
  - 100M - TwinCAT-Intel PCI Ethernet Adapter
  - 1G - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
- Compatible devices
- Incompatible devices
- Disabled devices

Fig. 73: Overview of network interfaces

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

A Windows warning regarding the unsigned driver can be ignored.

Alternatively an EtherCAT-device can be inserted first of all as described in chapter Offline configuration creation, section “Creating the EtherCAT device” [▶ 89] in order to view the compatible ethernet ports via its EtherCAT properties (tab “Adapter”, button “Compatible Devices…”):



- SYSTEM - Configuration
- NC - Configuration
- PLC - Configuration
- I/O - Configuration
  - I/O Devices
    - Device 1 (EtherCAT)
  - Mappings

● 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}

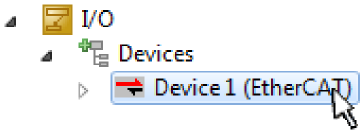
PCI Bus/Slot:

MAC Address: 00 01 05 05 f9 54

IP Address: 169.254.1.1 (255.255.0.0)

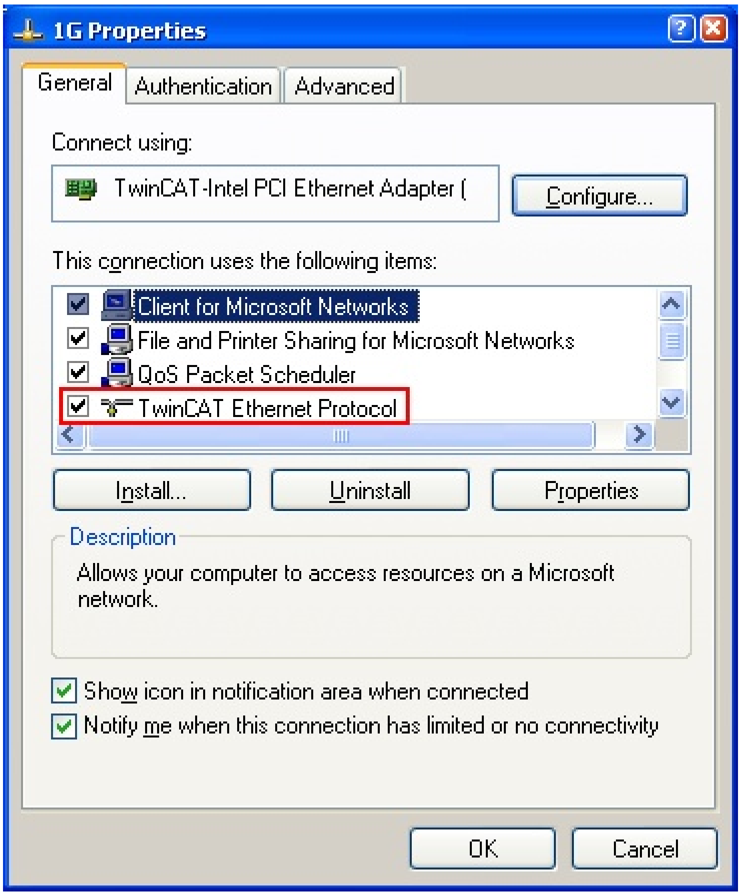
Fig. 74: EtherCAT device properties(TwinCAT 2): click on “Compatible Devices…” of tab “Adapte””

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



- I/O
  - Devices
    - Device 1 (EtherCAT)

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:

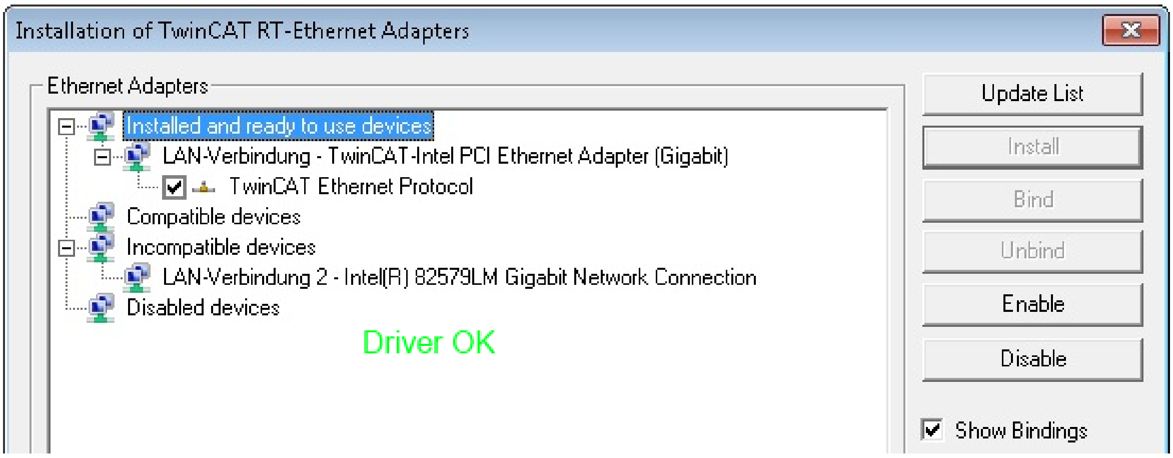
- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ TwinCAT Ethernet Protocol

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. 75: Windows properties of the network interface

A correct setting of the driver could be:

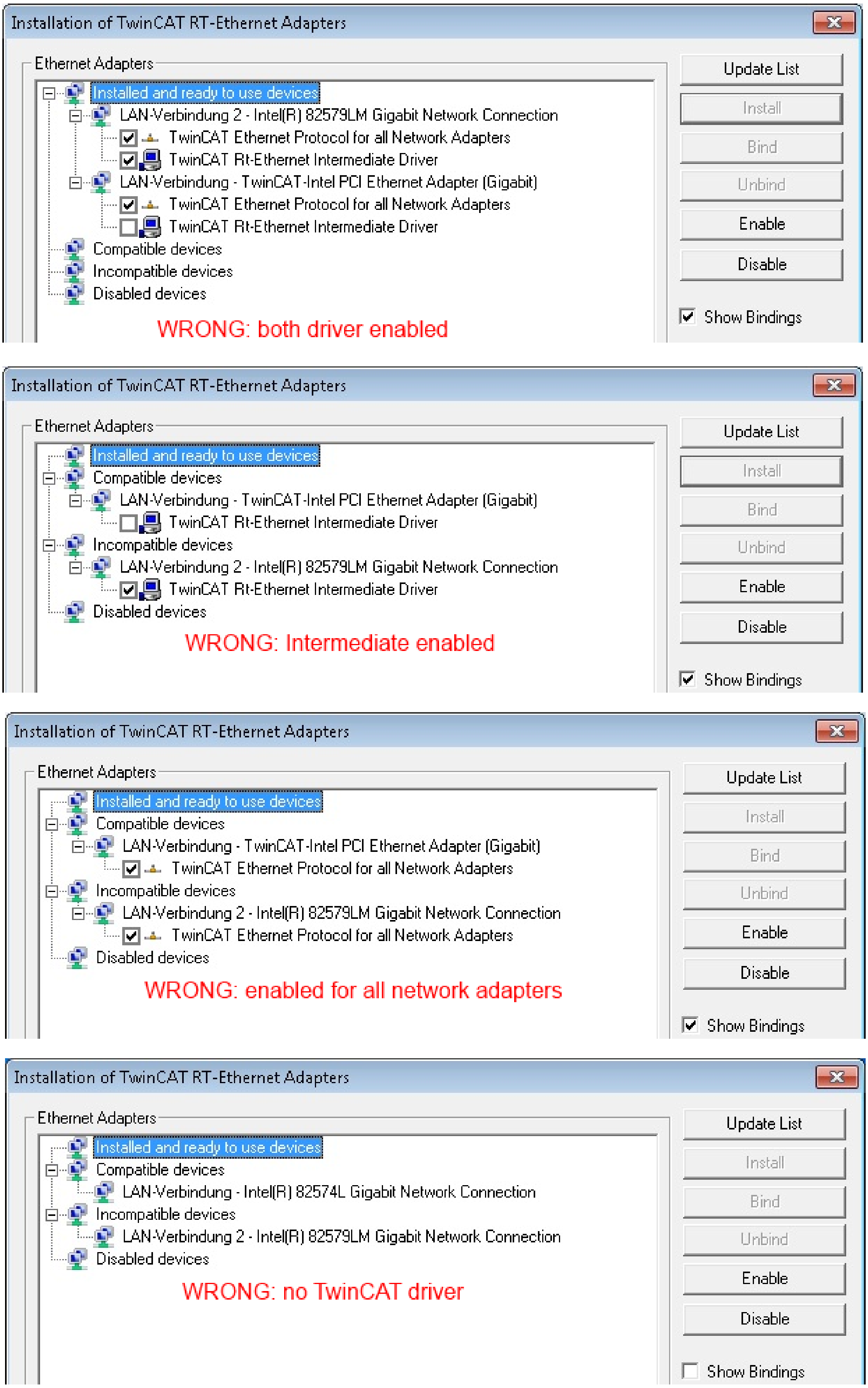


- Installed and ready to use devices
  - LAN-Verbindung - TwinCAT-Intel PCI Ethernet Adapter (Gigabit)
    - ☑ TwinCAT Ethernet Protocol
- Compatible devices
- Incompatible devices
  - LAN-Verbindung 2 - Intel(R) 82579LM Gigabit Network Connection
- Disabled devices

☑ Show Bindings

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

Other possible settings have to be avoided:



Installation of TwinCAT RT-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

- 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

- 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

- 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

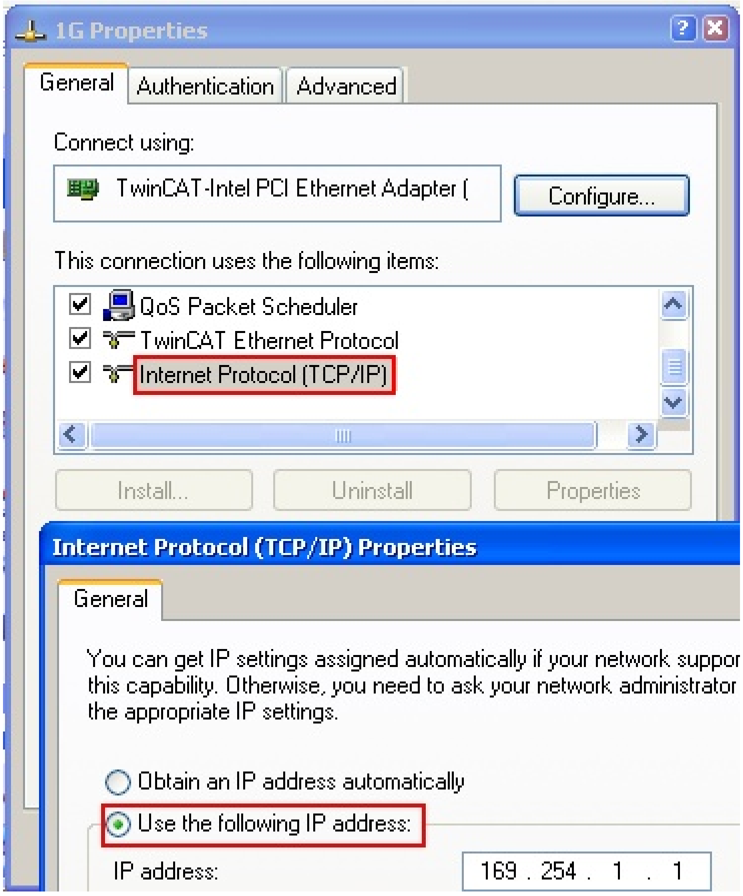
☐ Show Bindings

Fig. 77: 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.



Connect using:

TwinCAT-Intel PCI Ethernet Adapter (

This connection uses the following items:

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

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. 78: 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 [▶ 88] is available for this purpose.

ESI

The *.xml files are associated with *.xsd files, which describe the structure of the ESI XML files. To update the ESI device descriptions, both file types should therefore be updated.

Device differentiation

EtherCAT devices/slaves are distinguished by four properties, which determine the full device identifier. For example, the device identifier EL2521-0025-1018 consists of:

- family key “EL”
- name “2521”
- type “0025”
- and revision “1018”

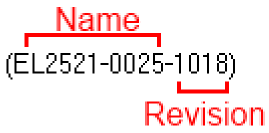
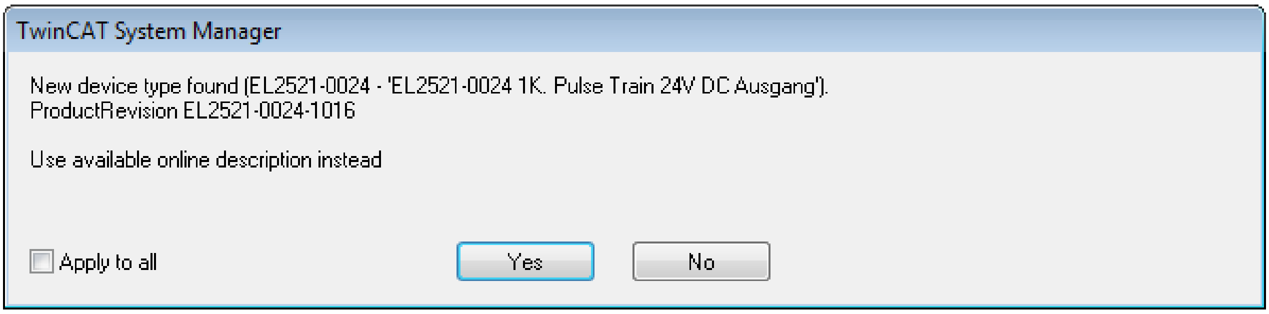


Fig. 79: Identifier structure

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



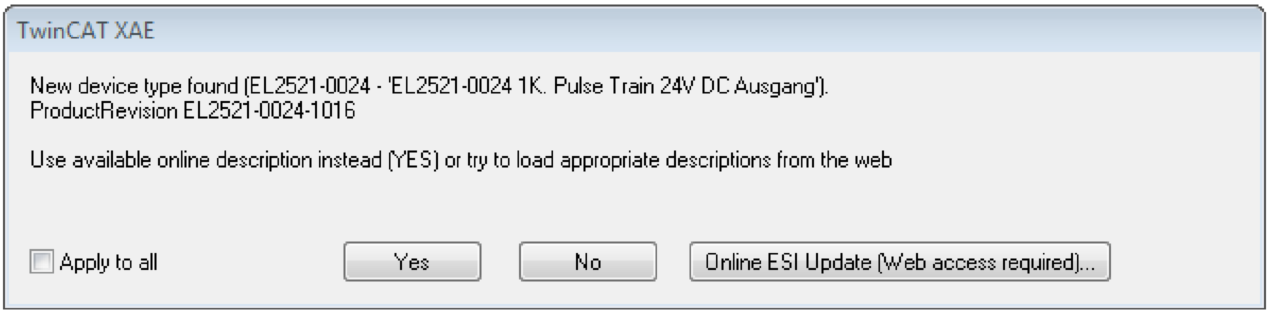
TwinCAT System Manager

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

Use available online description instead

Fig. 80: 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. 81: Information window OnlineDescription (TwinCAT 3)

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

NOTE

Changing the "usual" configuration through a scan

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

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

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

Refer in particular to the chapter "General notes on the use of Beckhoff EtherCAT IO components" and for manual configuration to the chapter "Offline configuration creation [▶ 89]".

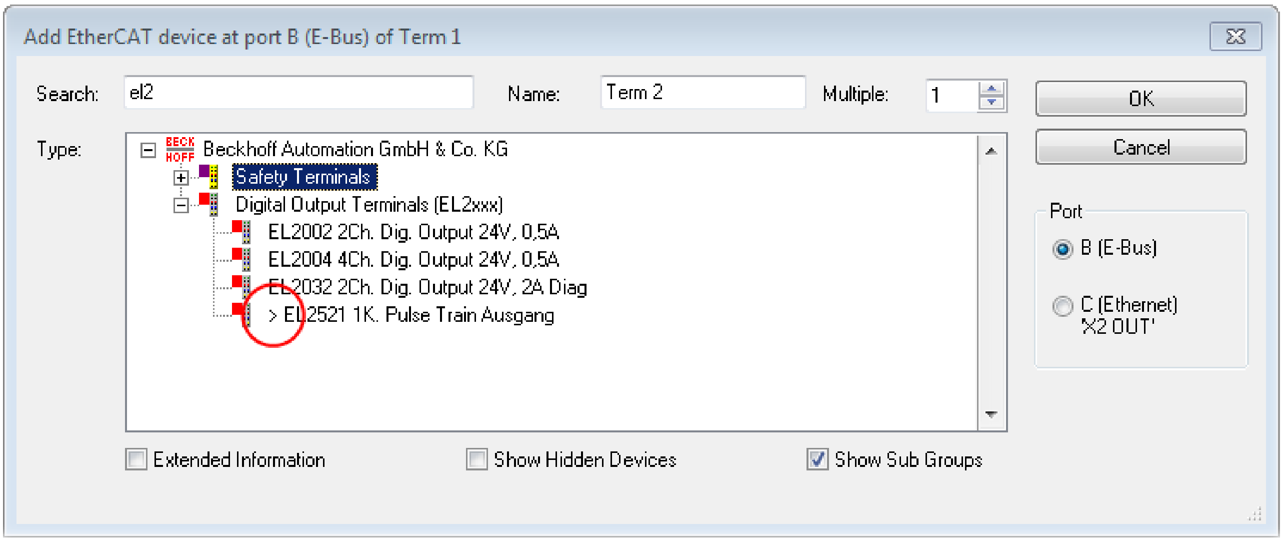
If the OnlineDescription is used regardless, the System Manager reads a copy of the device description from the EEPROM in the EtherCAT slave. In complex slaves the size of the EEPROM may not be sufficient for the complete ESI, in which case the ESI would be *incomplete* in the configurator. Therefore it's recommended using an offline ESI file with priority in such a case.

The System Manager creates for online recorded device descriptions a new file "OnlineDescription0000...xml" in its ESI directory, which contains all ESI descriptions that were read online.

OnlineDescriptionCache00000002.xml

Fig. 82: File OnlineDescription.xml created by the System Manager

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

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

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

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

OnlineDescription for TwinCAT 3.x

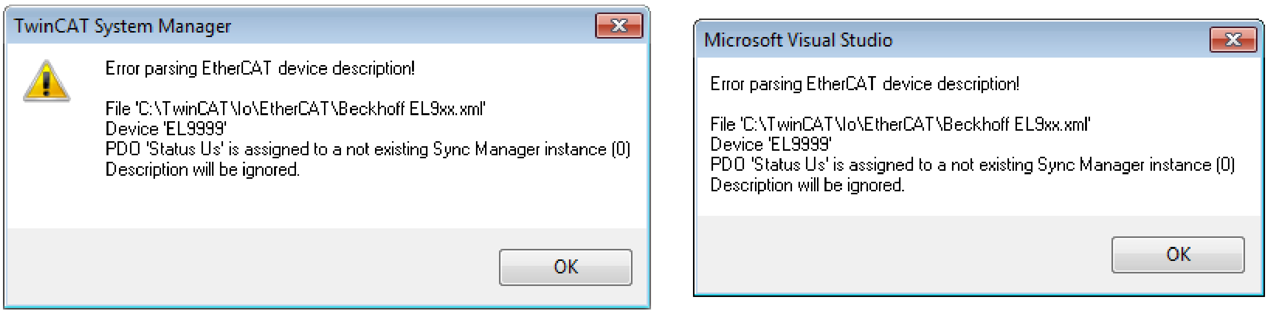
In addition to the file described above “OnlineDescription0000...xml”, a so called EtherCAT cache with new discovered devices is created by TwinCAT 3.x, e.g. under Windows 7:

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

Reasons may include:

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

5.2.3 TwinCAT ESI Updater

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

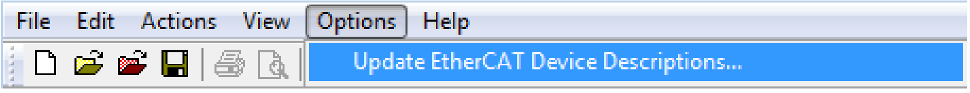
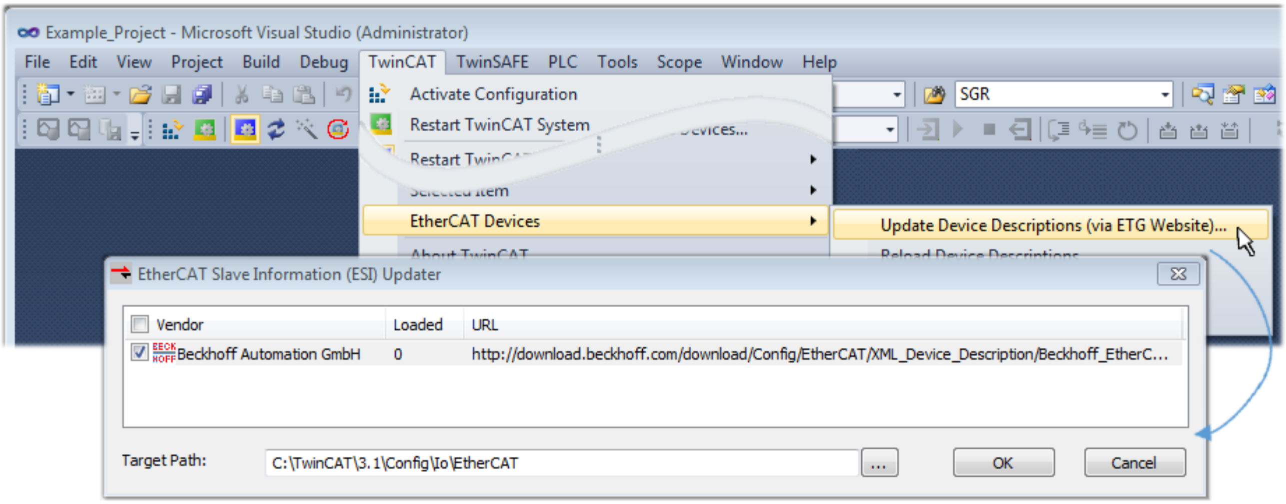


Fig. 85: Using the ESI Updater (>= TwinCAT 2.11)

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

Selection under TwinCAT 3:



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

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

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

5.2.4 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” [▶ 84].

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 [▶ 94] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 95]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 98]

The scan with existing configuration [▶ 99] 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.

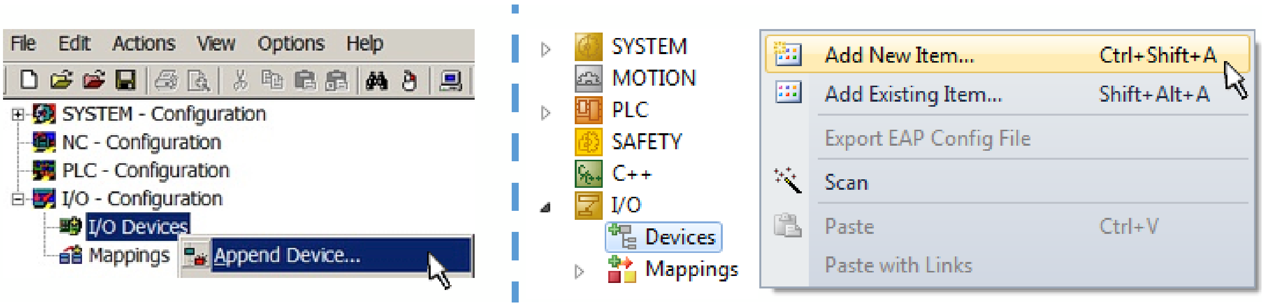


Fig. 87: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

Select type “EtherCAT” for an EtherCAT I/O application with EtherCAT slaves. For the present publisher/subscriber service in combination with an EL6601/EL6614 terminal select “EtherCAT Automation Protocol via EL6601”.

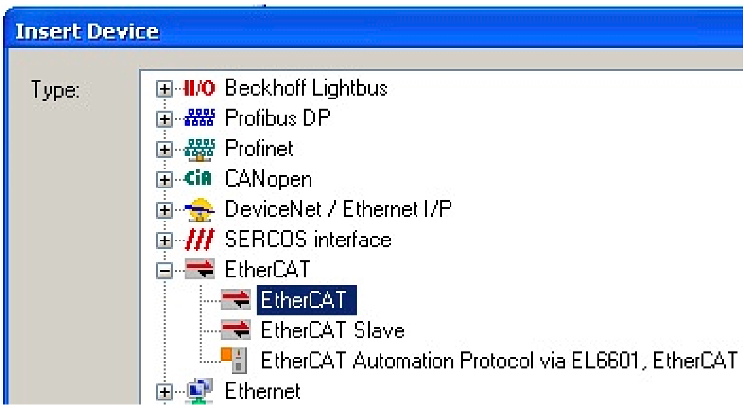
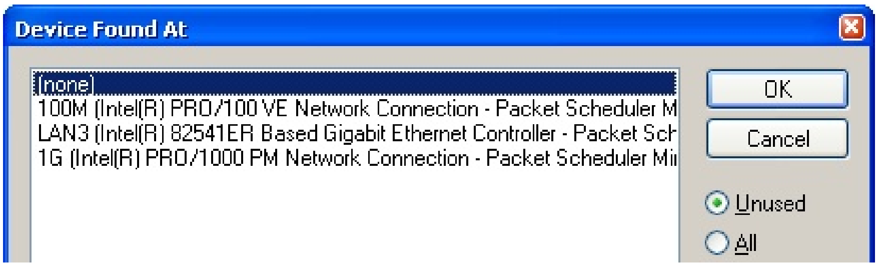


Fig. 88: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

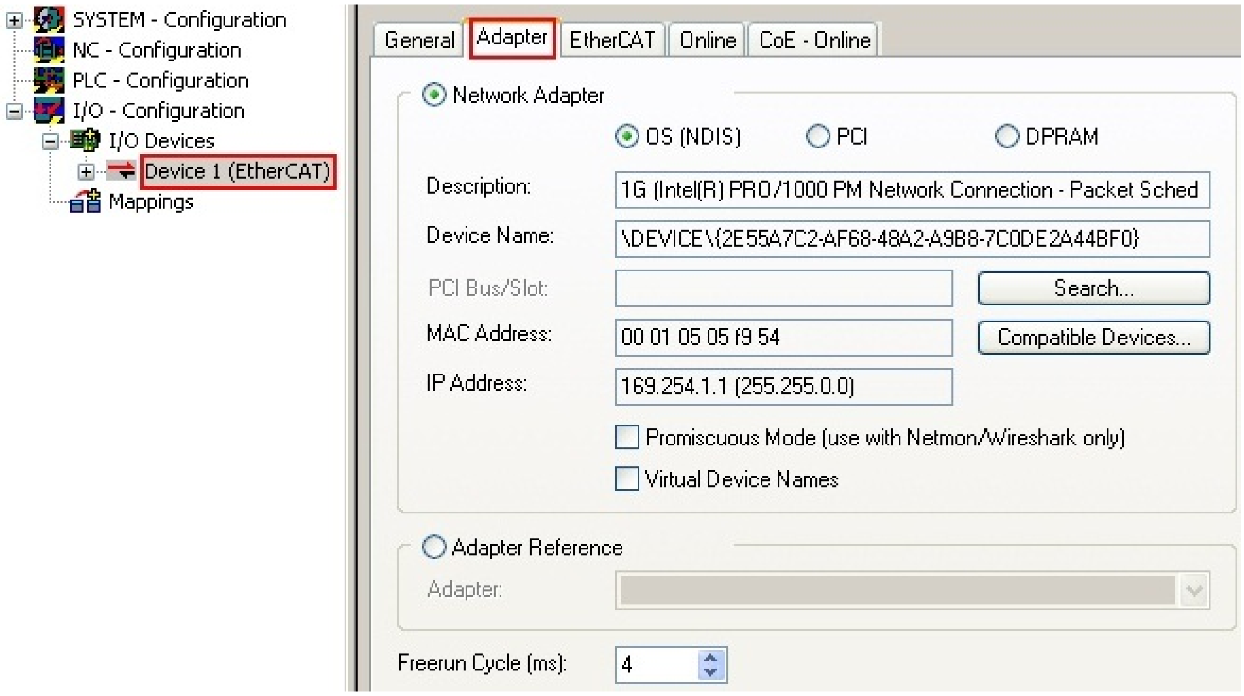
Then assign a real Ethernet port to this virtual device in the runtime system.



(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

Fig. 89: Selecting the Ethernet port

This query may appear automatically when the EtherCAT device is created, or the assignment can be set/modified later in the properties dialog; see Fig. “EtherCAT device properties (TwinCAT 2)”.



● 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

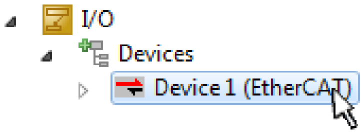
○ Adapter Reference

Adapter:

Freerun Cycle (ms): 4

Fig. 90: EtherCAT device properties (TwinCAT 2)

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



Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 79].

Defining EtherCAT slaves

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

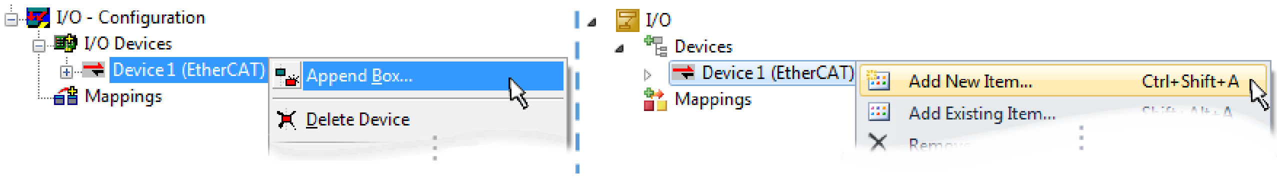


Fig. 91: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

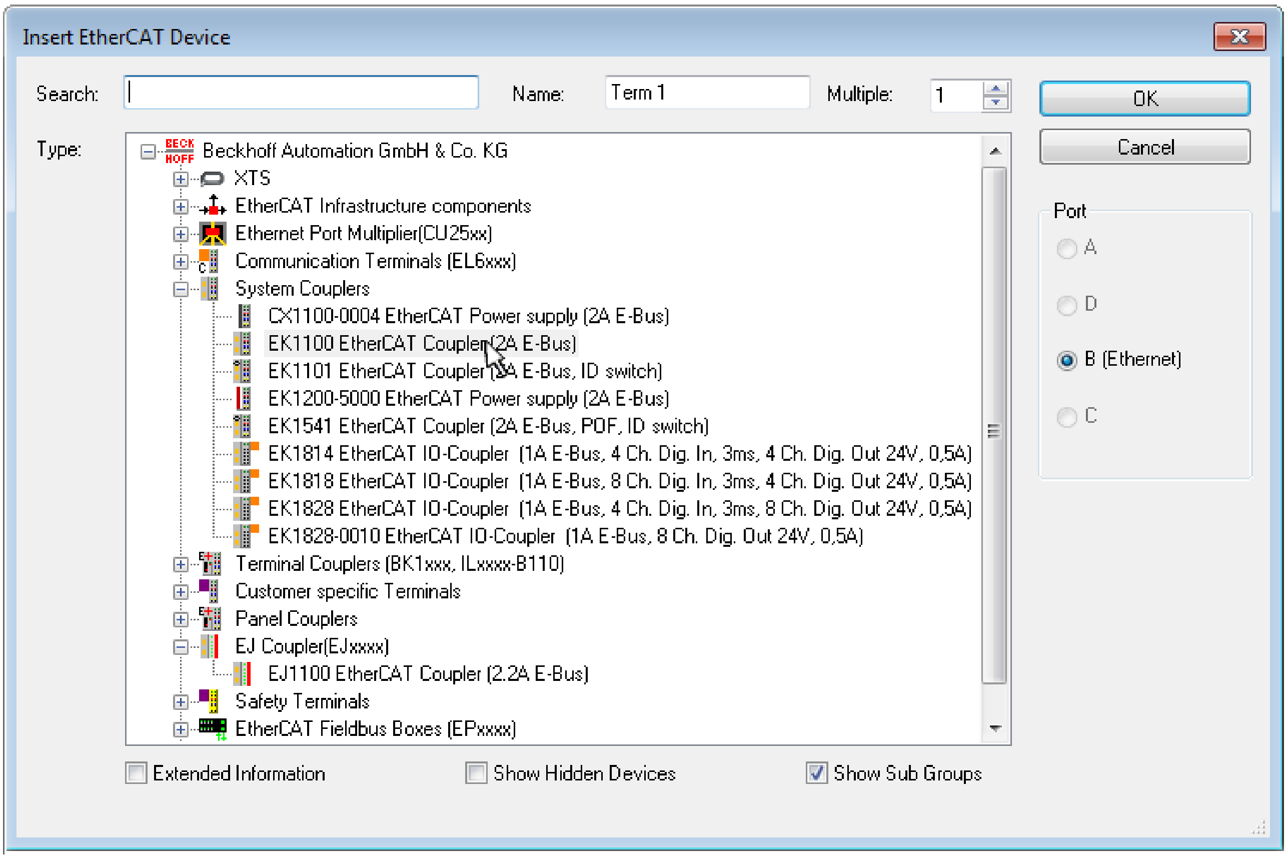
Only devices are offered for selection that can be appended to the previously selected device. Therefore the physical layer available for this port is also displayed (Fig. “Selection dialog for new EtherCAT device”, A). In the case of cable-based Fast-Ethernet physical layer with PHY transfer, then also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

Overview of physical layer

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

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

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


Insert EtherCAT Device

Search:

Name: Term 1

Multiple: 1

Type:

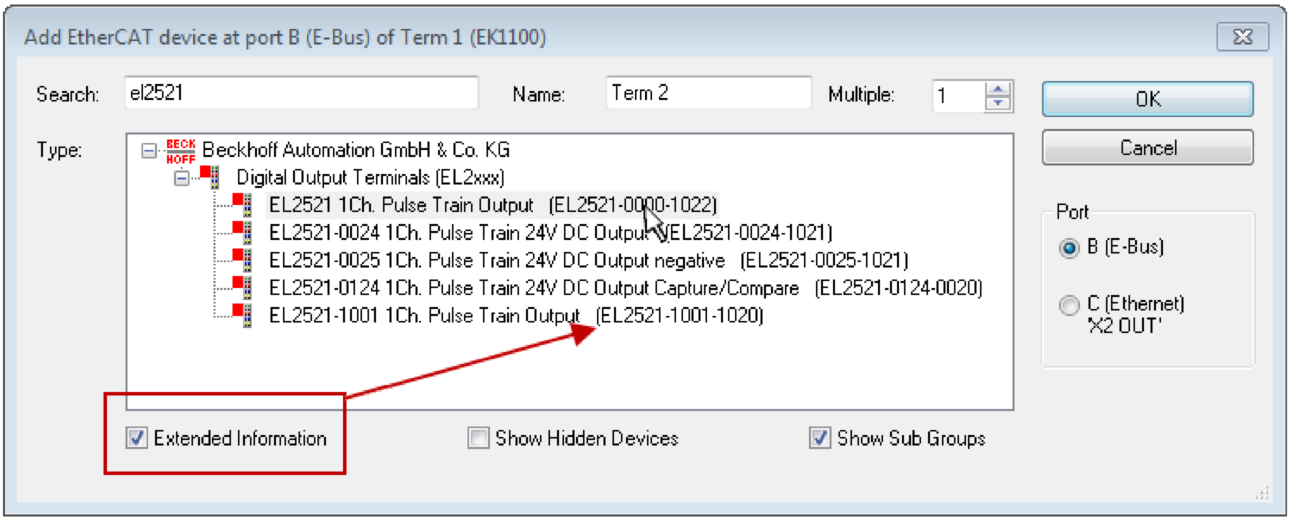
- Beckhoff Automation GmbH & Co. KG
  - XTS
  - EtherCAT Infrastructure components
  - Ethernet Port Multiplier(CU25xx)
  - Communication Terminals (EL6xxx)
  - System Couplers
    - CX1100-0004 EtherCAT Power supply (2A E-Bus)
    - EK1100 EtherCAT Coupler (2A E-Bus)
    - EK1101 EtherCAT Coupler (2A 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. 92: Selection dialog for new EtherCAT device

By default only the name/device type is used as selection criterion. For selecting a specific revision of the device the revision can be displayed as “Extended Information”.


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

Search: el2521

Name: Term 2

Multiple: 1

Type:

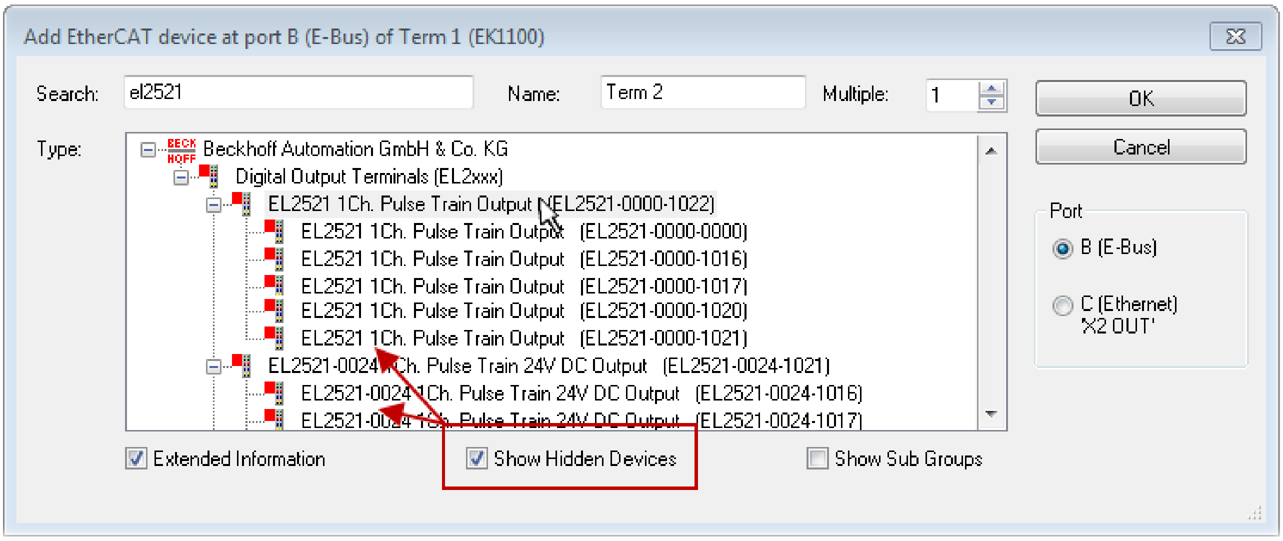
- Beckhoff Automation GmbH & Co. KG
  - Digital Output Terminals (EL2xxx)
    - EL2521 1Ch. Pulse Train Output (EL2521-0000-1022)
    - EL2521-0024 1Ch. Pulse Train 24V DC Output (EL2521-0024-1021)
    - EL2521-0025 1Ch. Pulse Train 24V DC Output negative (EL2521-0025-1021)
    - EL2521-0124 1Ch. Pulse 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. 93: Display of device revision

In many cases several device revisions were created for historic or functional reasons, e.g. through technological advancement. For simplification purposes (see Fig. “Selection dialog for new EtherCAT device”) only the last (i.e. highest) revision and therefore the latest state of production is displayed in the selection dialog for Beckhoff devices. To show all device revisions available in the system as ESI descriptions tick the “Show Hidden Devices” check box, see Fig. “Display of previous revisions”.



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. 94: 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.

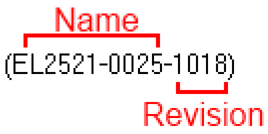


Fig. 95: Name/revision of the terminal

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

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

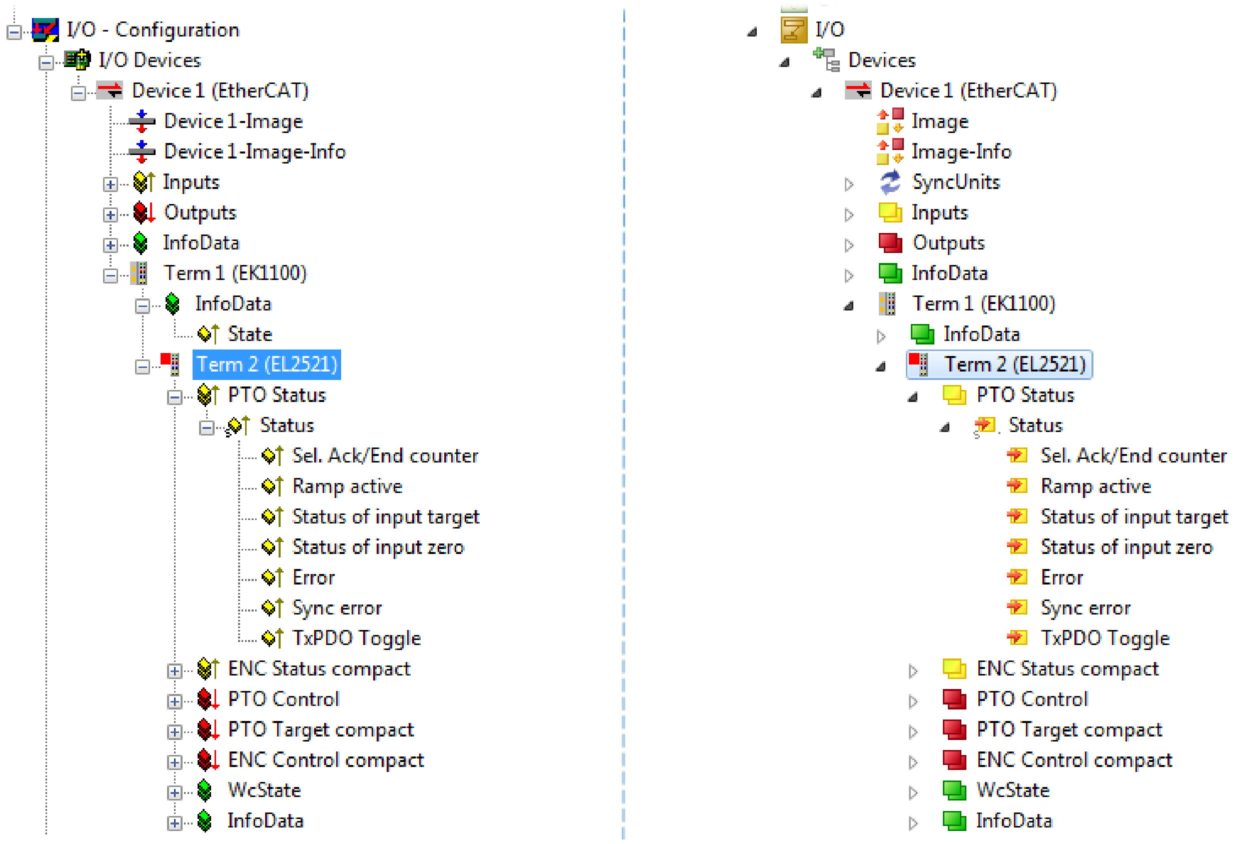


Fig. 96: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

5.2.6 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.

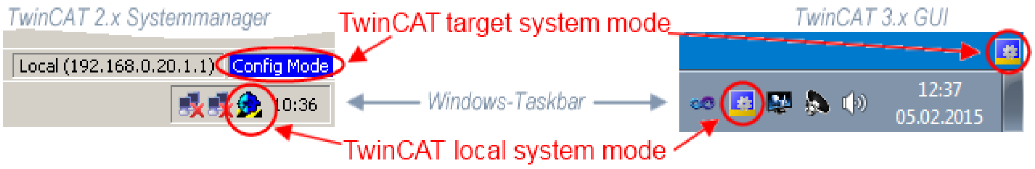


Fig. 97: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

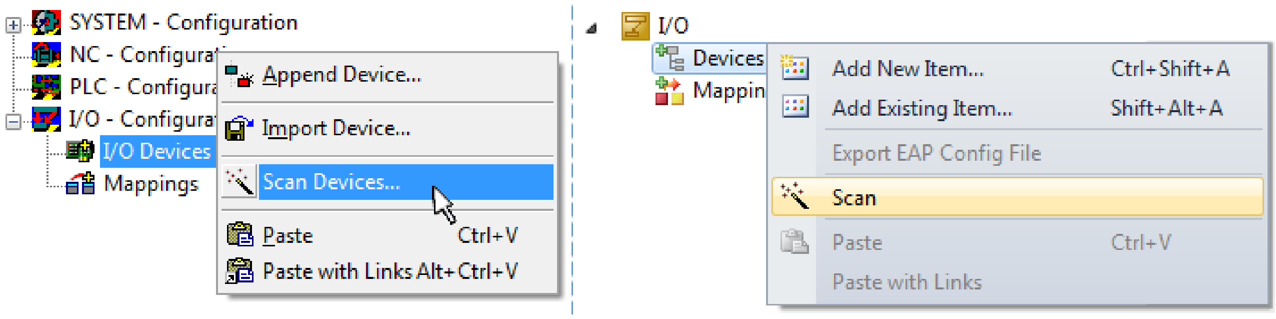
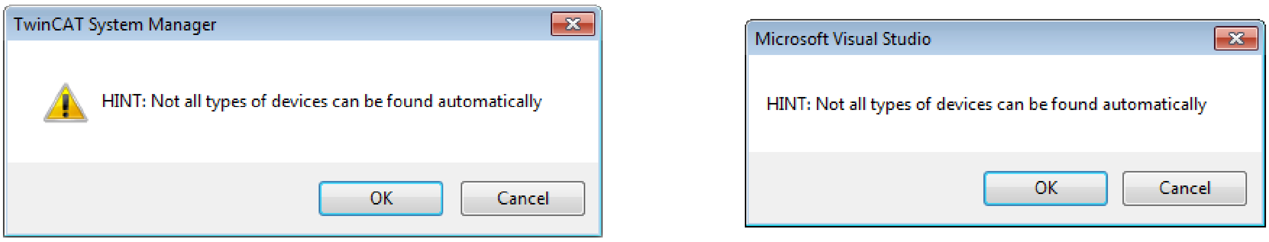


Fig. 98: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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


TwinCAT System Manager

HINT: Not all types of devices can be found automatically

OK Cancel

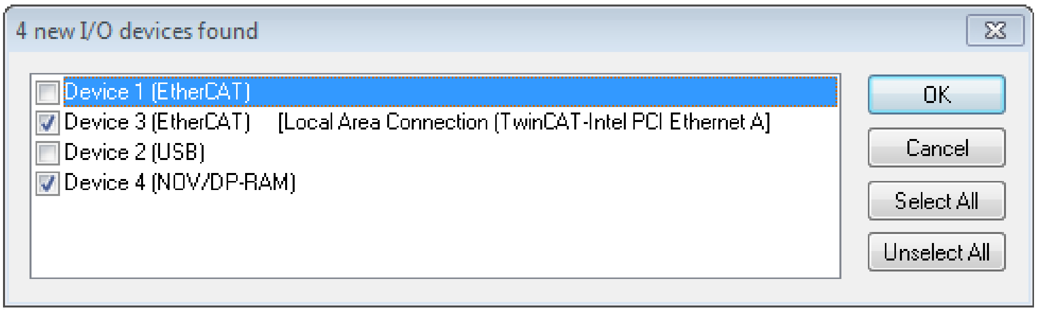
Microsoft Visual Studio

HINT: Not all types of devices can be found automatically

OK Cancel

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

Ethernet ports with installed TwinCAT real-time driver are shown as “RT Ethernet” devices. An EtherCAT frame is sent to these ports for testing purposes. If the scan agent detects from the response that an EtherCAT slave is connected, the port is immediately shown as an “EtherCAT Device” .



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. 100: Detected Ethernet devices

Via respective checkboxes devices can be selected (as illustrated in Fig. “Detected Ethernet devices” e.g. Device 3 and Device 4 were chosen). After confirmation with “OK” a device scan is suggested for all selected devices, see Fig.: “Scan query after automatic creation of an EtherCAT device”.

Selecting the Ethernet port

Ethernet ports can only be selected for EtherCAT devices for which the TwinCAT real-time driver is installed. This has to be done separately for each port. Please refer to the respective installation page [▶ 79].

Detecting/Scanning the EtherCAT devices

Online scan functionality

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

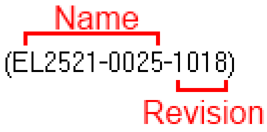


Fig. 101: Example default state

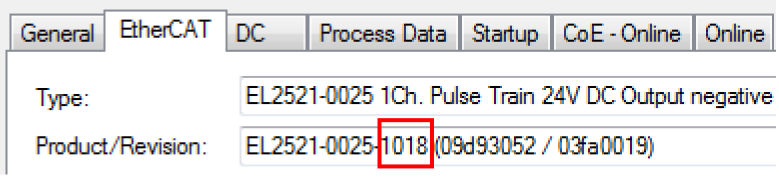
NOTE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▶ 99] 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)

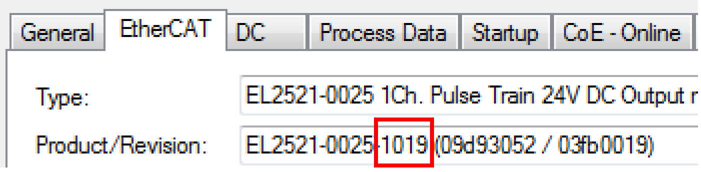
Fig. 102: Installing EthetCAT terminal with revision -1018

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 [▶ 99] 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. 103: Detection of EtherCAT terminal with revision -1019

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

In addition, it could be the case that, due to the development accompanying production in company A, the new feature C of the EL2521-0025-1019 (for example, an improved analog filter or an additional process data for the diagnosis) is discovered and used without in-house consultation. The previous stock of spare part devices are then no longer to be used for the new configuration “B2.tsm” created in this way. 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
Yes
No
Microsoft Visual Studio
Yes
No

Scan for boxes

Scan for boxes

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

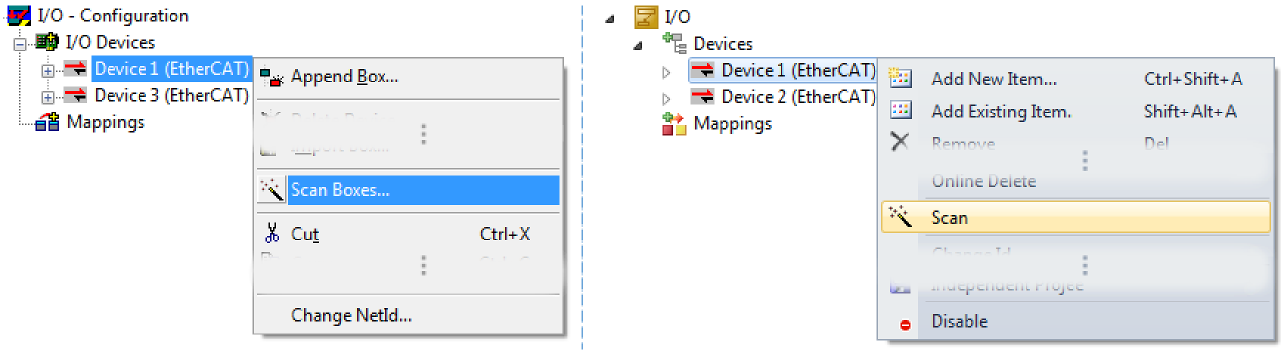


Fig. 105: Manual triggering of a device scan on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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

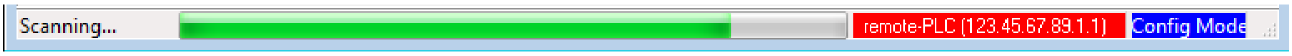


Fig. 106: *Scan progressexemplary by TwinCAT 2*

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

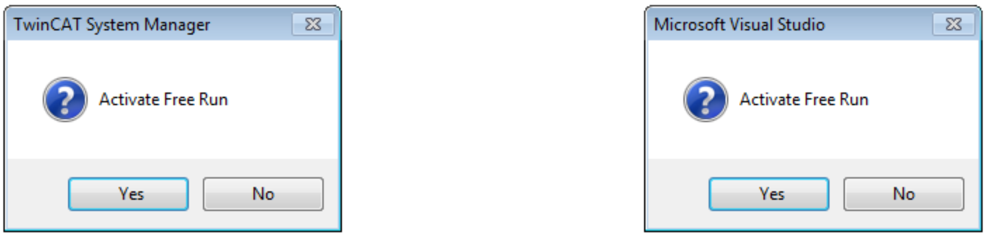


Fig. 107: Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3)

In Config/FreeRun mode the System Manager display alternates between blue and red, and the EtherCAT device continues to operate with the idling cycle time of 4 ms (default setting), even without active task (NC, PLC).

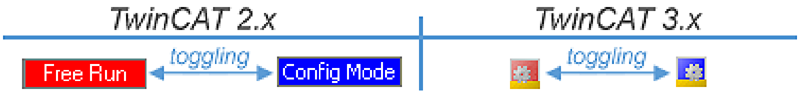


Fig. 108: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

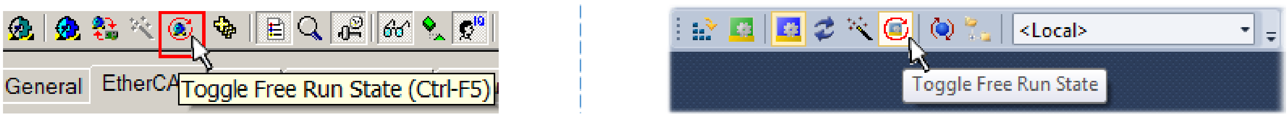
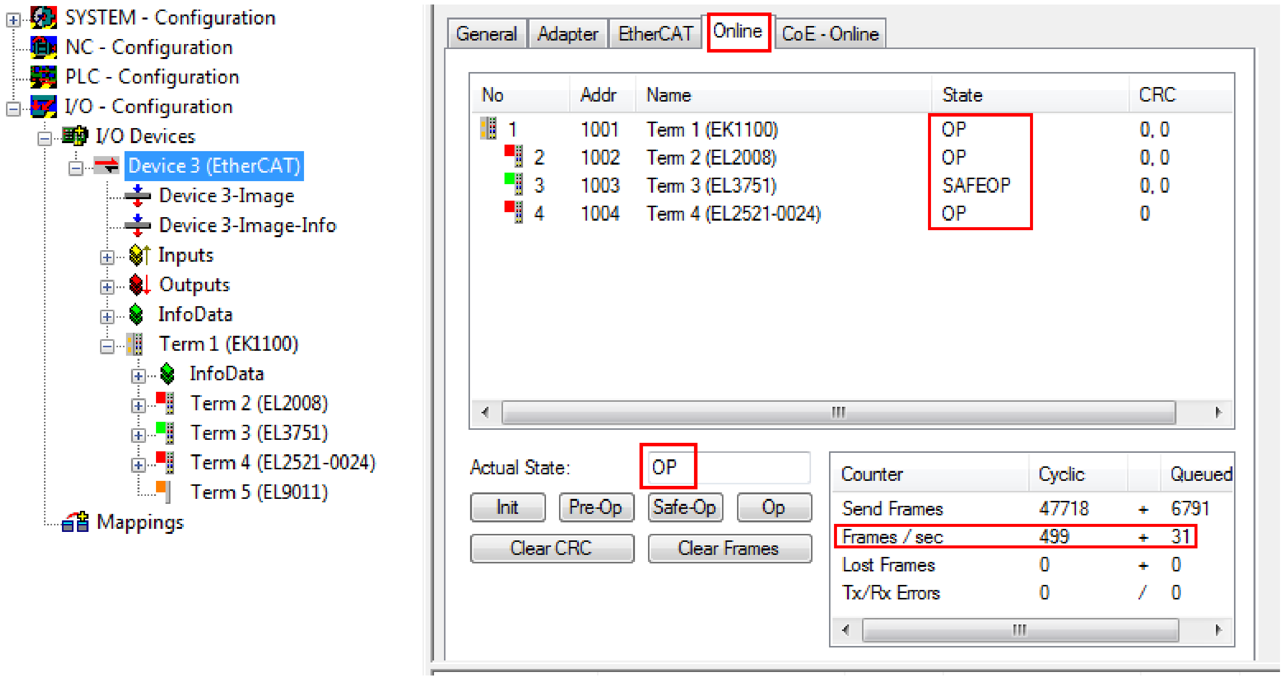


Fig. 109: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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



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

General | Adapter | EtherCAT | Online | CoE - Online

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

Init | Pre-Op | Safe-Op | Op
Clear CRC | Clear Frames

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

Fig. 110: Online display example

Please note:

- all slaves should be in OP state
- the EtherCAT master should be in “Actual State” OP
- “frames/sec” should match the cycle time taking into account the sent number of frames
- no excessive “LostFrames” or CRC errors should occur

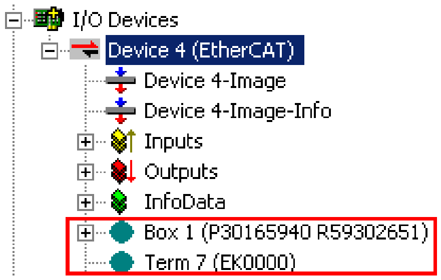
The configuration is now complete. It can be modified as described under manual procedure [▶ 89].

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. 111: Faulty identification

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

Scan over existing Configuration

NOTE

Change of the configuration after comparison

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

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

TwinCAT System Manager

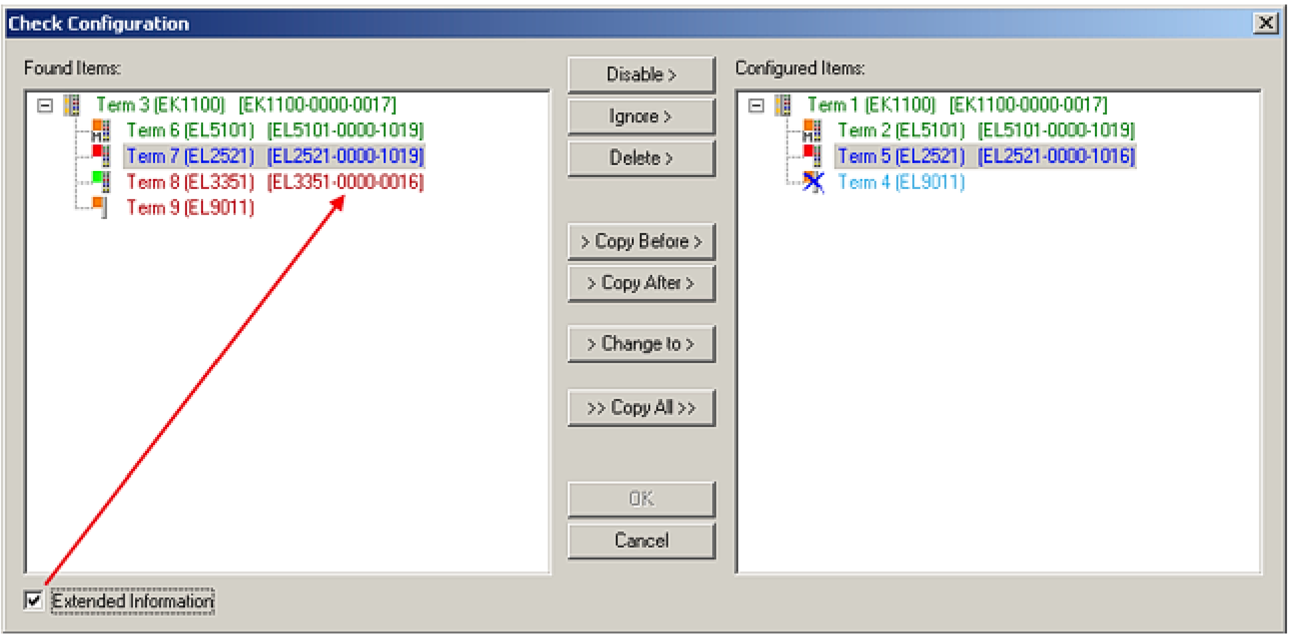
Configuration is identical

Microsoft Visual Studio

Configuration is identical

Fig. 112: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

If differences are detected, they are shown in the correction dialog, so that the user can modify the configuration as required.



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. 113: 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.

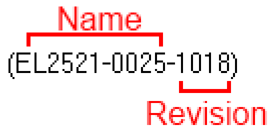
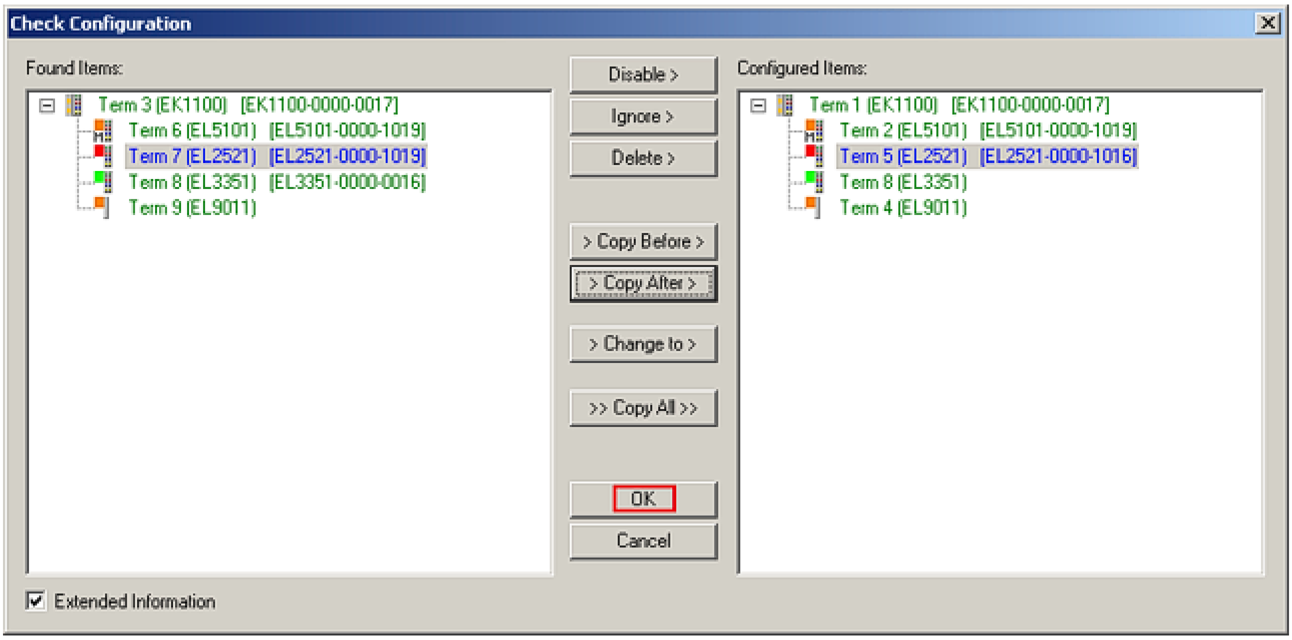


Fig. 114: Name/revision of the terminal

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

In this case the process image of the device is shown in the configuration tree and can be 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. 115: Correction dialog with modifications

Once all modifications have been saved or accepted, click “OK” to transfer them to the real *.tsm configuration.

Change to Compatible Type

TwinCAT offers a function *Change to Compatible Type…* for the exchange of a device whilst retaining the links in the task.

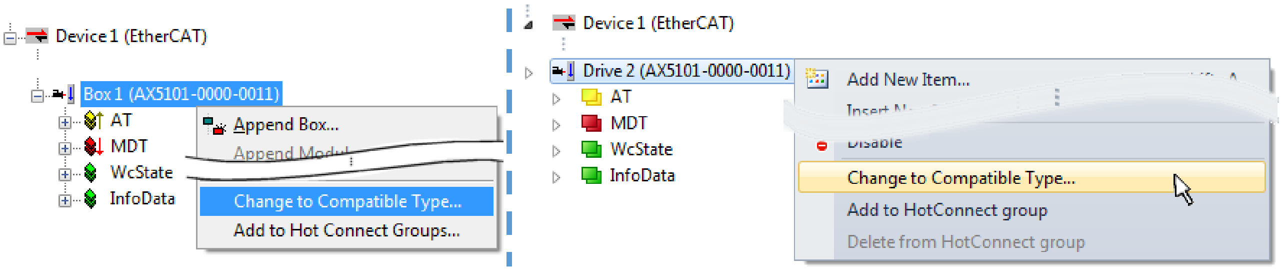


Fig. 116: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

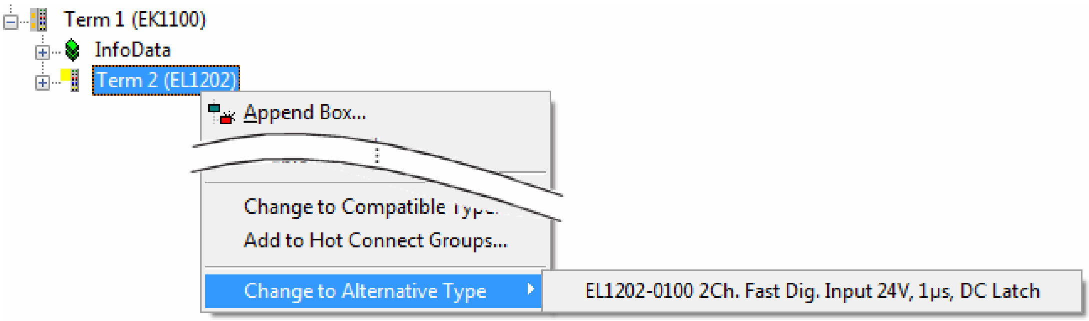

Term 1 (EK1100)
InfoData
Term 2 (EL1202)
Append Box...
Change to Compatible Type
Add to Hot Connect Groups...
Change to Alternative Type
EL1202-0100 2Ch. Fast Dig. Input 24V, 1µs, DC Latch

Fig. 117: 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).

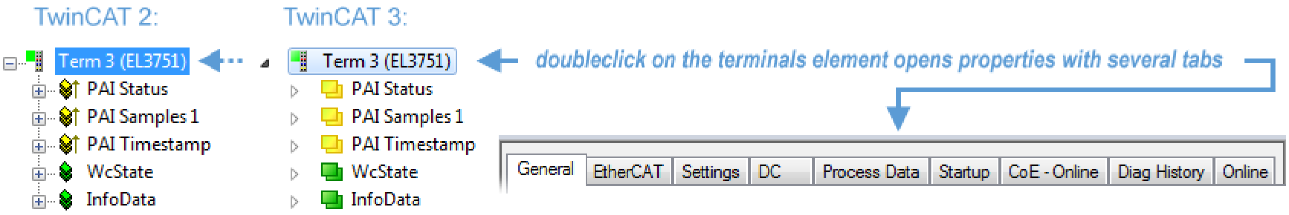
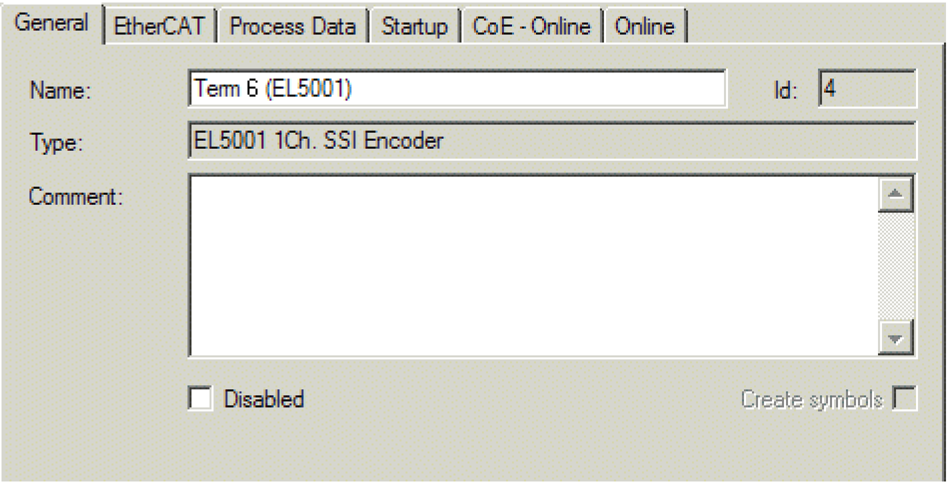


Fig. 118: Branch element as terminal EL3751

In the right-hand window of the TwinCAT System Manager (TwinCAT 2) or the Development Environment (TwinCAT 3), various tabs are now available for configuring the terminal. And yet the dimension of complexity of a subscriber determines which tabs are provided. Thus as illustrated in the example above the terminal EL3751 provides many setup options and also a respective number of tabs are available. On the contrary by the terminal EL1004 for example the tabs “General”, “EtherCAT”, “Process Data” and “Online“ are available only. Several terminals, as for instance the EL6695 provide special functions by a tab with its own terminal name, so “EL6695” in this case. A specific tab “Settings” by terminals with a wide range of setup options will be provided also (e.g. EL3751).

“General” tab



Name: Term 6 (EL5001) Id: 4

Type: EL5001 1Ch. SSI Encoder

Comment:

□ Disabled

Create symbols □

Fig. 119: “General” tab

Name Name of the EtherCAT device

Id Number of the EtherCAT device

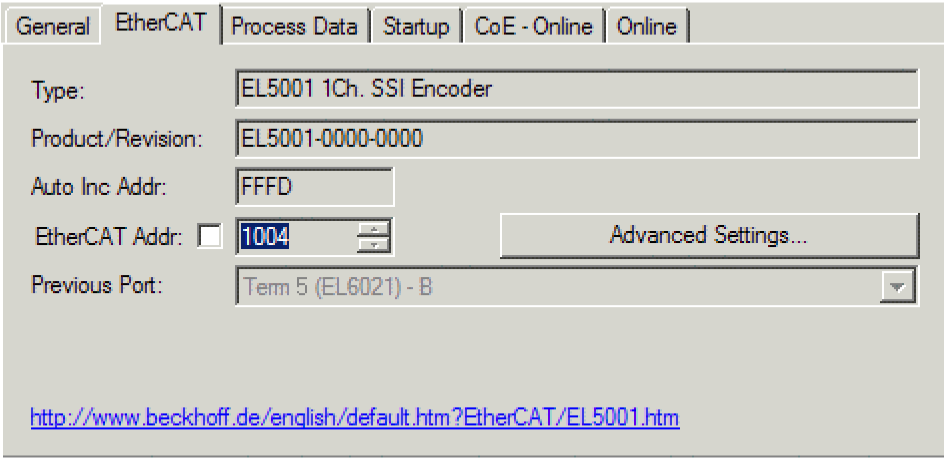
Type EtherCAT device type

Comment Here you can add a comment (e.g. regarding the system).

Disabled Here you can deactivate the EtherCAT device.

Create symbols Access to this EtherCAT slave via ADS is only available if this control box is activated.

“EtherCAT” tab



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. 120: “EtherCAT” tab

Type EtherCAT device type

Product/Revision Product and revision number of the EtherCAT device

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

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

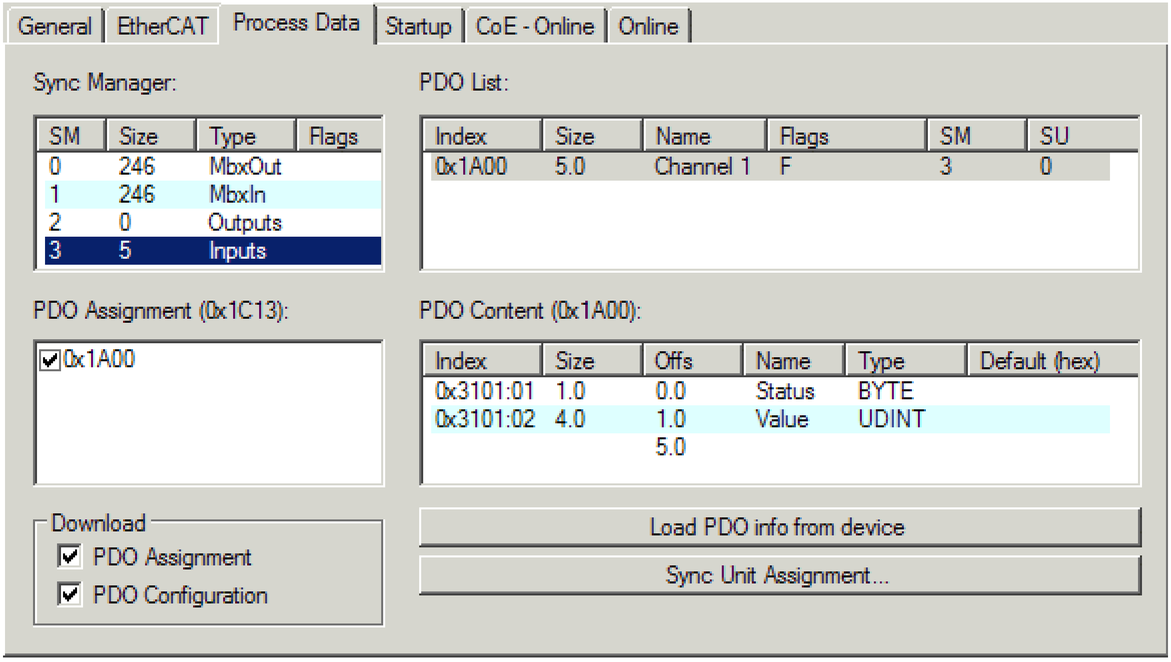
Previous Port Name and port of the EtherCAT device to which this device is connected. If it is possible to connect this device with another one without changing the order of the EtherCAT devices in the communication ring, then this combination field is activated and the EtherCAT device to which this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

Advanced Settings This button opens the dialogs for advanced settings.

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

“Process Data” tab

Indicates the configuration of the process data. The input and output data of the EtherCAT slave are represented as CANopen process data objects (**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. 121: “Process Data” tab

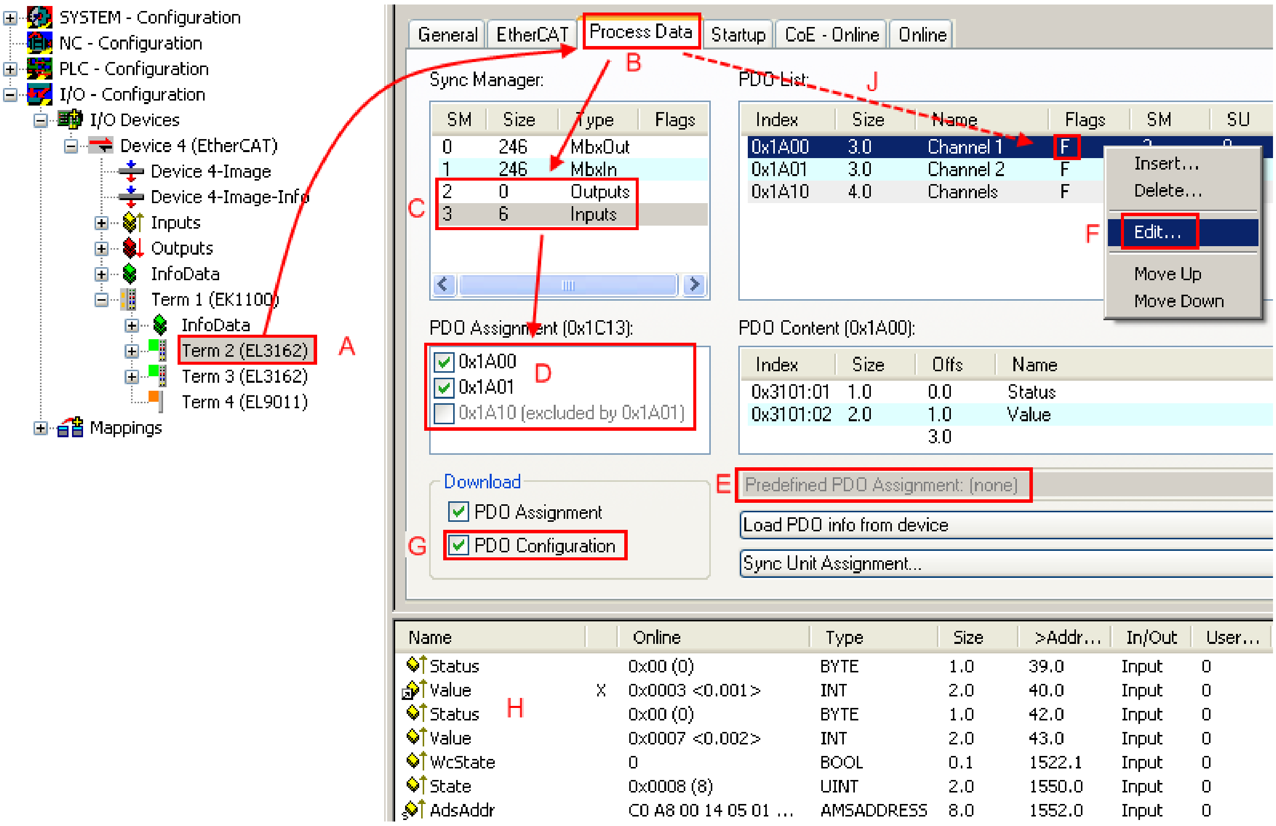
The process data (PDOs) transferred by an EtherCAT slave during each cycle are user data which the application expects to be updated cyclically or which are sent to the slave. To this end the EtherCAT master (Beckhoff TwinCAT) parameterizes each EtherCAT slave during the start-up phase to define which process data (size in bits/bytes, source location, transmission type) it wants to transfer to or from this slave. Incorrect configuration can prevent successful start-up of the slave.

For Beckhoff EtherCAT EL, ES, EM, EJ and EP slaves the following applies in general:

- The input/output process data supported by the device are defined by the manufacturer in the ESI/XML description. The TwinCAT EtherCAT Master uses the ESI description to configure the slave correctly.
- The process data can be modified in the System Manager. See the device documentation.
  Examples of modifications include: mask out a channel, displaying additional cyclic information, 16-bit display instead of 8-bit data size, etc.
- In so-called “intelligent” EtherCAT devices the process data information is also stored in the CoE directory. Any changes in the CoE directory that lead to different PDO settings prevent successful startup of the slave. It is not advisable to deviate from the designated process data, because the device firmware (if available) is adapted to these PDO combinations.

If the device documentation allows modification of process data, proceed as follows (see Figure *Configuring the process data*).

- A: select the device to configure
- B: in the “Process Data” tab select Input or Output under SyncManager (C)
- D: the PDOs can be selected or deselected
- H: the new process data are visible as linkable variables in the System Manager
  The new process data are active once the configuration has been activated and TwinCAT has been restarted (or the EtherCAT master has been restarted)
- E: if a slave supports this, Input and Output PDO can be modified simultaneously by selecting a so-called PDO record (“predefined PDO settings”).



Sync Manager:

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

PDO List:

Index	Size	Name	Flags	SM	SU
0x1A00	3.0	Channel 1	F	3	0
0x1A01	3.0	Channel 2	F		
0x1A10	4.0	Channels	F		

PDO Assignment (0x1C13):

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

PDO Content (0x1A00):

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

Download
☑ PDO Assignment
☑ PDO Configuration

Predefined PDO Assignment: (none)

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. 122: Configuring the process data

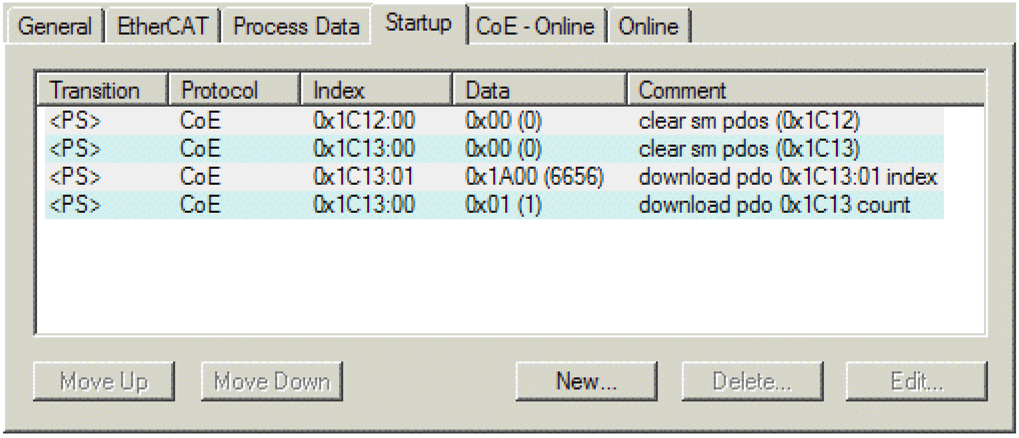
Manual modification of the process data

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 [▶ 110] 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

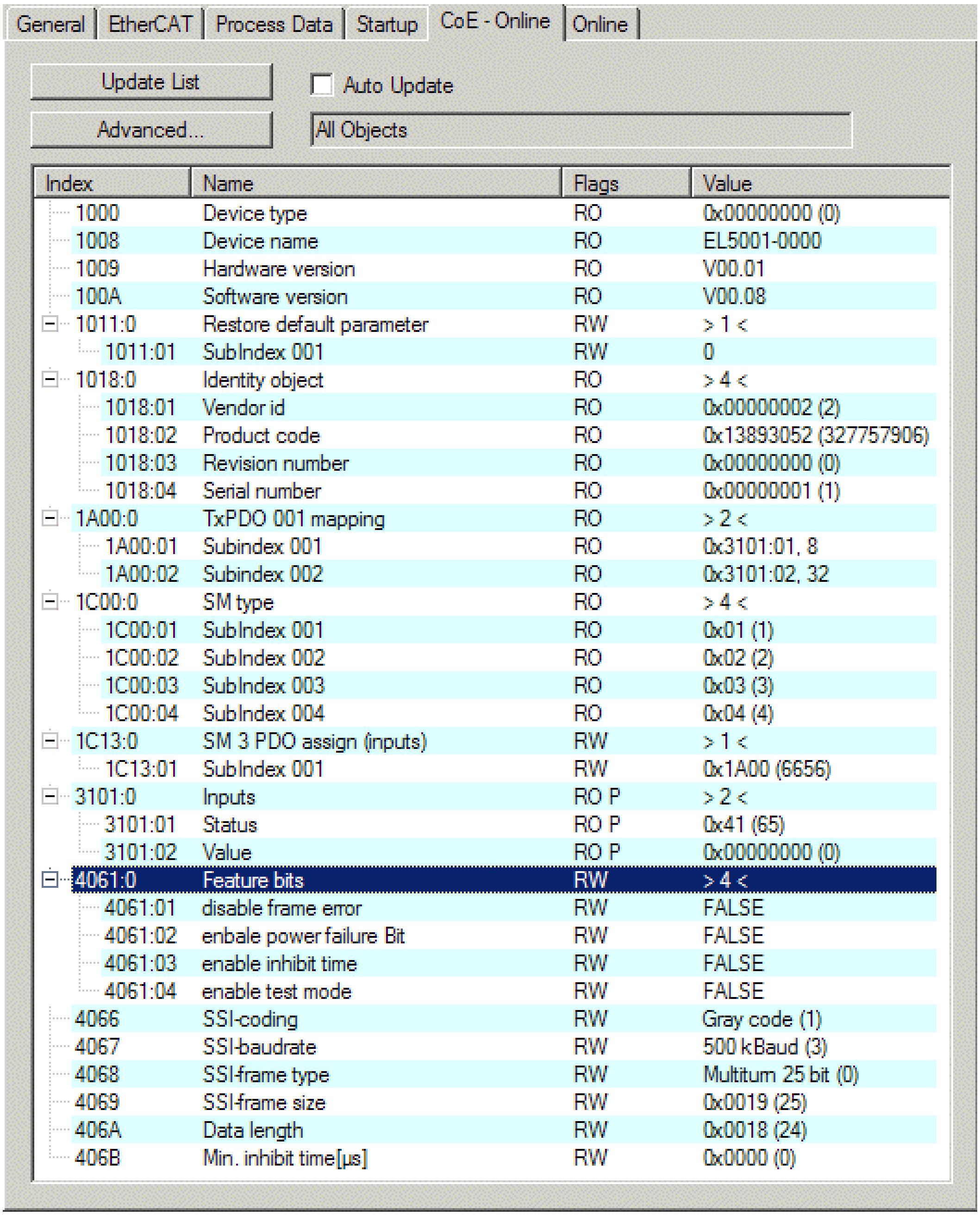
Fig. 123: “Startup” tab

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.



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. 124: “CoE - Online” tab

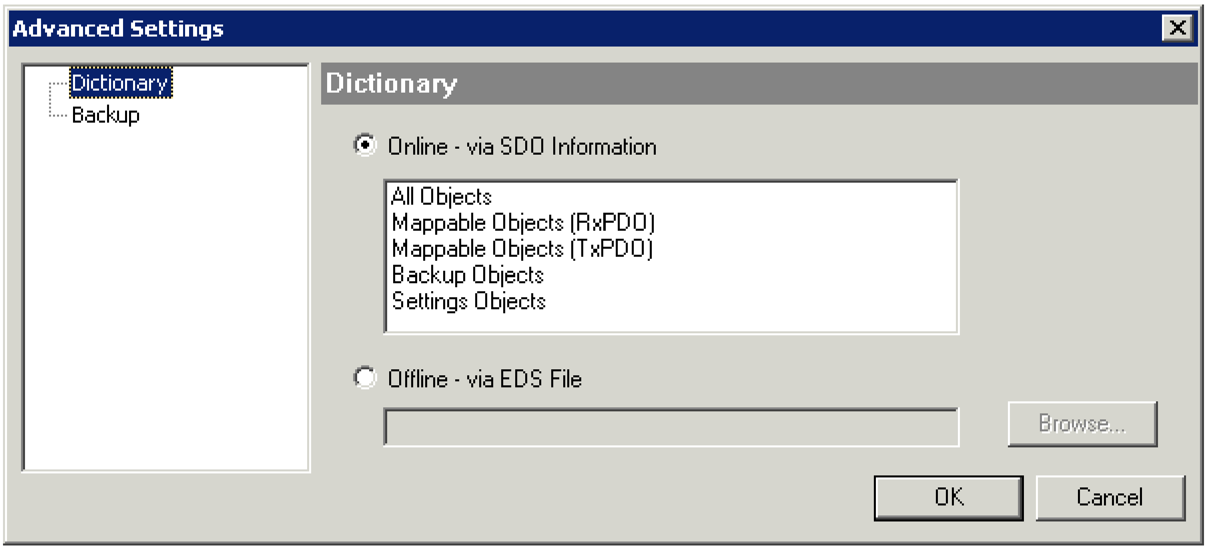
Object list display

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

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

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

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


Advanced Settings

Dictionary
Backup

Dictionary

● Online - via SDO Information

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

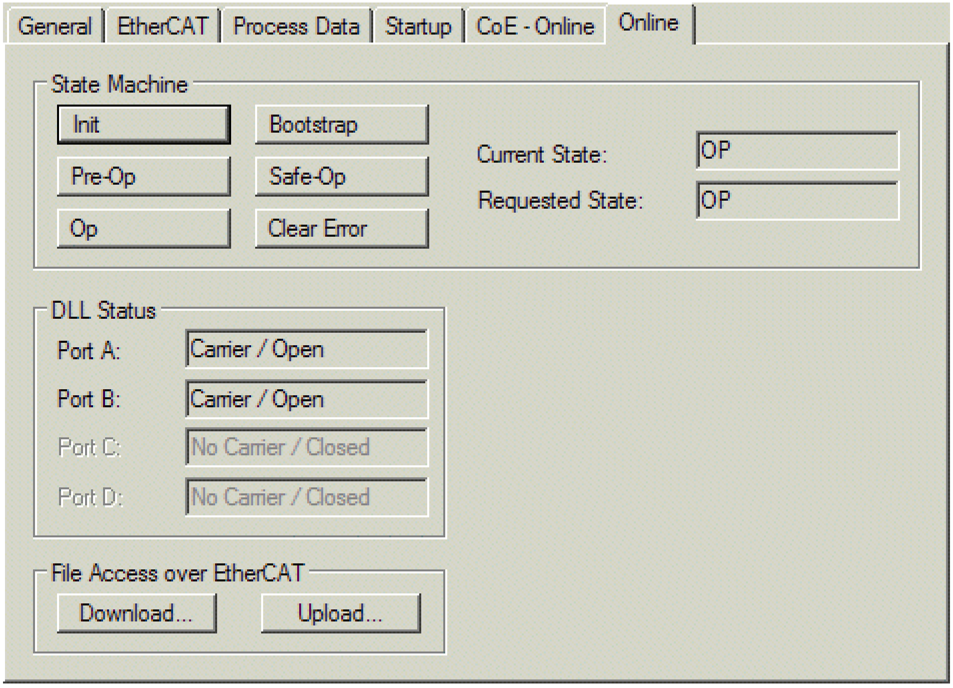
○ Offline - via EDS File

Fig. 125: Dialog “Advanced settings”

Online - via SDO Information If this option button is selected, the list of the objects included in the object list of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.

Offline - via EDS File If this option button is selected, the list of the objects included in the object list is read from an EDS file provided by the user.

“Online” tab



Current State: OP

Requested State: OP

DLL Status

Port A: Carrier / Open

Port B: Carrier / Open

Port C: No Carrier / Closed

Port D: No Carrier / Closed

Fig. 126: “Online” tab

State Machine

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

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

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

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

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

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

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

Current State Indicates the current state of the EtherCAT device.

Requested State Indicates the state requested for the EtherCAT device.

DLL Status

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

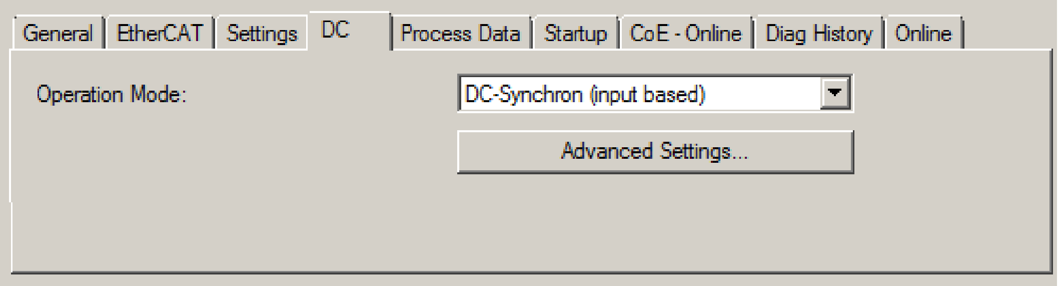
Status	Description
No Carrier / Open	No carrier signal is available at the port, but the port is open.
No Carrier / Closed	No carrier signal is available at the port, and the port is closed.
Carrier / Open	A carrier signal is available at the port, and the port is open.
Carrier / Closed	A carrier signal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

File Access over EtherCAT

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

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

“DC” tab (Distributed Clocks)



Operation Mode: DC-Synchron (input based)

Fig. 127: “DC” tab (Distributed Clocks)

Operation Mode Options (optional):

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

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

Detailed information to Distributed Clocks 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 [▸ 108]),

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 [▸ 105] tab.

PDO Configuration

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

5.2.8 Import/Export of EtherCAT devices with SCI and XTI

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

5.2.8.1 Basic principles

An EtherCAT slave is basically parameterized through the following elements:

- Cyclic process data (PDO)
- Synchronization (Distributed Clocks, FreeRun, SM-Synchron)
- CoE parameters (acyclic object dictionary)

Note: Not all three elements may be present, depending on the slave.

For a better understanding of the export/import function, let's consider the usual procedure for IO configuration:

- The user/programmer processes the IO configuration in the TwinCAT system environment. This involves all input/output devices such as drives that are connected to the fieldbuses used.
  Note: In the following sections, only EtherCAT configurations in the TwinCAT system environment are considered.
- For example, the user manually adds devices to a configuration or performs a scan on the online system.
- This results in the IO system configuration.
- On insertion, the slave appears in the system configuration in the default configuration provided by the vendor, consisting of default PDO, default synchronization method and CoE StartUp parameter as defined in the ESI (XML device description).
- If necessary, elements of the slave configuration can be changed, e.g. the PDO configuration or the synchronization method, based on the respective device documentation.

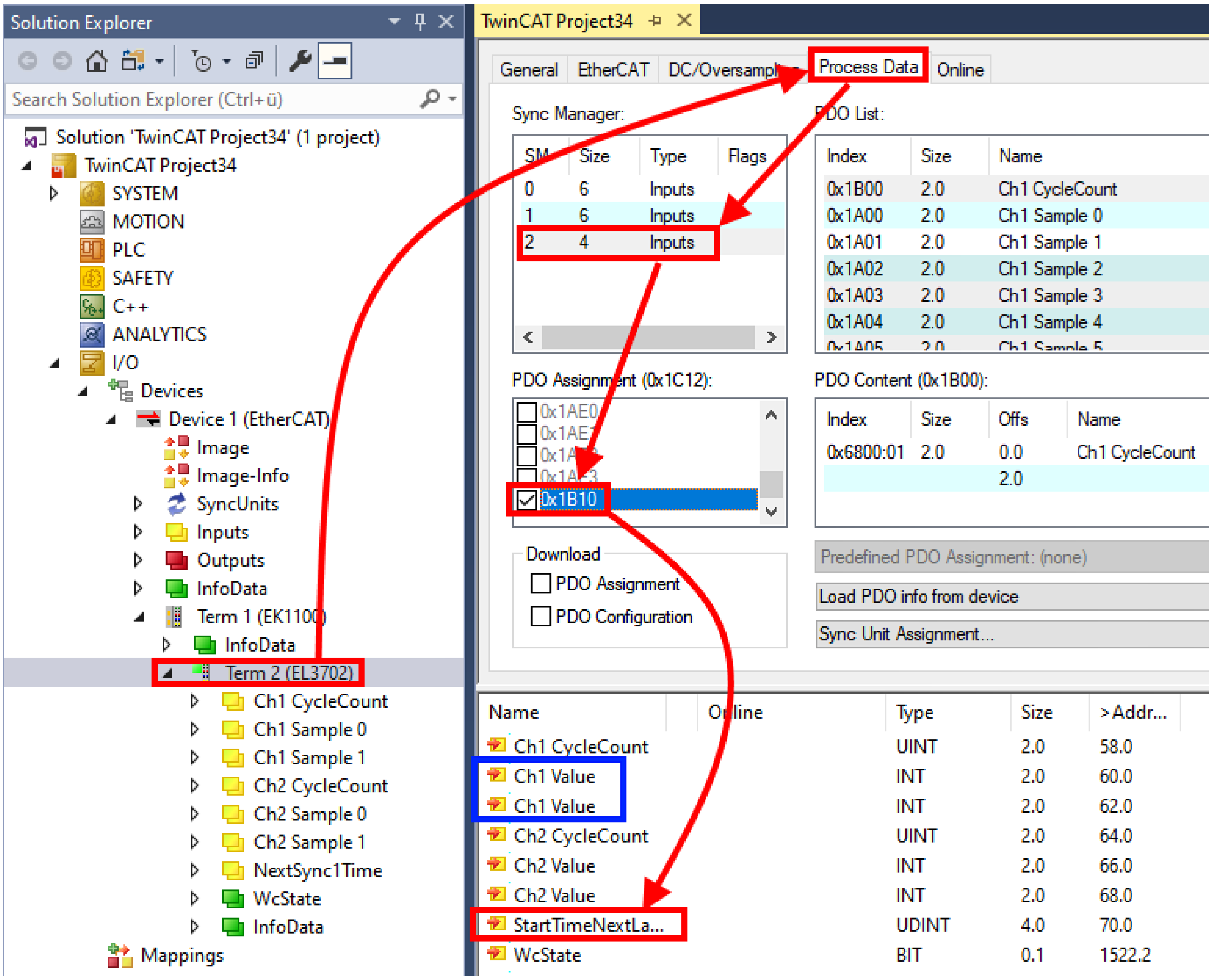
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

- Export the slave configuration from the project,
- Store and transport as a file,
- Import into another EtherCAT project.

TwinCAT offers two methods for this purpose:

- within the TwinCAT environment: Export/Import as **xti** file or
- outside, i.e. beyond the TwinCAT limits: Export/Import as **sci** file.

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



Solution Explorer

Search Solution Explorer (Ctrl+ü)

Solution 'TwinCAT Project34' (1 project)
TwinCAT Project34
SYSTEM
MOTION
PLC
SAFETY
C++
ANALYTICS
I/O
Devices
Device 1 (EtherCAT)
Image
Image-Info
SyncUnits
Inputs
Outputs
InfoData
Term 1 (EK1100)
InfoData
Term 2 (EL3702)
Ch1 CycleCount
Ch1 Sample 0
Ch1 Sample 1
Ch2 CycleCount
Ch2 Sample 0
Ch2 Sample 1
NextSync1Time
WcState
InfoData
Mappings

TwinCAT Project34

General | EtherCAT | DC/Oversampl... | Process Data | Online

Sync Manager:

SM	Size	Type	Flags
0	6	Inputs	
1	6	Inputs	
2	4	Inputs	

PDO List:

Index	Size	Name
0x1B00	2.0	Ch1 CycleCount
0x1A00	2.0	Ch1 Sample 0
0x1A01	2.0	Ch1 Sample 1
0x1A02	2.0	Ch1 Sample 2
0x1A03	2.0	Ch1 Sample 3
0x1A04	2.0	Ch1 Sample 4
0x1A05	2.0	Ch1 Sample 5

PDO Assignment (0x1C12):

□ 0x1AE0
□ 0x1AE
□ 0x1A
□ 0x1A 3
☑ 0x1B10

PDO Content (0x1B00):

Index	Size	Offs	Name
0x6800:01	2.0	0.0	Ch1 CycleCount
		2.0	

Download
□ PDO Assignment
□ PDO Configuration

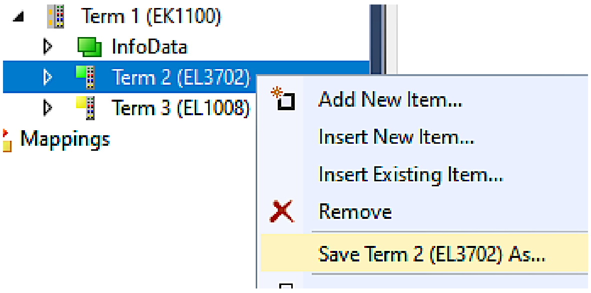
Predefined PDO Assignment: (none)
Load PDO info from device
Sync Unit Assignment...

Name	Online	Type	Size	>Addr...
Ch1 CycleCount		UINT	2.0	58.0
Ch1 Value		INT	2.0	60.0
Ch1 Value		INT	2.0	62.0
Ch2 CycleCount		UINT	2.0	64.0
Ch2 Value		INT	2.0	66.0
Ch2 Value		INT	2.0	68.0
StartTimeNextLa...		UDINT	4.0	70.0
WcState		BIT	0.1	1522.2

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

5.2.8.2 Procedure within TwinCAT with xti files

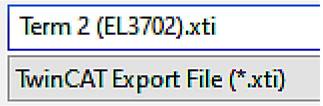
Each IO device can be exported/saved individually:



Term 1 (EK1100)
InfoData
Term 2 (EL3702)
Term 3 (EL1008)
Mappings

Add New Item...
Insert New Item...
Insert Existing Item...
Remove
Save Term 2 (EL3702) As...

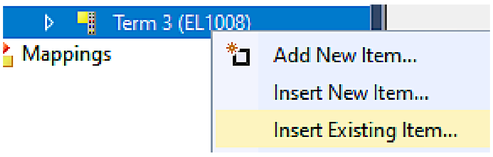
The xti file can be stored:



Term 2 (EL3702).xti

TwinCAT Export File (*.xti)

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



5.2.8.3 Procedure within and outside TwinCAT with sci file

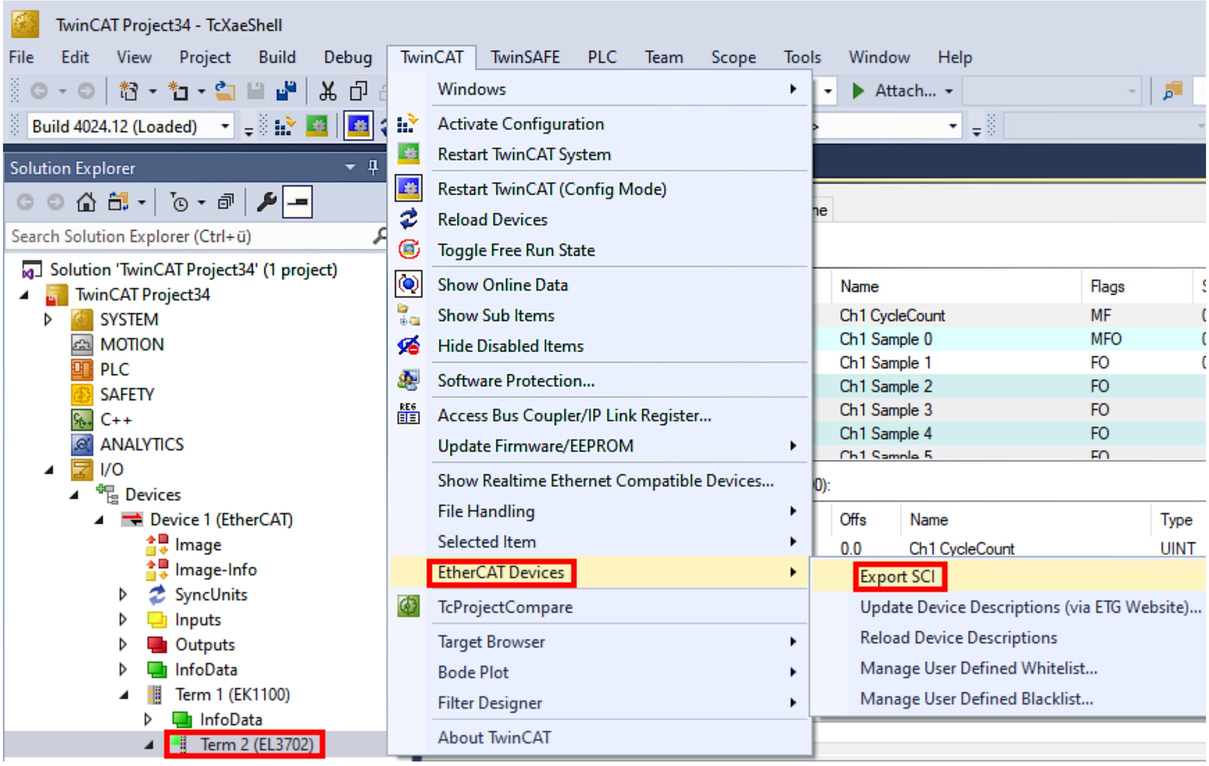
Note regarding availability (2021/01)

The SCI method is available from TwinCAT 3.1 build 4024.14.

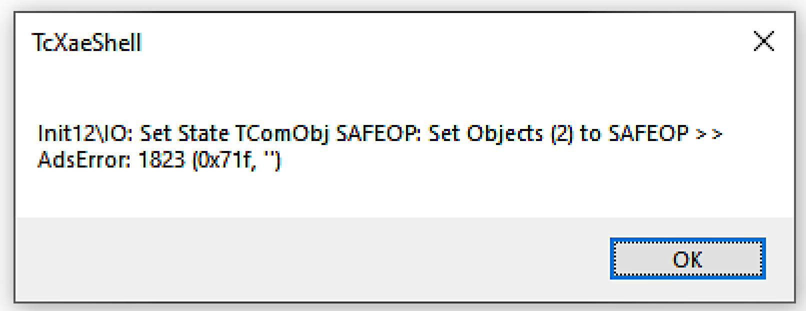
The Slave Configuration Information (SCI) describes a specific complete configuration for an EtherCAT slave (terminal, box, drive...) based on the setting options of the device description file (ESI, EtherCAT Slave Information). That is, it includes PDO, CoE, synchronization.

Export:

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

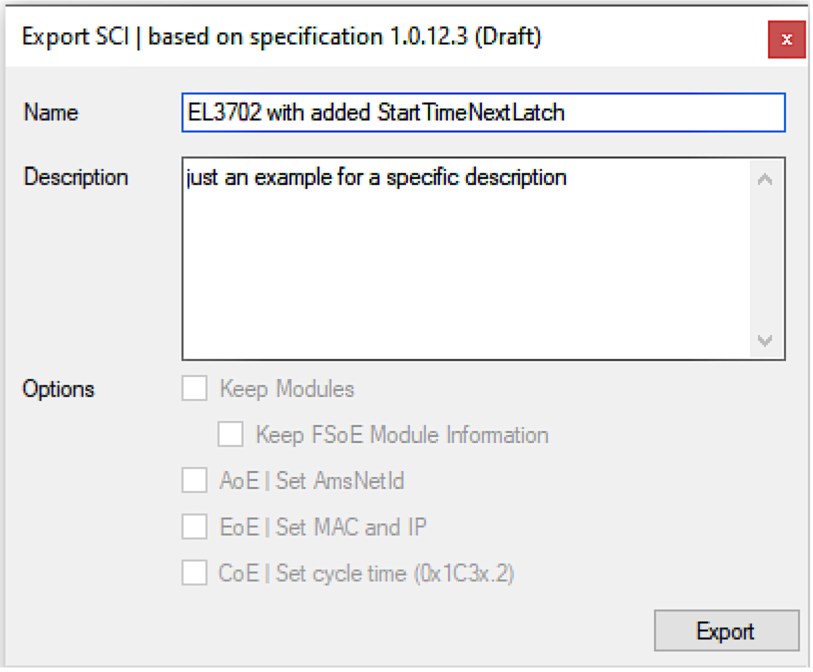


- If TwinCAT is offline (i.e. if there is no connection to an actual running controller) a warning message may appear, because after executing the function the system attempts to reload the EtherCAT segment. However, in this case this is not relevant for the result and can be acknowledged by clicking OK:


TcXaeShell

Init12\IO: Set State TComObj SAFEOP: Set Objects (2) to SAFEOP >> AdsError: 1823 (0x71f, '')

- A description may also be provided:



Export SCI | based on specification 1.0.12.3 (Draft)

Name: EL3702 with added StartTimeNextLatch

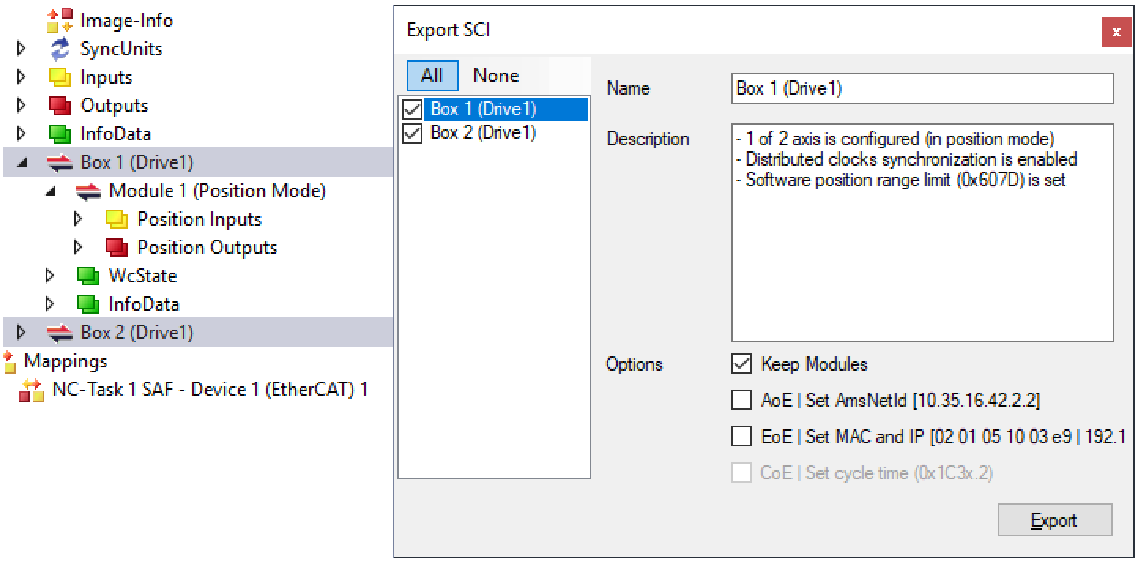
Description: just an example for a specific description

Options:
□ Keep Modules
□ Keep FSoE Module Information
□ AoE | Set AmsNetId
□ EoE | Set MAC and IP
□ CoE | Set cycle time (0x1C3x.2)

- Explanation of the dialog box:

Name		Name of the SCI, assigned by the user.
Description		Description of the slave configuration for the use case, assigned by the user.
Options	Keep modules	If a slave supports modules/slots, the user can decide whether these are to be exported or whether the module and device data are to be combined during export.
	AoE | Set AmsNetId	The configured AmsNetId is exported. Usually this is network-dependent and cannot always be determined in advance.
	EoE | Set MAC and IP	The configured virtual MAC and IP addresses are stored in the SCI. Usually these are network-dependent and cannot always be determined in advance.
	CoE | Set cycle time(0x1C3x.2)	The configured cycle time is exported. Usually this is network-dependent and cannot always be determined in advance.
ESI		Reference to the original ESI file.
Export		Save SCI file.

- A list view is available for multiple selections (*Export multiple SCI files*):



☑ Box 1 (Drive1)
☑ Box 2 (Drive1)

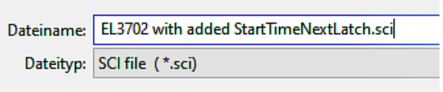
Name: Box 1 (Drive1)

Description:
- 1 of 2 axis is configured (in position mode)
- Distributed clocks synchronization is enabled
- Software position range limit (0x607D) is set

Options:
☑ Keep Modules
□ AoE | Set AmsNetId [10.35.16.42.2.2]
□ EoE | Set MAC and IP [02 01 05 10 03 e9 | 192.1
□ CoE | Set cycle time (0x1C3x.2)

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

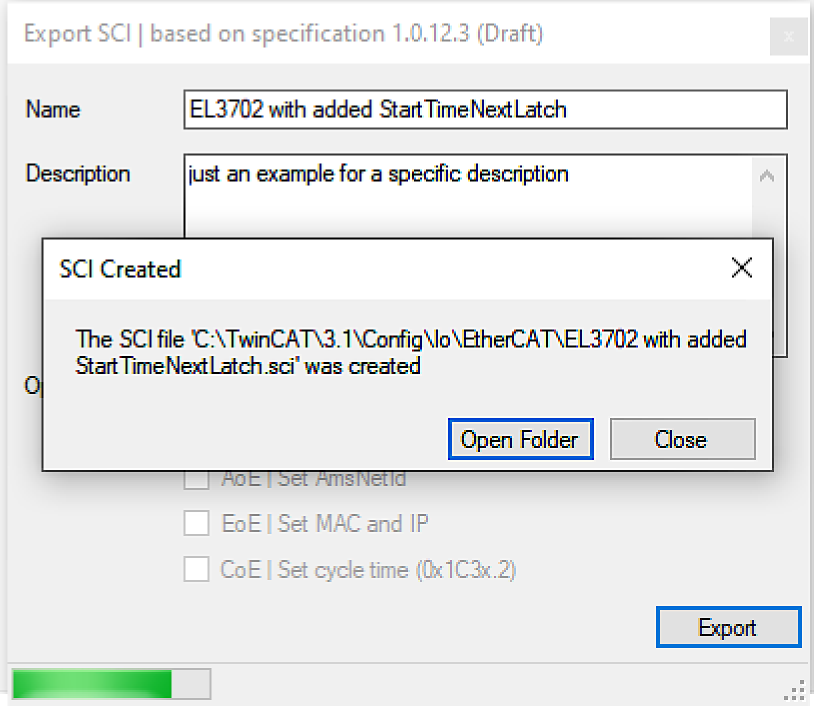
  - None:
    All slaves are deselected.
- The sci file can be saved locally:



Dateiname: EL3702 with added StartTimeNextLatch.sci

Dateityp: SCI file (*.sci)

- The export takes place:



Export SCI | based on specification 1.0.12.3 (Draft)

Name: EL3702 with added StartTimeNextLatch

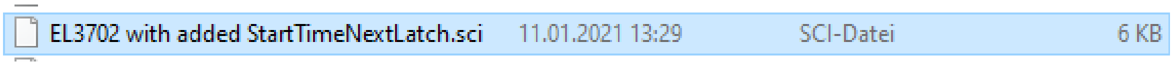
Description: just an example for a specific description

SCI Created

The SCI file 'C:\TwinCAT\3.1\Config\Io\EtherCAT\EL3702 with added StartTimeNextLatch.sci' was created

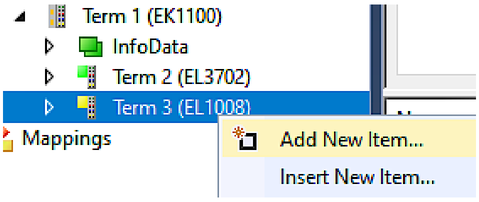
Import

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

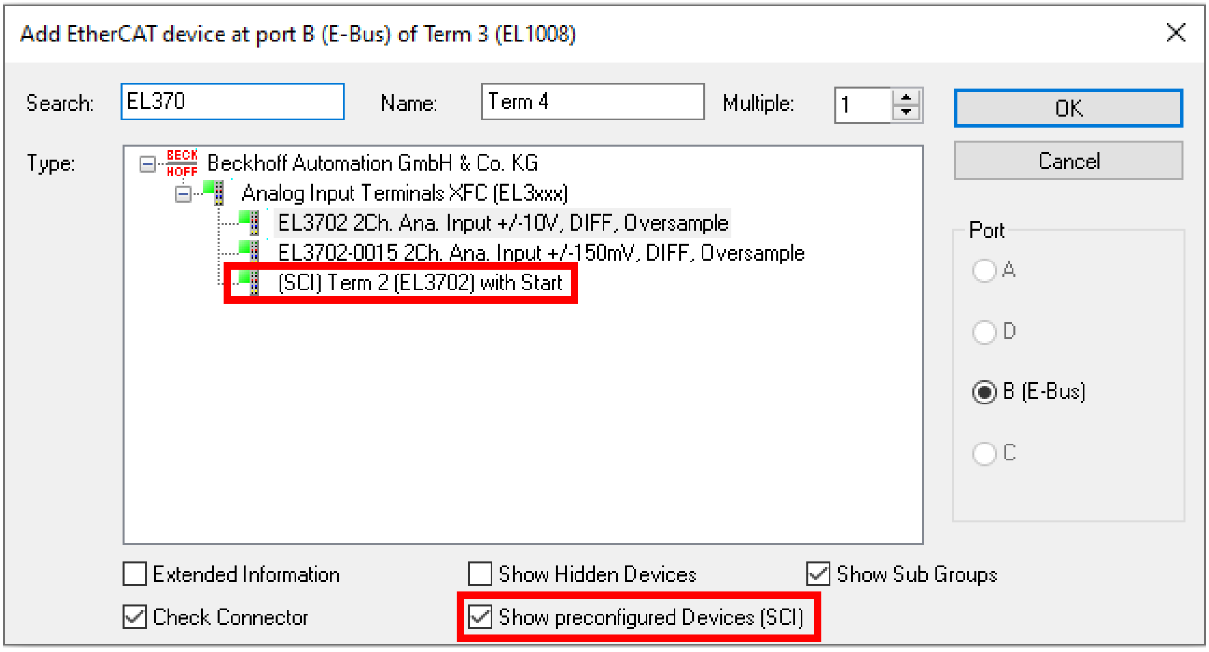


EL3702 with added StartTimeNextLatch.sci	11.01.2021 13:29	SCI-Datei	6 KB

- Open the selection dialog:



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



Add EtherCAT device at port B (E-Bus) of Term 3 (EL1008)

Search: EL370

Name: Term 4

Multiple: 1

Type:
- Beckhoff Automation GmbH & Co. KG
  - Analog Input Terminals XFC (EL3xxx)
    - EL3702 2Ch. Ana. Input +/-10V, DIFF, Oversample
    - EL3702-0015 2Ch. Ana. Input +/-150mV, DIFF, Oversample
    - (SCI) Term 2 (EL3702) with Start

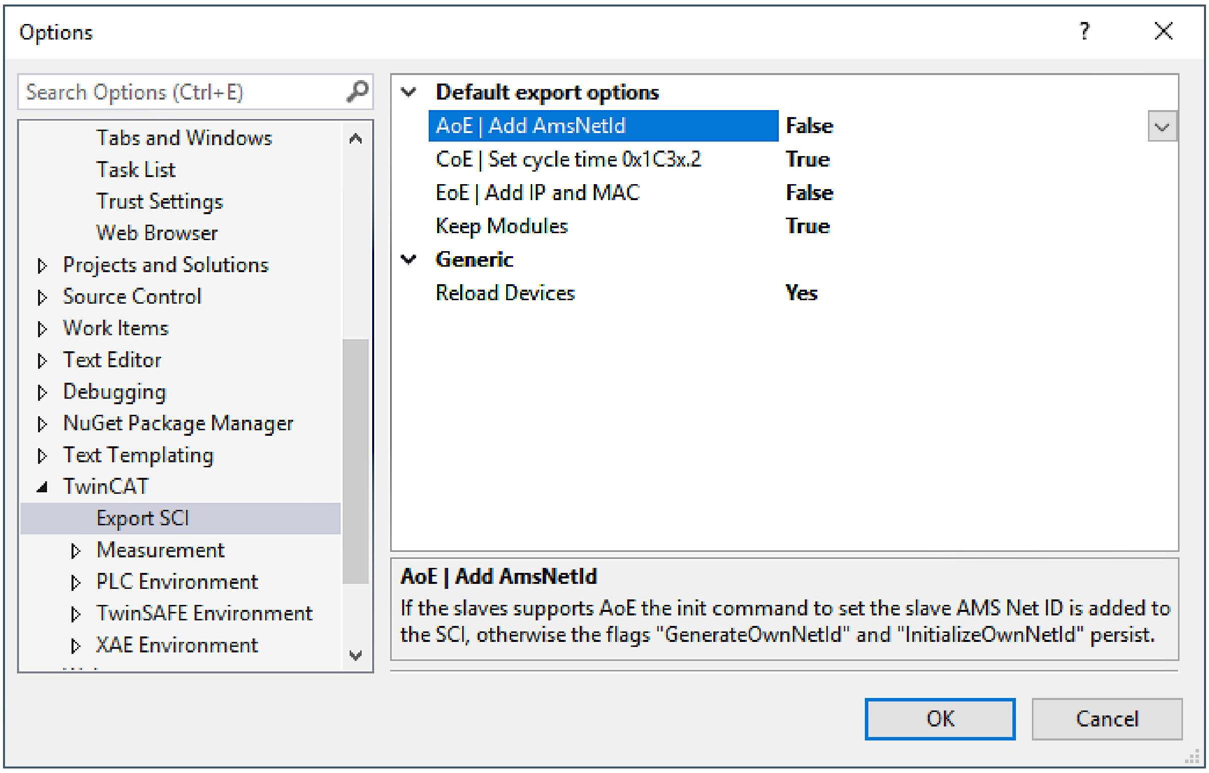
Port: ○ A ○ D ● B (E-Bus) ○ C

□ Extended Information □ Show Hidden Devices ☑ Show Sub Groups

☑ Check Connector ☑ Show preconfigured Devices (SCI)

Additional Notes

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



Options

Default export options

AoE | Add AmsNetId	False
CoE | Set cycle time 0x1C3x.2	True
EoE | Add IP and MAC	False
Keep Modules	True

Generic

Reload Devices	Yes

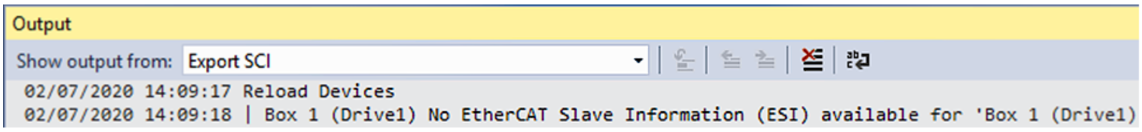
AoE | Add AmsNetId

If the slaves supports AoE the init command to set the slave AMS Net ID is added to the SCI, otherwise the flags "GenerateOwnNetId" and "InitializeOwnNetId" persist.

Explanation of the settings:

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

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



Export SCI

```
02/07/2020 14:09:17 Reload Devices
02/07/2020 14:09:18 | Box 1 (Drive1) No EtherCAT Slave Information (ESI) available for 'Box 1 (Drive1)
```

5.3 General Notes - EtherCAT Slave Application

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

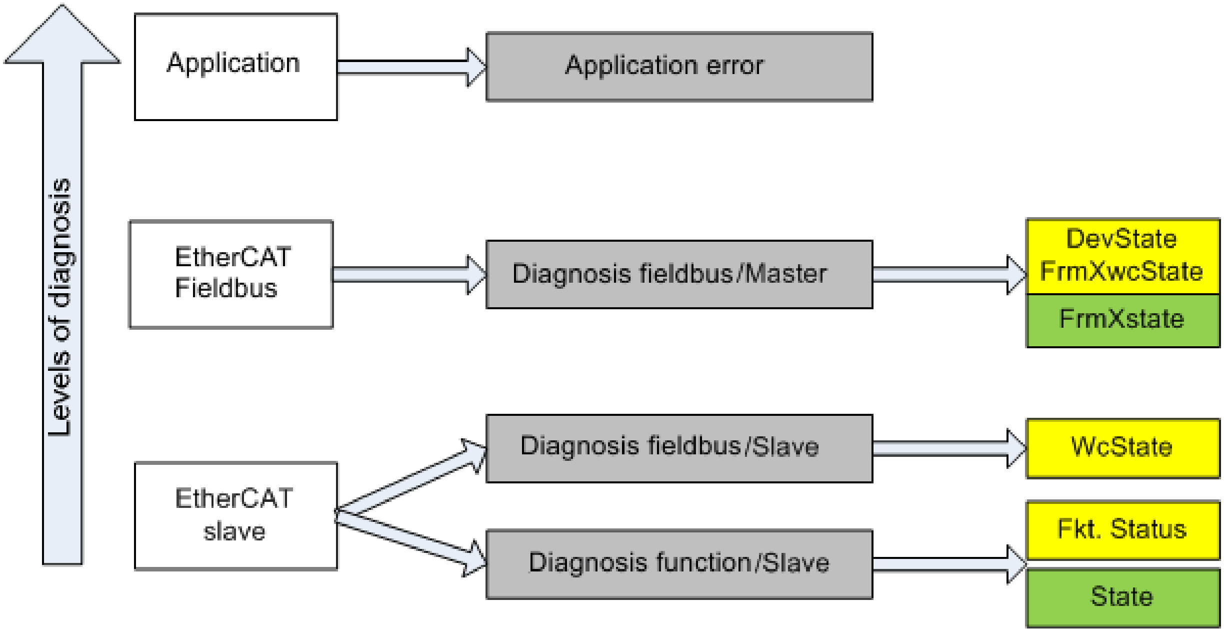


Fig. 128: Selection of the diagnostic information of an EtherCAT Slave

In general, an EtherCAT Slave offers

- communication diagnosis typical for a slave (diagnosis of successful participation in the exchange of process data, and correct operating mode)
  This diagnosis is the same for all slaves.

  as well as
- function diagnosis typical for a channel (device-dependent)
  See the corresponding device documentation

The colors in Fig. *Selection of the diagnostic information of an EtherCAT Slave* also correspond to the variable colors in the System Manager, see Fig. *Basic EtherCAT Slave Diagnosis in the PLC*.

Colour	Meaning
yellow	Input variables from the Slave to the EtherCAT Master, updated in every cycle
red	Output variables from the Slave to the EtherCAT Master, updated in every cycle
green	Information variables for the EtherCAT Master that are updated acyclically. This means that it is possible that in any particular cycle they do not represent the latest possible status. It is therefore useful to read such variables through ADS.

Fig. *Basic EtherCAT Slave Diagnosis in the PLC* shows an example of an implementation of basic EtherCAT Slave Diagnosis. A Beckhoff EL3102 (2-channel analogue input terminal) is used here, as it offers both the communication diagnosis typical of a slave and the functional diagnosis that is specific to a channel. Structures are created as input variables in the PLC, each corresponding to the process image.

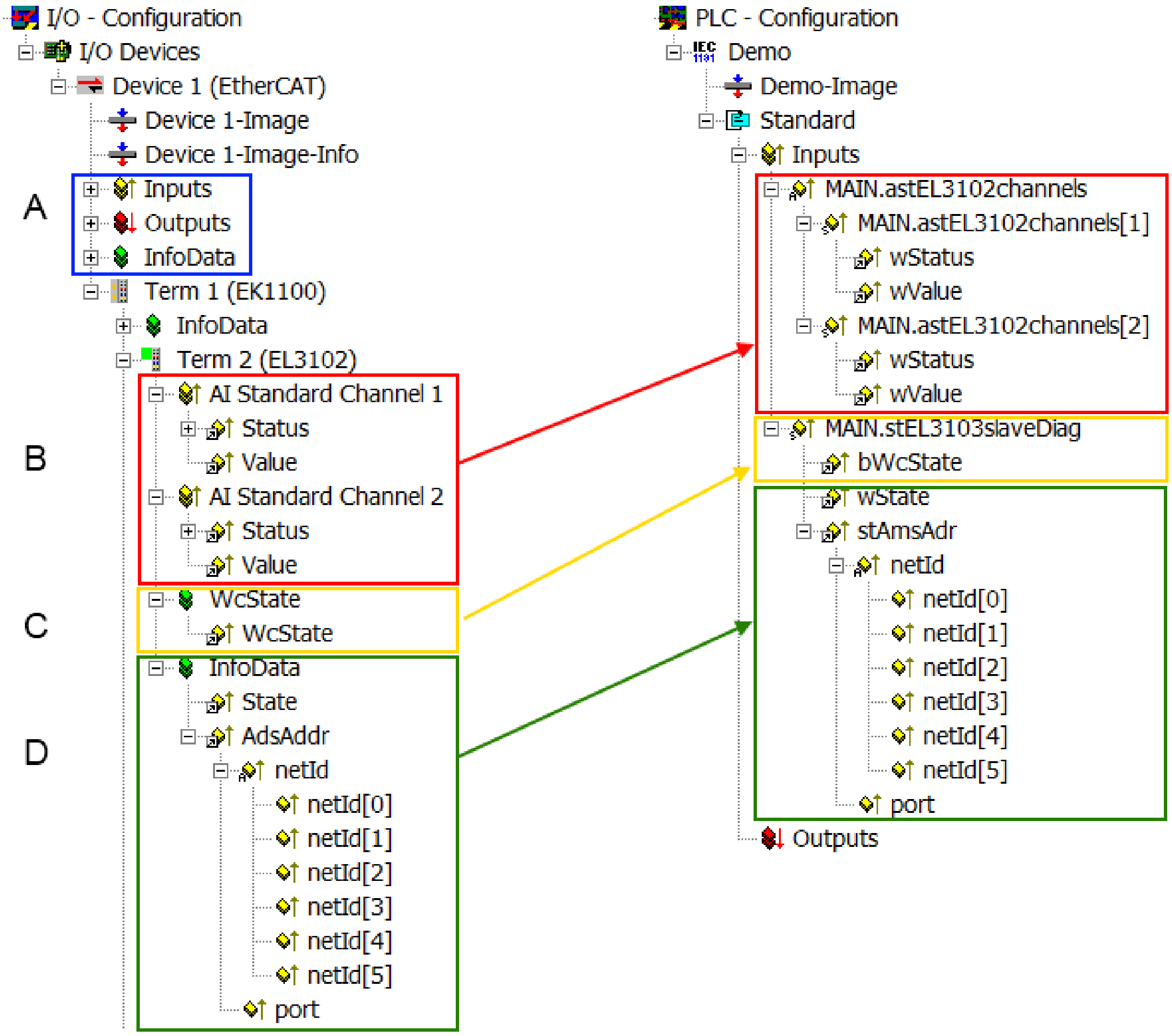


Fig. 129: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated acyclically (yellow) or provided acyclically (green).		At least the DevState is to be evaluated for the most recent cycle in the PLC. The EtherCAT Master's diagnostic information offers many more possibilities than are treated in the EtherCAT System Documentation. A few keywords: • 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.

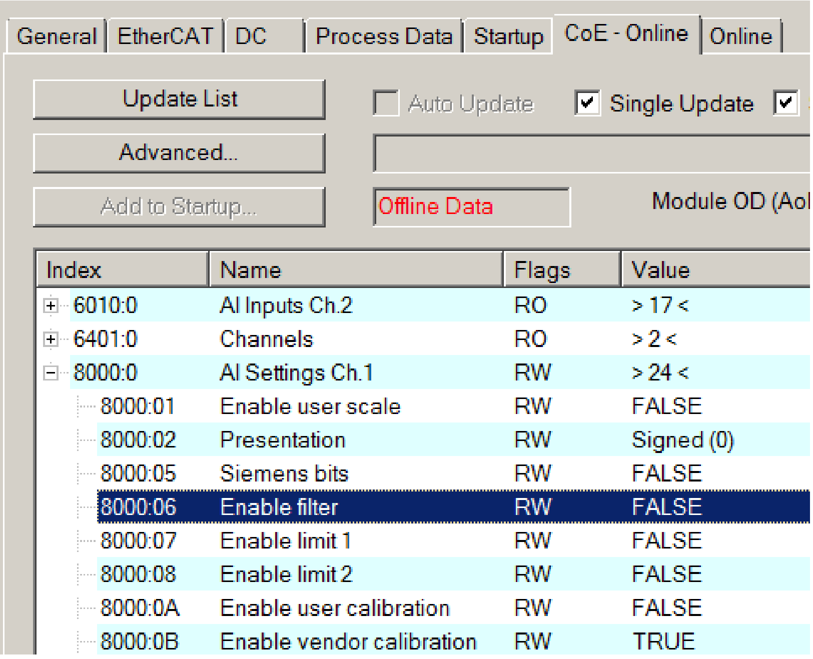
NOTE

Diagnostic information

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

CoE Parameter Directory

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



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

Fig. 130: EL3102, CoE directory

EtherCAT System Documentation

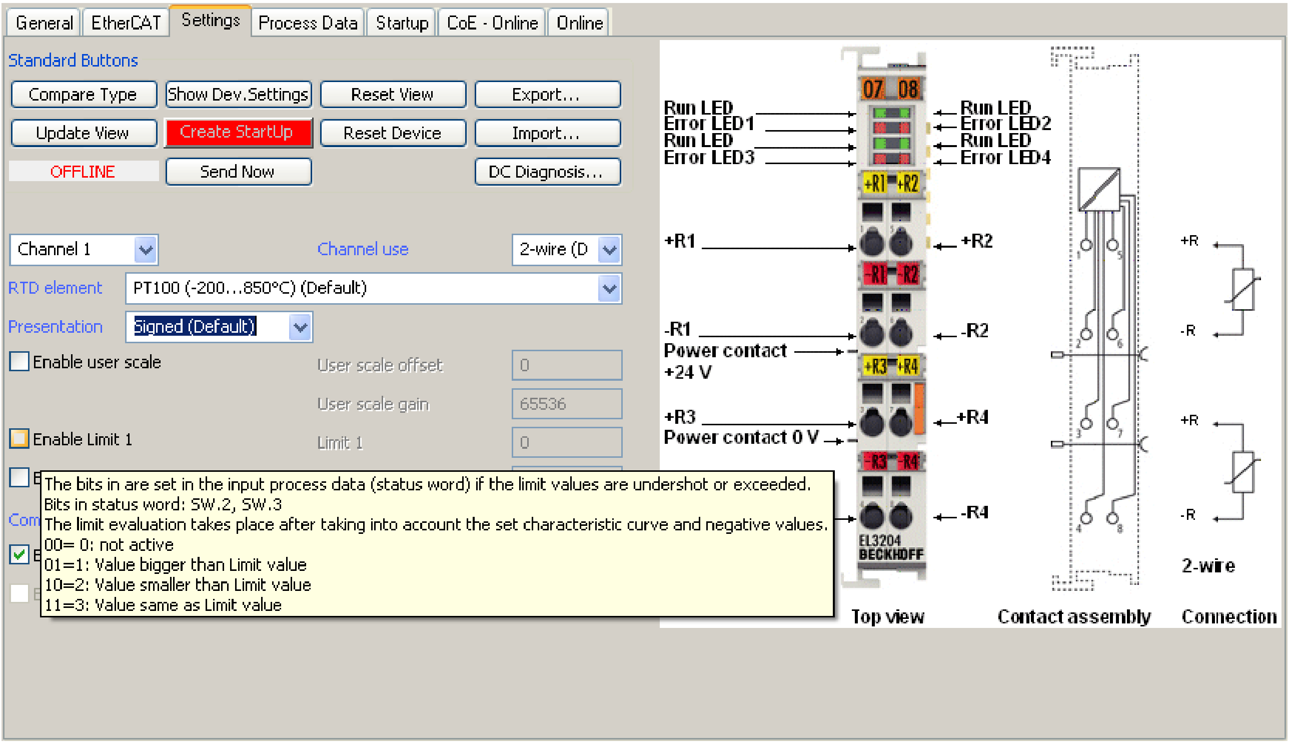
The comprehensive description in the EtherCAT System Documentation (EtherCAT Basics --> CoE Interface) must be observed!

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. 131: Example of commissioning aid for a EL3204

This commissioning process simultaneously manages

- CoE Parameter Directory
- DC/FreeRun mode
- the available process data records (PDO)

Although the “Process Data”, “DC”, “Startup” and “CoE-Online” that used to be necessary for this are still displayed, it is recommended that, if the commissioning aid is used, the automatically generated settings are not changed by it.

The commissioning tool does not cover every possible application of an EL/EP device. If the available setting options are not adequate, the user can make the DC, PDO and CoE settings manually, as in the past.

EtherCAT State: automatic default behaviour of the TwinCAT System Manager and manual operation

After the operating power is switched on, an EtherCAT Slave must go through the following statuses

- INIT
- PREOP
- SAFEOP
- OP

to ensure sound operation. The EtherCAT Master directs these statuses in accordance with the initialization routines that are defined for commissioning the device by the ES/XML and user settings (Distributed Clocks (DC), PDO, CoE). See also the section on "Principles of Communication, EtherCAT State Machine [▶ 29]" in this connection. Depending how much configuration has to be done, and on the overall communication, booting can take up to a few seconds.

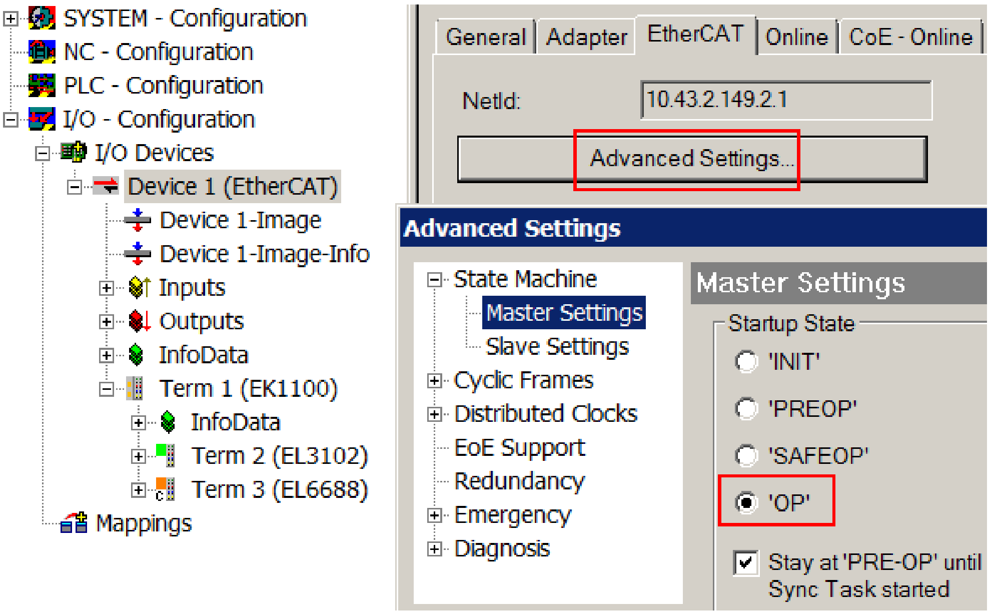
The EtherCAT Master itself must go through these routines when starting, until it has reached at least the OP target state.

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.

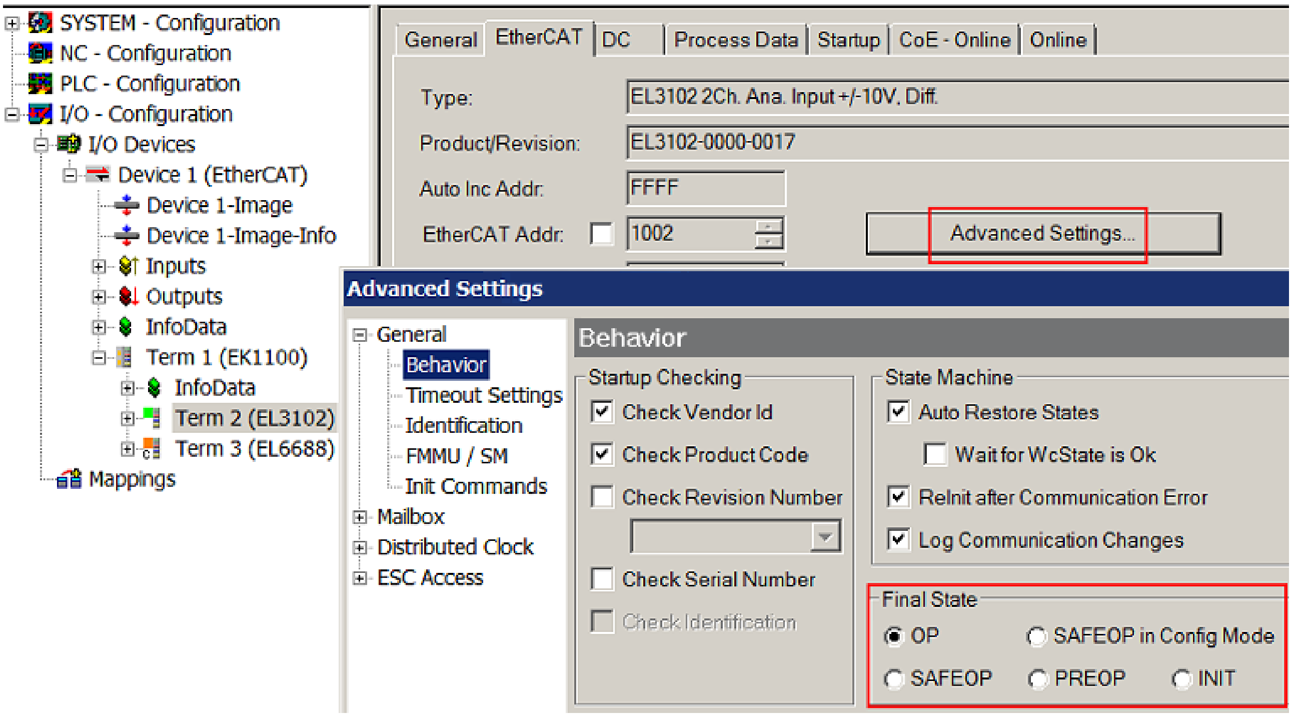


NetId: 10.43.2.149.2.1

Startup State
○ 'INIT'
○ 'PREOP'
○ 'SAFEOP'
● 'OP'
☑ Stay at 'PRE-OP' until Sync Task started

Fig. 132: Default behaviour of the System Manager

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



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

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. 133: Default target state in the Slave

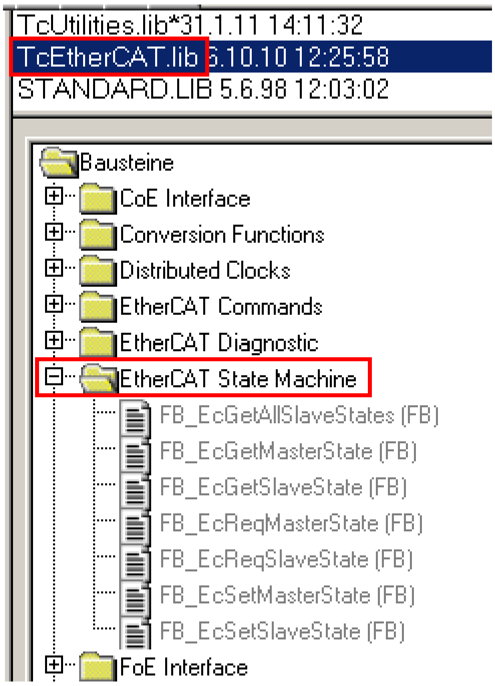
Manual Control

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

- for diagnostic reasons
- to induce a controlled restart of axes
- because a change in the times involved in starting is desirable

In that case it is appropriate in the PLC application to use the PLC function blocks from the *TcEtherCAT.lib*, which is available as standard, and to work through the states in a controlled manner using, for instance, *FB_EcSetMasterState*.

It is then useful to put the settings in the EtherCAT Master to INIT for master and slave.



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

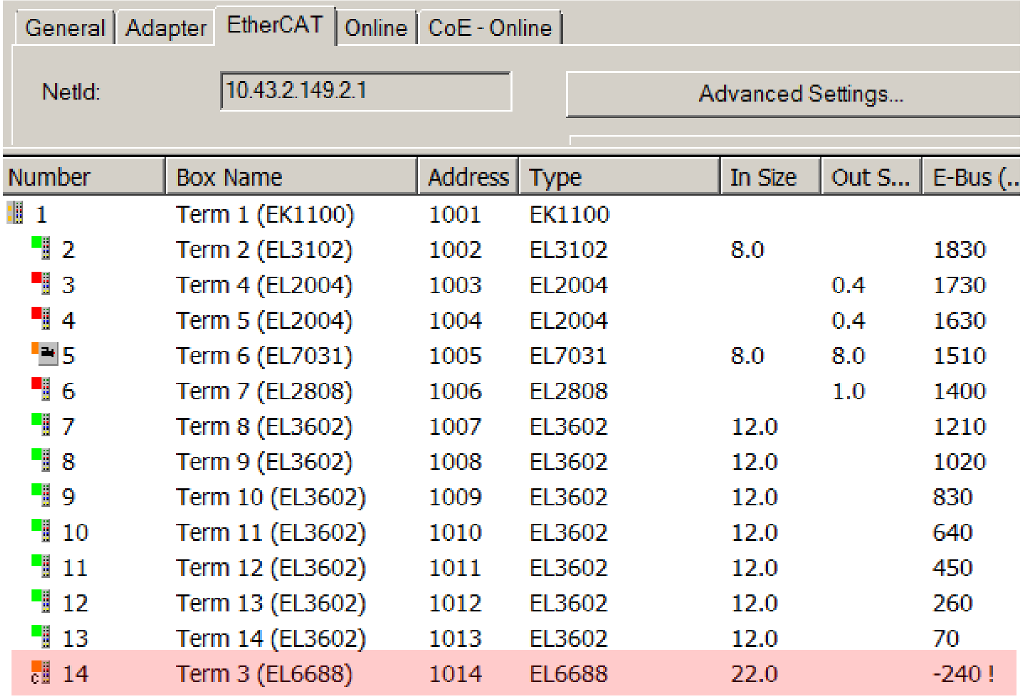
- Bausteine
  - CoE Interface
  - Conversion Functions
  - Distributed Clocks
  - EtherCAT Commands
  - EtherCAT Diagnostic
  - EtherCAT State Machine
    - FB_EcGetAllSlaveStates (FB)
    - FB_EcGetMasterState (FB)
    - FB_EcGetSlaveState (FB)
    - FB_EcReqMasterState (FB)
    - FB_EcReqSlaveState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

Fig. 134: PLC function blocks

Note regarding E-Bus current

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

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

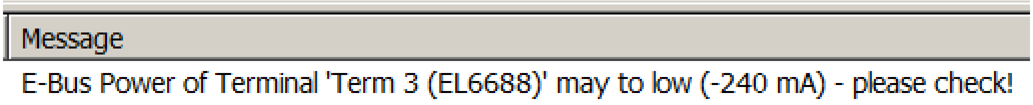


NetId: 10.43.2.149.2.1

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. 135: Illegally exceeding the E-Bus current

From TwinCAT 2.11 and above, a warning message “E-Bus Power of Terminal...” is output in the logger window when such a configuration is activated:



Message

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

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

NOTE

Caution! Malfunction possible!

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

5.4 Basics and commissioning hints

5.4.1 Oversampling terminals/boxes and TwinCAT Scope

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

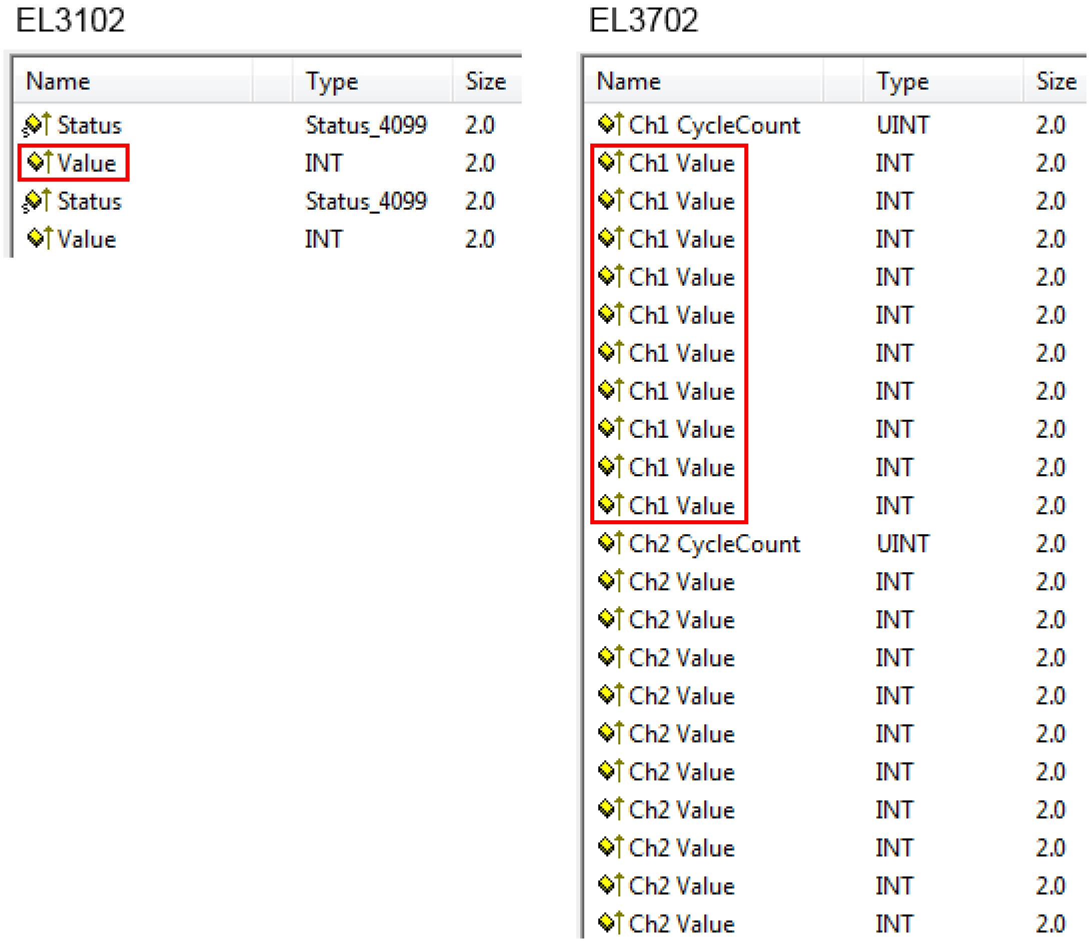
Oversampling means that an analog or digital input device supplies not only one measured value for each process data cycle/EtherCAT cycle (duration T), but several, which are determined at a constant interval t < T. The ratio T/t is the oversampling factor n.

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

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

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

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



EL3102

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

EL3702

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

Fig. 137: Oversampling PDO of the EL37xx series and in the comparison with EL31xx

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

5.4.1.1 TwinCAT 3 procedure

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

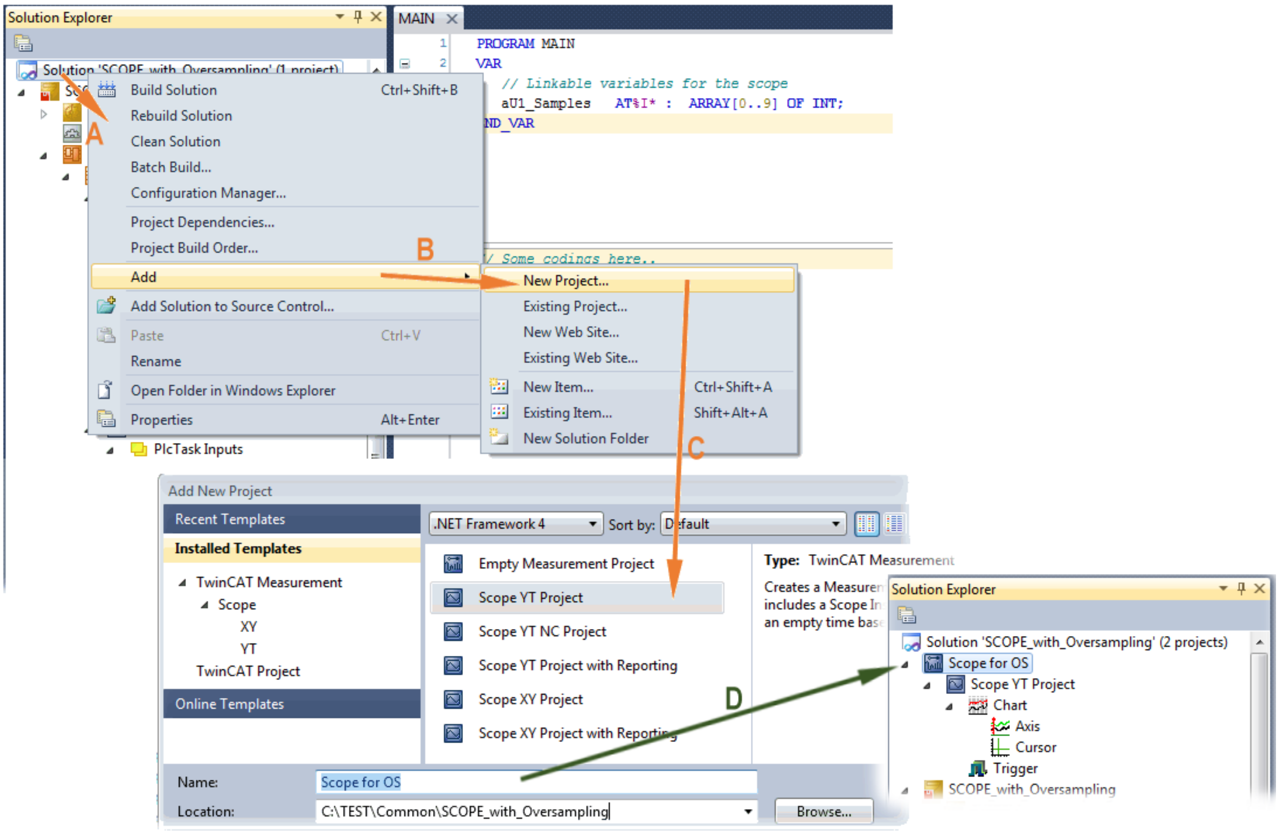
Terminal	Revision
EL4732	all
EL4712	all
EL3783	EL3783-0000-0017
EL3773	EL3773-0000-0019
EL3751	all
EL3742	all
EL3702	all
EL3632	all
EL2262	all
EL1262-0050	all
EL1262	all
EP3632-0001	all
EPP3632-0001	all

Recording a PLC Variable with the TwinCAT 3 – ScopeView

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

Step 1: Adding a project „Scope YT“

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



```
PROGRAM MAIN
VAR
   // Linkable variables for the scope
   aU1_Samples   AT%I* :  ARRAY[0..9] OF INT;
ND_VAR
```

```
/ Some codings here..
```

Name: Scope for OS

Location: C:\TEST\Common\SCOPE_with_Oversampling

Fig. 138: Adding a Scope project into an already existing project

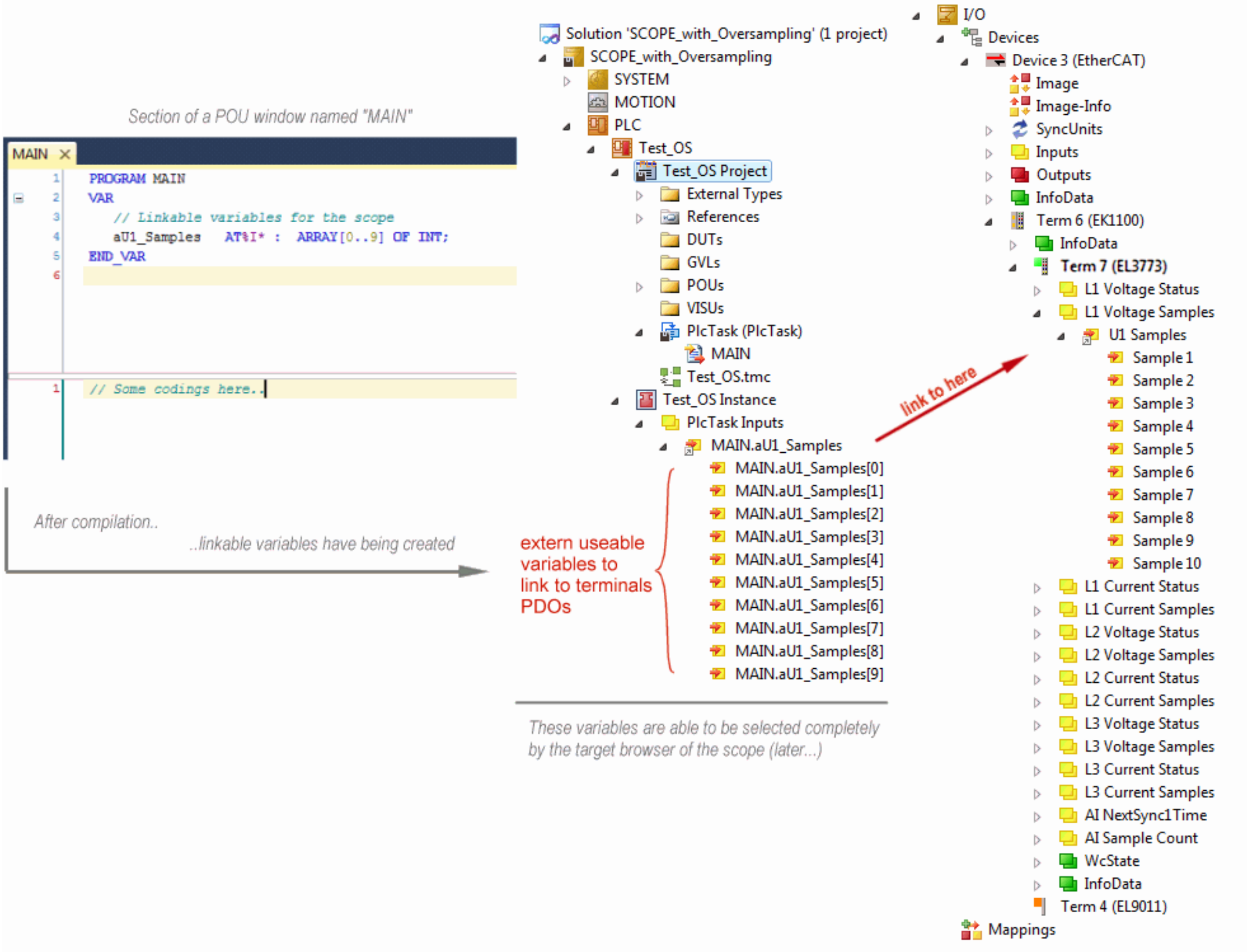
Step 2a: Creation of a PLC variable within a POU

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

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

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

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



Section of a POU window named "MAIN"

```
PROGRAM MAIN
VAR
    // Linkable variables for the scope
    aU1_Samples   AT%I* :  ARRAY[0..9] OF INT;
END_VAR

// Some codings here..
```

After compilation..

..linkable variables have being created

extern useable variables to link to terminals PDOs

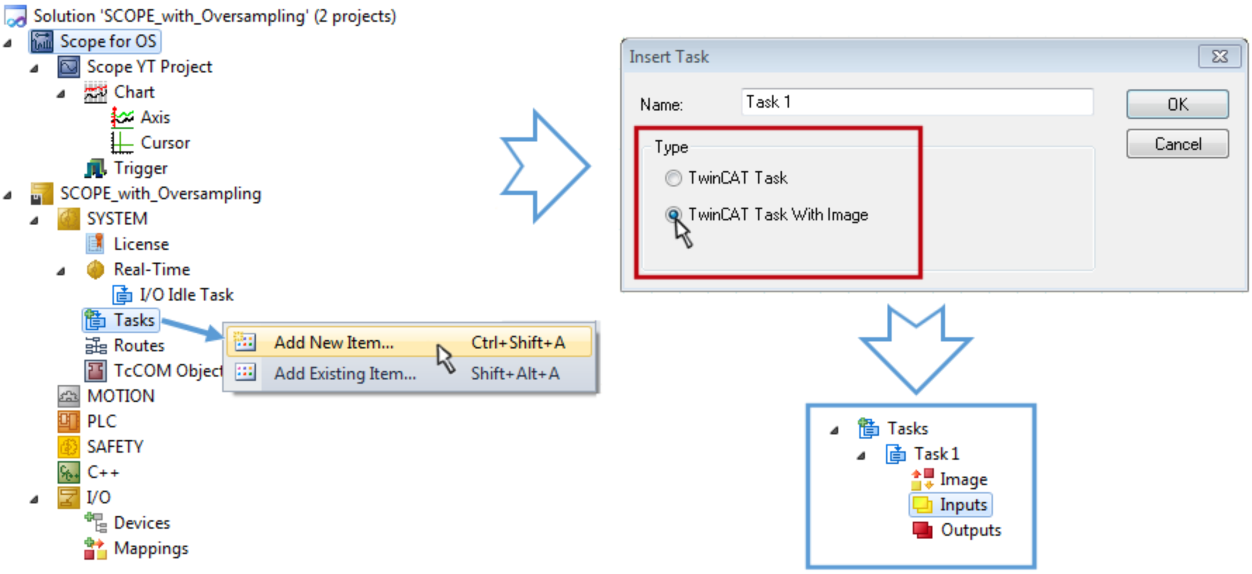
link to here

These variables are able to be selected completely by the target browser of the scope (later...)

Fig. 139: Representation of a created PLC array variable („aUI_Samples“) to link with oversampling PDOs of EL3773

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

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



Insert Task

Name: Task 1

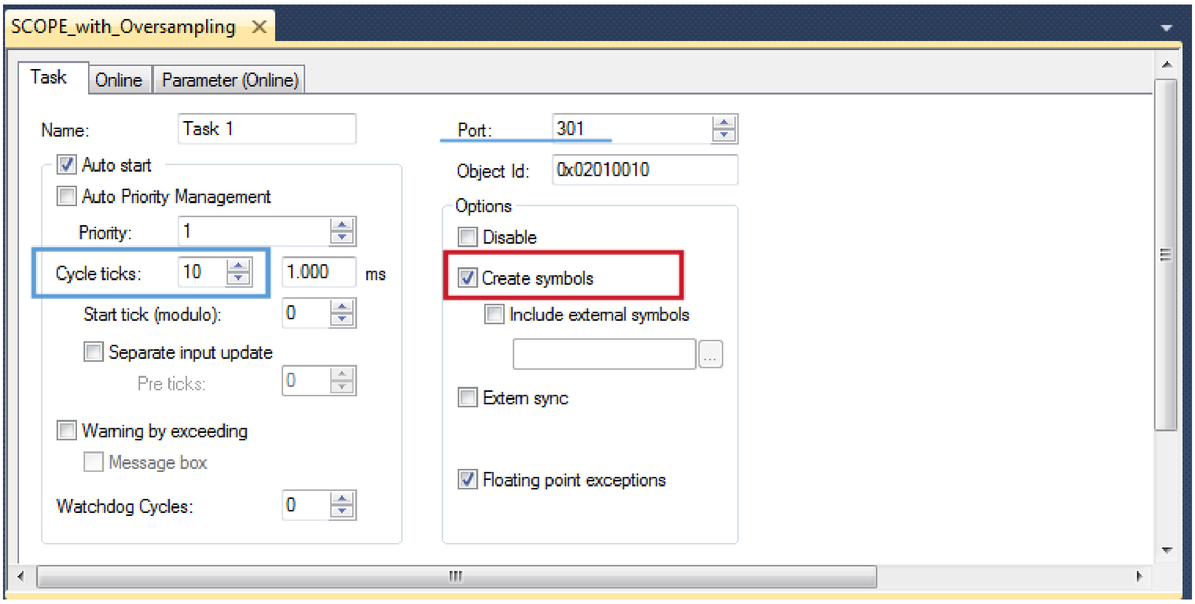
Type

○ TwinCAT Task

● TwinCAT Task With Image

Fig. 140: Insertion of a free task

The Task has to be inserted as “TwinCAT Task With Image” and also creates an “Inputs” and “Outputs” folder therefore. The properties of the new (or as the case may be already existing) task must have activated the attribute “Create symbols” to make them selectable by the “Target Browser” of the Scope later on. The task cycle time has to be changed if so. Then, with10 x Oversampling 1 ms at 100 µs base time, resulting 10 ticks will be set by the usage of the EL3751 for example:



SCOPE_with_Oversampling

Task | Online | Parameter (Online)

Name: Task 1

☑ Auto start

☐ Auto Priority Management

Priority: 1

Cycle ticks: 10 1.000 ms

Start tick (modulo): 0

☐ Separate input update

Pre ticks: 0

☐ Warning by exceeding

☐ Message box

Watchdog Cycles: 0

Port: 301

Object Id: 0x02010010

Options

☐ Disable

☑ Create symbols

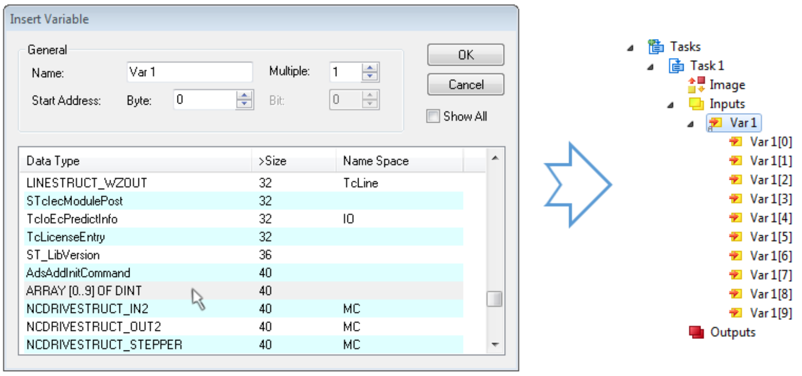
☐ Include external symbols

☐ Extern sync

☑ Floating point exceptions

Fig. 141: Task property "Create symbols" must be activated

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



Insert Variable

General

Name: Var 1 Multiple: 1

Start Address: Byte: 0 Bit: 0

☐ Show All

Data Type	>Size	Name Space
LINESTRUCT_WZOUT	32	TcLine
STclecModulePost	32	
TcloEcPredictInfo	32	IO
TcLicenseEntry	32	
ST_LibVersion	36	
AdsAddInitCommand	40	
ARRAY [0..9] OF DINT	40	
NCDRIVESTRUCT_IN2	40	MC
NCDRIVESTRUCT_OUT2	40	MC
NCDRIVESTRUCT_STEPPER	40	MC

Fig. 142: Insertion of variable "Var 1" fitting to the oversampling (-factor)

Step 3: Linking an array variable with an oversampling PDO

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

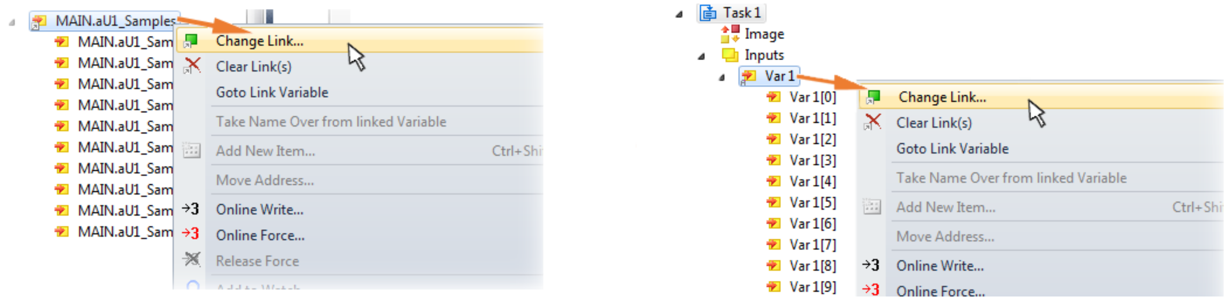
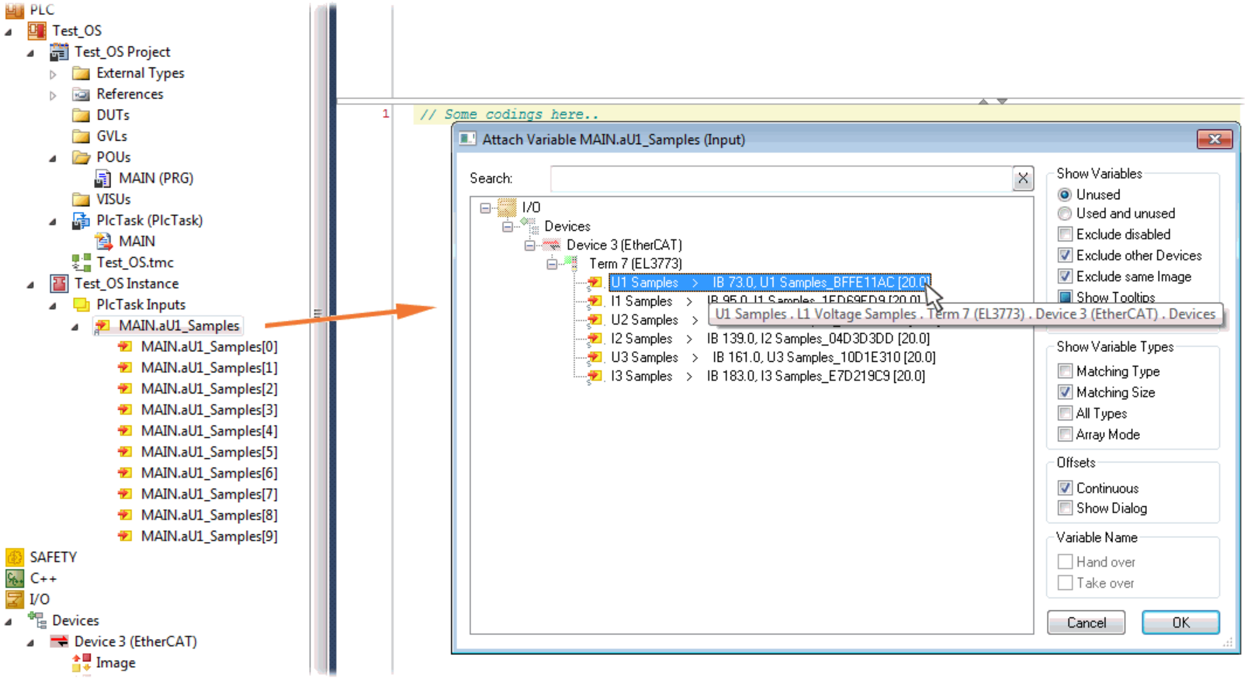


Fig. 143: Set up the link of the PLC array variable (left: for the last preceding paragraph Step 2a, right: for the last preceding paragraph Step 2b)



```
// Some codings here..
```

Attach Variable MAIN.aU1_Samples (Input)

Search:

- I/O
  - Devices
    - Device 3 (EtherCAT)
      - Term 7 (EL3773)
        - U1 Samples > IB 73.0, U1 Samples_BFFE11AC [20.0]
        - I1 Samples > IB 95.0, I1 Samples_1FD69FD9 [20.0]
        - U2 Samples >
        - I2 Samples > IB 139.0, I2 Samples_04D3D3DD [20.0]
        - U3 Samples > IB 161.0, U3 Samples_10D1E310 [20.0]
        - I3 Samples > IB 183.0, I3 Samples_E7D219C9 [20.0]

U1 Samples . L1 Voltage Samples . Term 7 (EL3773) . Device 3 (EtherCAT) . 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

Cancel OK

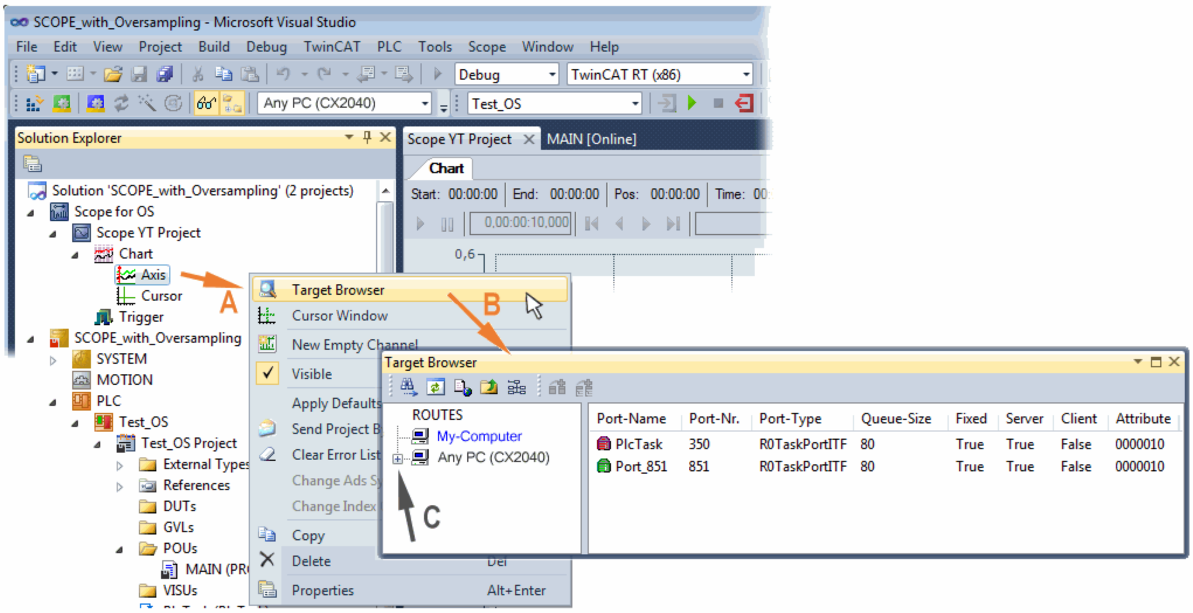
Fig. 144: Select the EL3773 PDO "L1 Voltage Samples" to create a link to the PLC array variable „aUI_Samples“

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

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

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

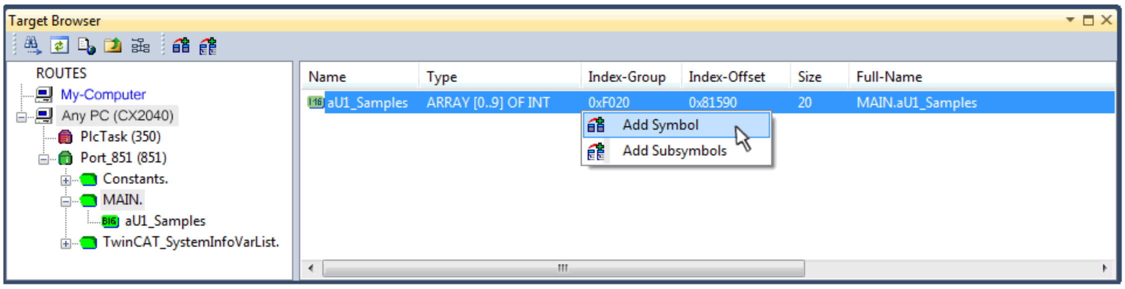
Thereby the drop down menu will be opened by right clicking on “Axis” (A) for selection of the scope features (B):



Port-Name	Port-Nr.	Port-Type	Queue-Size	Fixed	Server	Client	Attribute
PlcTask	350	R0TaskPortITF	80	True	True	False	0000010
Port_851	851	R0TaskPortITF	80	True	True	False	0000010

Fig. 145: Selection of the oversampling variable with the target browser

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

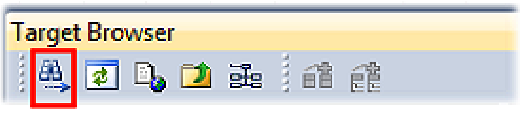


Name	Type	Index-Group	Index-Offset	Size	Full-Name
aU1_Samples	ARRAY [0..9] OF INT	0xF020	0x81590	20	MAIN.aU1_Samples

Fig. 146: Appending the variable "aUI_Samples“ below “axis” within the scope project of the solution explorer

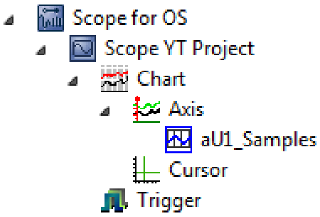
Variable don’t appears into the target browser

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



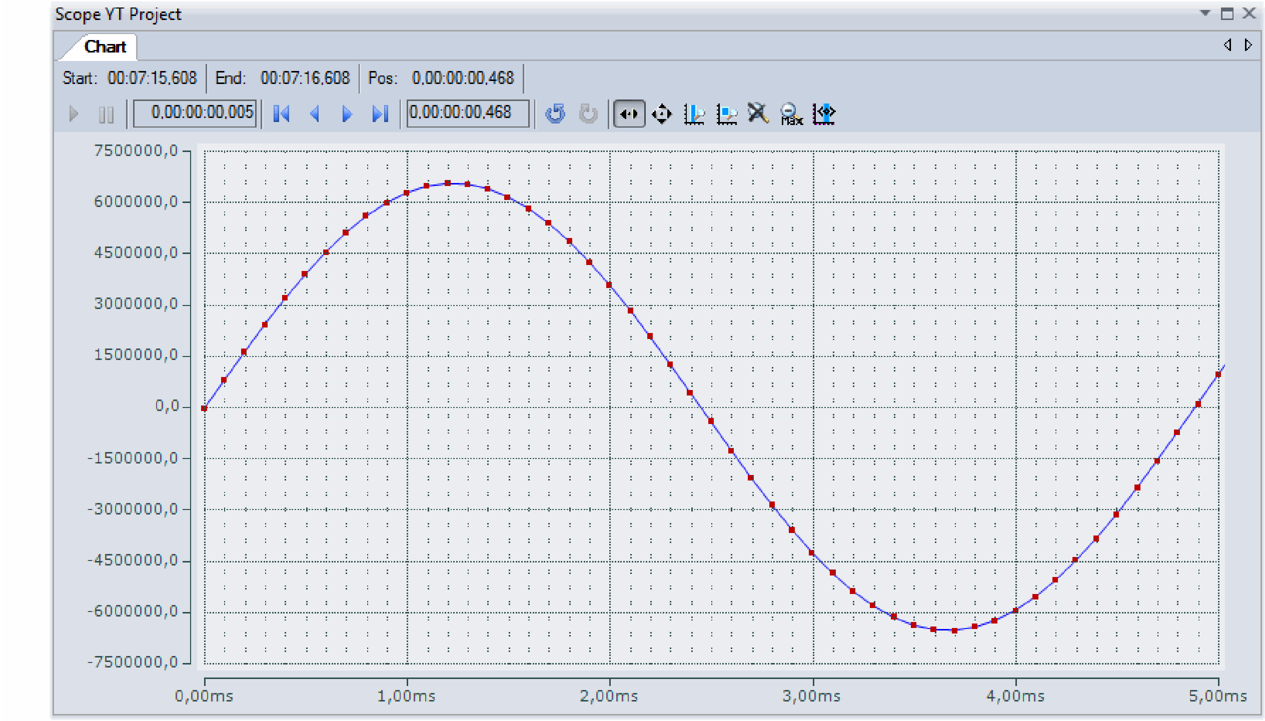
"Enable Server Ports"

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



Now the program start has to be done with ▶ formally although there's no program still yet. Using "Start Recording" the process data value of the oversampling PDO "L1 Voltage Samples " via the linked PLC array variable can be recorded time dependent now.

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



Scope YT Project

Chart

Start: 00:07:15,608 End: 00:07:16,608 Pos: 0,00:00:00,468

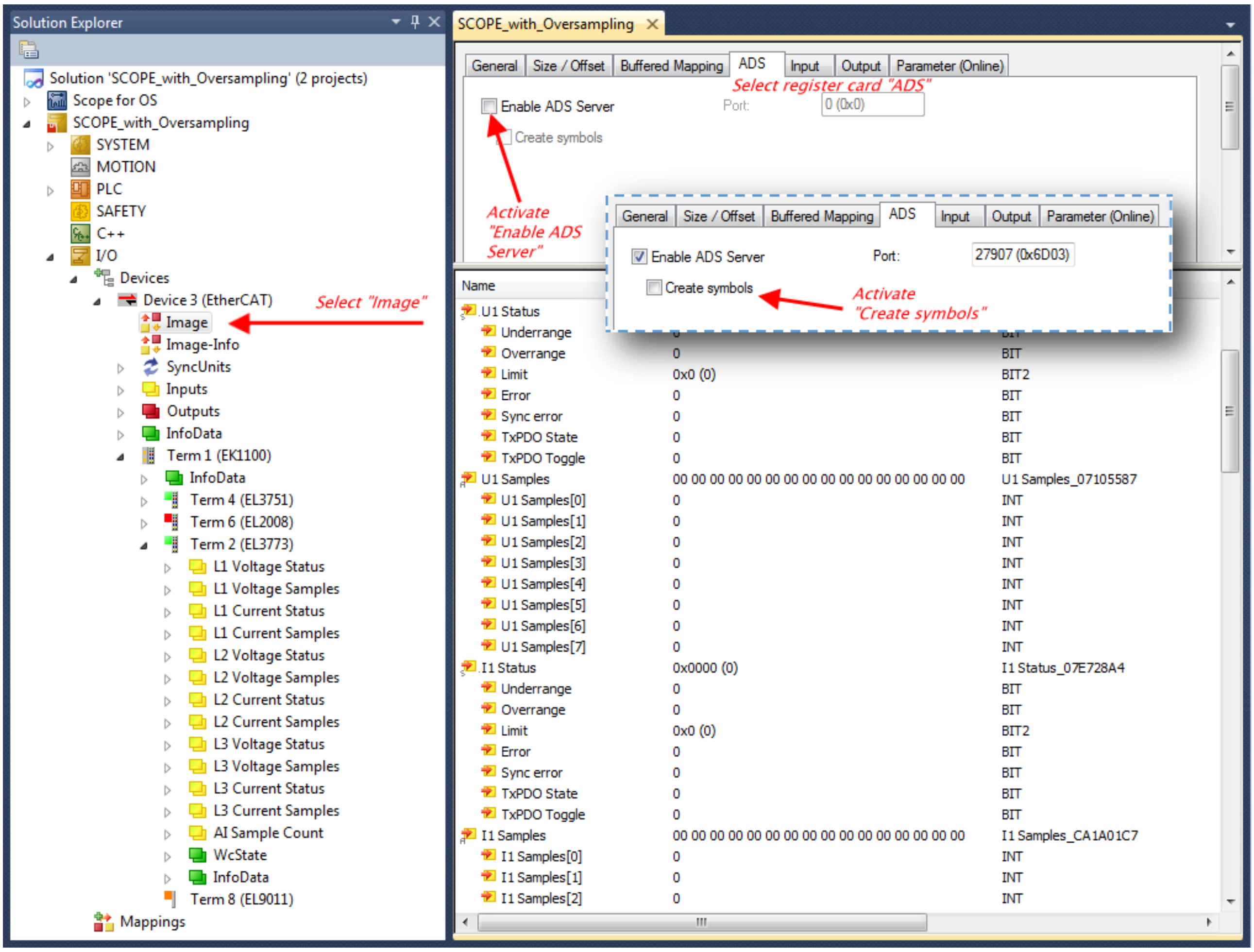
0,00:00:00,005 0,00:00:00,468

Fig. 147: Example of recording a sine signal with 10 x oversampling at 1 ms measurement cycle time

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

Proceeding with / via ADS alternatively

In former TwinCAT 3 versions (or a lower revision as specified in the table [▶ 127] above) the oversampling PDO of the respective oversampling able terminal/box can be made visible for the ScopeView by activation of the ADS server.



Solution Explorer

- Solution 'SCOPE_with_Oversampling' (2 projects)
  - Scope for OS
  - SCOPE_with_Oversampling
    - SYSTEM
    - MOTION
    - PLC
    - SAFETY
    - C++
    - I/O
      - Devices
        - Device 3 (EtherCAT)
          - Image
          - Image-Info
          - SyncUnits
          - Inputs
          - Outputs
          - InfoData
          - Term 1 (EK1100)
            - InfoData
            - Term 4 (EL3751)
            - Term 6 (EL2008)
            - Term 2 (EL3773)
              - L1 Voltage Status
              - L1 Voltage Samples
              - L1 Current Status
              - L1 Current Samples
              - L2 Voltage Status
              - L2 Voltage Samples
              - L2 Current Status
              - L2 Current Samples
              - L3 Voltage Status
              - L3 Voltage Samples
              - L3 Current Status
              - L3 Current Samples
              - AI Sample Count
              - WcState
              - InfoData
            - Term 8 (EL9011)
      - Mappings

Select "Image"

SCOPE_with_Oversampling

General | Size / Offset | Buffered Mapping | ADS | Input | Output | Parameter (Online)

Select register card "ADS"

□ Enable ADS Server Port: 0 (0x0)

□ Create symbols

Activate "Enable ADS Server"

General | Size / Offset | Buffered Mapping | ADS | Input | Output | Parameter (Online)

☑ Enable ADS Server Port: 27907 (0x6D03)

□ Create symbols

Activate "Create symbols"

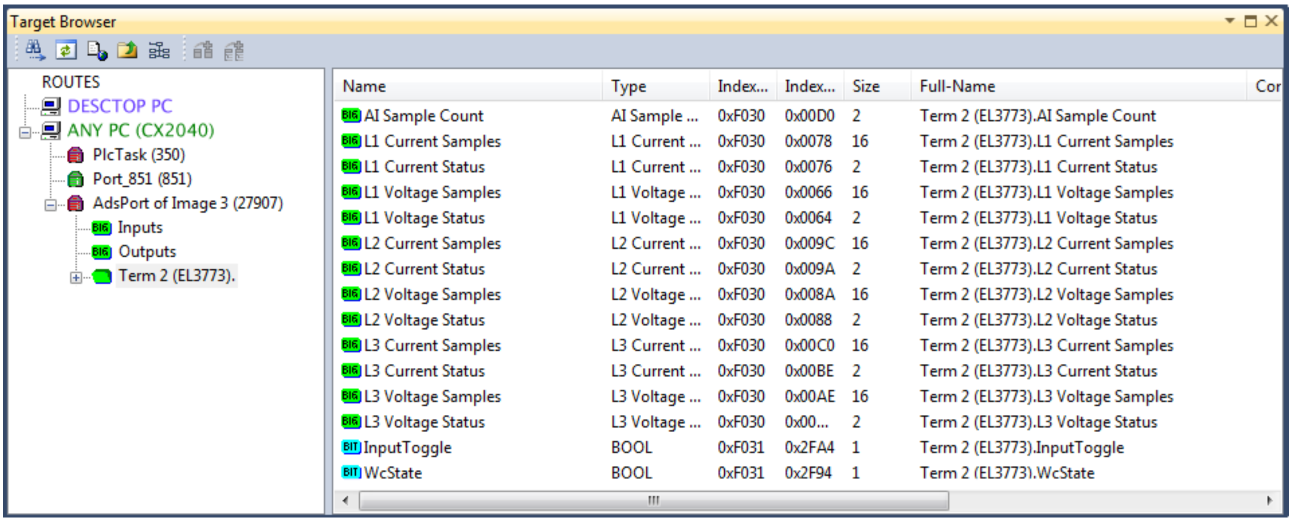
Name		
U1 Status		
Underrange	0	BIT
Overrange	0	BIT
Limit	0x0 (0)	BIT2
Error	0	BIT
Sync error	0	BIT
TxPDO State	0	BIT
TxPDO Toggle	0	BIT
U1 Samples	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	U1 Samples_07105587
U1 Samples[0]	0	INT
U1 Samples[1]	0	INT
U1 Samples[2]	0	INT
U1 Samples[3]	0	INT
U1 Samples[4]	0	INT
U1 Samples[5]	0	INT
U1 Samples[6]	0	INT
U1 Samples[7]	0	INT
I1 Status	0x0000 (0)	I1 Status_07E728A4
Underrange	0	BIT
Overrange	0	BIT
Limit	0x0 (0)	BIT2
Error	0	BIT
Sync error	0	BIT
TxPDO State	0	BIT
TxPDO Toggle	0	BIT
I1 Samples	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	I1 Samples_CA1A01C7
I1 Samples[0]	0	INT
I1 Samples[1]	0	INT
I1 Samples[2]	0	INT

Fig. 148: Activation of the ADS server of the EtherCAT device (TwinCAT 3)

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

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

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



Target Browser

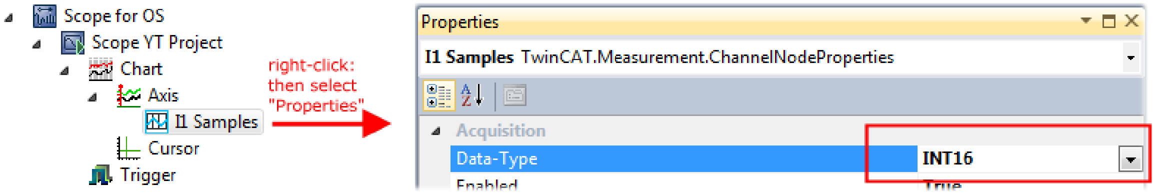
ROUTES
- DESCTOP PC
- ANY PC (CX2040)
  - PlcTask (350)
  - Port_851 (851)
  - AdsPort of Image 3 (27907)
    - Inputs
    - Outputs
    - Term 2 (EL3773).

Name	Type	Index...	Index...	Size	Full-Name	Cor
AI Sample Count	AI Sample ...	0xF030	0x00D0	2	Term 2 (EL3773).AI Sample Count	
L1 Current Samples	L1 Current ...	0xF030	0x0078	16	Term 2 (EL3773).L1 Current Samples	
L1 Current Status	L1 Current ...	0xF030	0x0076	2	Term 2 (EL3773).L1 Current Status	
L1 Voltage Samples	L1 Voltage ...	0xF030	0x0066	16	Term 2 (EL3773).L1 Voltage Samples	
L1 Voltage Status	L1 Voltage ...	0xF030	0x0064	2	Term 2 (EL3773).L1 Voltage Status	
L2 Current Samples	L2 Current ...	0xF030	0x009C	16	Term 2 (EL3773).L2 Current Samples	
L2 Current Status	L2 Current ...	0xF030	0x009A	2	Term 2 (EL3773).L2 Current Status	
L2 Voltage Samples	L2 Voltage ...	0xF030	0x008A	16	Term 2 (EL3773).L2 Voltage Samples	
L2 Voltage Status	L2 Voltage ...	0xF030	0x0088	2	Term 2 (EL3773).L2 Voltage Status	
L3 Current Samples	L3 Current ...	0xF030	0x00C0	16	Term 2 (EL3773).L3 Current Samples	
L3 Current Status	L3 Current ...	0xF030	0x00BE	2	Term 2 (EL3773).L3 Current Status	
L3 Voltage Samples	L3 Voltage ...	0xF030	0x00AE	16	Term 2 (EL3773).L3 Voltage Samples	
L3 Voltage Status	L3 Voltage ...	0xF030	0x00...	2	Term 2 (EL3773).L3 Voltage Status	
InputToggle	BOOL	0xF031	0x2FA4	1	Term 2 (EL3773).InputToggle	
WcState	BOOL	0xF031	0x2F94	1	Term 2 (EL3773).WcState	

Fig. 149: Direct access to PDOs of the terminal by ScopeView

Data type not valid

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



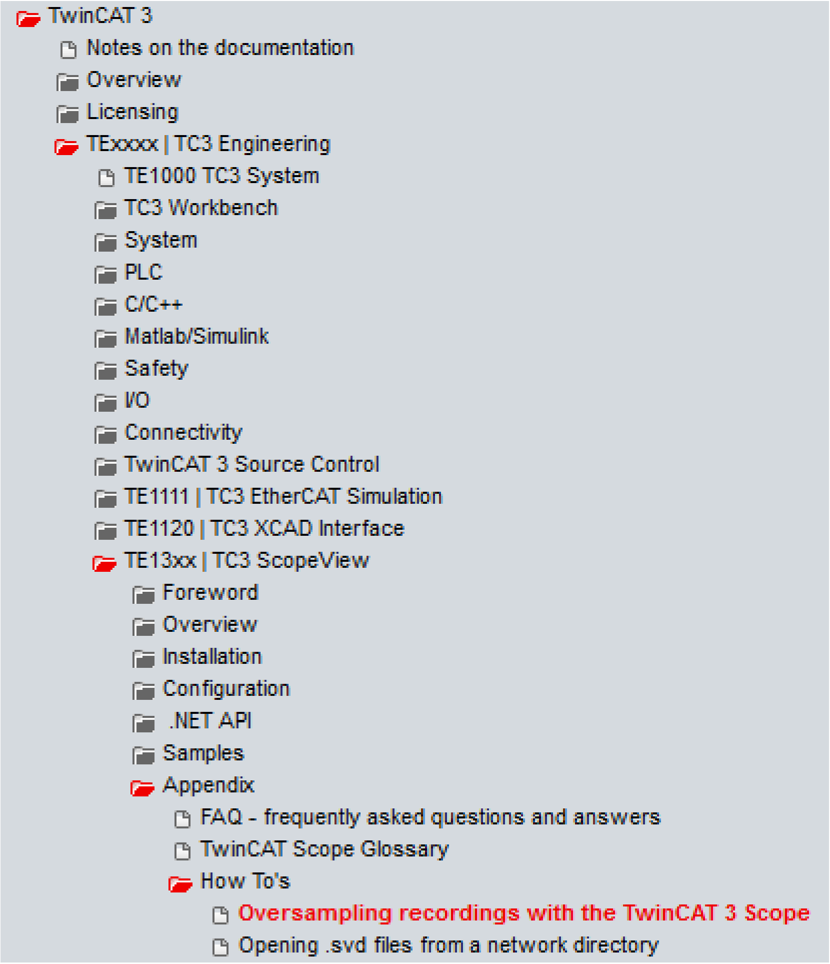
I1 Samples TwinCAT.Measurement.ChannelNodeProperties

Data-Type INT16

Enabled True

TwinCAT 3: Activate the ADS Server of an EtherCAT device

Also see Beckhoff Information System:



- TwinCAT 3
  - Notes on the documentation
  - Overview
  - Licensing
  - TExxxx | TC3 Engineering
    - TE1000 TC3 System
    - TC3 Workbench
    - System
    - PLC
    - C/C++
    - Matlab/Simulink
    - Safety
    - I/O
    - Connectivity
    - TwinCAT 3 Source Control
    - TE1111 | TC3 EtherCAT Simulation
    - TE1120 | TC3 XCAD Interface
    - TE13xx | TC3 ScopeView
      - Foreword
      - Overview
      - Installation
      - Configuration
      - .NET API
      - Samples
      - Appendix
        - FAQ - frequently asked questions and answers
        - TwinCAT Scope Glossary
        - How To's
          - Oversampling recordings with the TwinCAT 3 Scope
          - Opening .svd files from a network directory

5.4.1.2 TwinCAT 2 procedure

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

System requirements

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

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

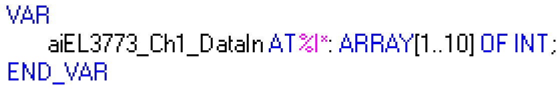
- in the PLC, see step 1a [▸ 136]
- or directly in the System Manager if only one free task is present, see step 1b [▸ 136]

The same settings are to be made in the Scope2 for both cases, see step 2 [▸ 138]

Recording of a PLC variable with the TwinCAT 2 – Scope2

Step 1a: TwinCAT 2 PLC

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



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

Fig. 150: PLC declaration

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

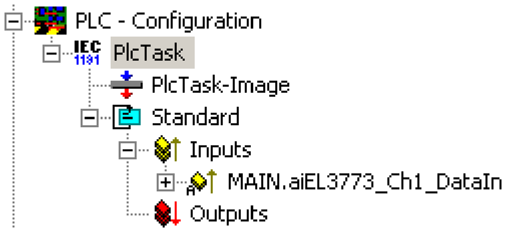


Fig. 151: PLC in the System Manager

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

Step 1b: TwinCAT 2 - free task

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

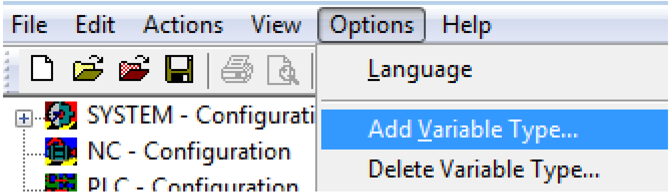
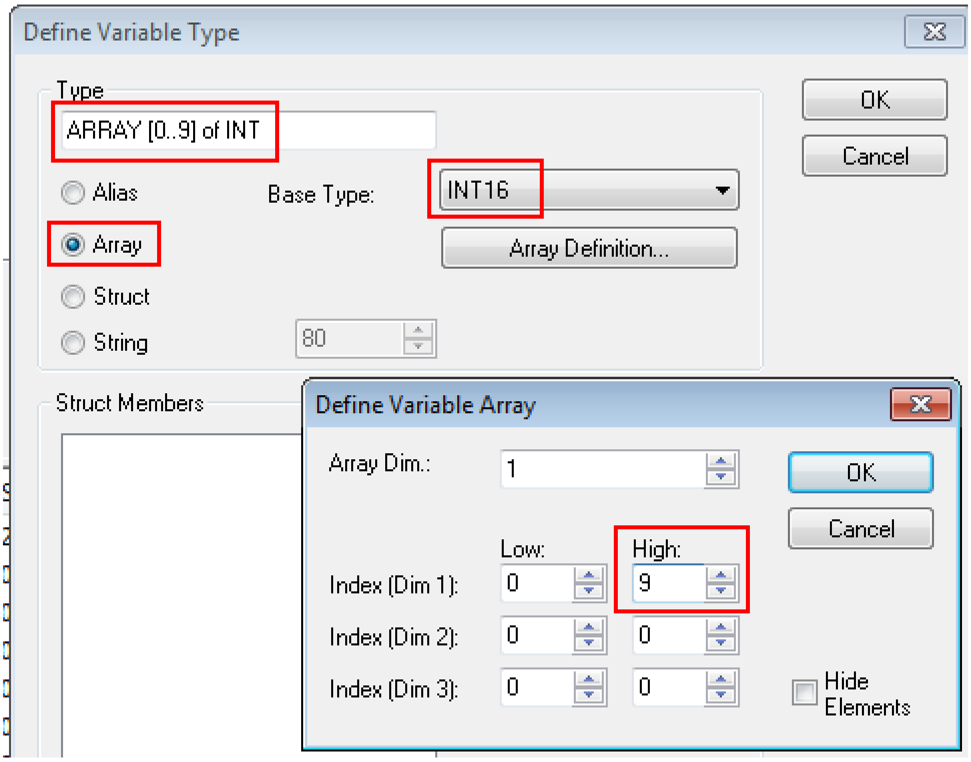


Fig. 152: Add Variable Type

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



Define Variable Type

Type: ARRAY [0..9] of INT

○ Alias

● Array

○ Struct

○ String 80

Base Type: INT16

Define Variable Array

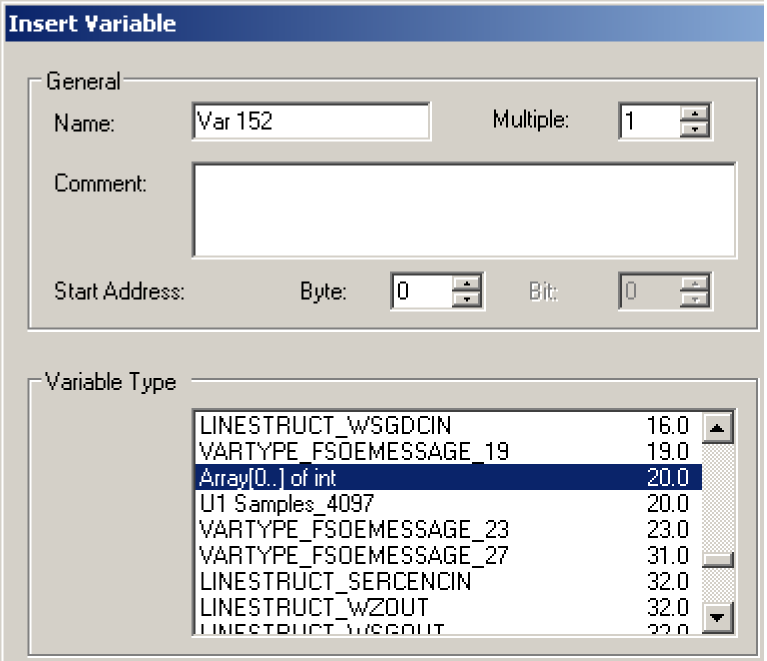
Array Dim.: 1

	Low:	High:
Index (Dim 1):	0	9
Index (Dim 2):	0	0
Index (Dim 3):	0	0

□ Hide Elements

Fig. 153: Definition of the variable type

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



Insert Variable

General

Name: Var 152

Multiple: 1

Comment:

Start Address: Byte: 0 Bit: 0

Variable Type

LINESTRUCT_WSGDCIN	16.0
VARTYPE_FSOEMESSAGE_19	19.0
Array[0..] of int	20.0
U1 Samples_4097	20.0
VARTYPE_FSOEMESSAGE_23	23.0
VARTYPE_FSOEMESSAGE_27	31.0
LINESTRUCT_SERCENCIN	32.0
LINESTRUCT_WZOUT	32.0
LINESTRUCT_WSGOUT	32.0

Fig. 154: Overview of declared types

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

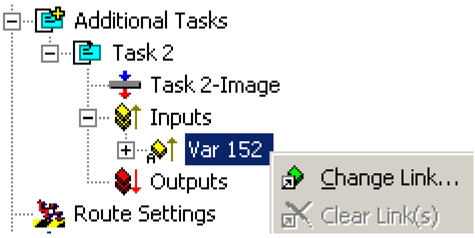
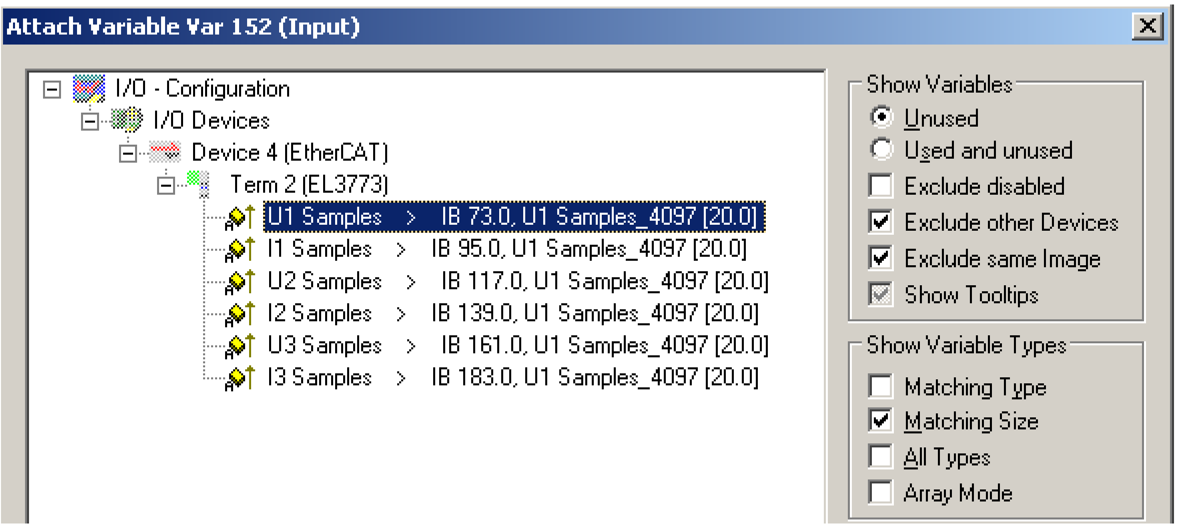


Fig. 155: Linking

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


Attach Variable Var 152 (Input)

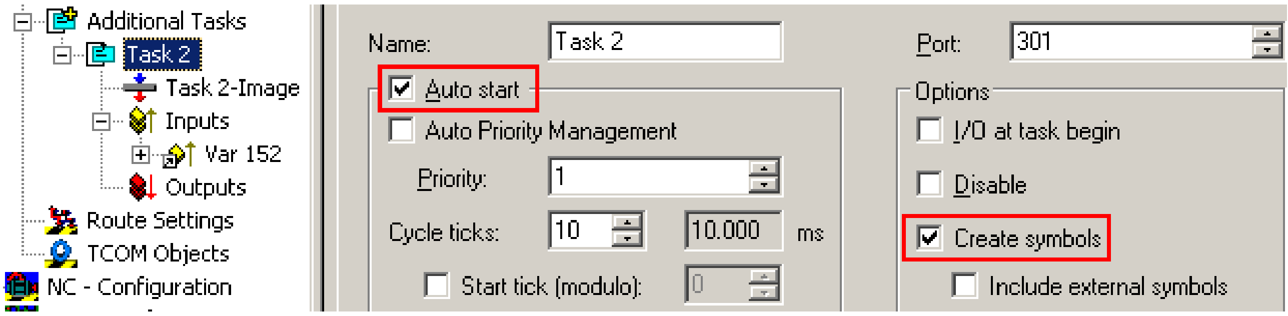
- I/O - Configuration
  - I/O Devices
    - Device 4 (EtherCAT)
      - Term 2 (EL3773)
        - U1 Samples > IB 73.0, U1 Samples_4097 [20.0]
        - I1 Samples > IB 95.0, U1 Samples_4097 [20.0]
        - U2 Samples > IB 117.0, U1 Samples_4097 [20.0]
        - I2 Samples > IB 139.0, U1 Samples_4097 [20.0]
        - U3 Samples > IB 161.0, U1 Samples_4097 [20.0]
        - I3 Samples > IB 183.0, U1 Samples_4097 [20.0]

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

Fig. 156: Array variables of an oversampling terminal

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



Name: Task 2
Port: 301
☑ Auto start
□ Auto Priority Management
Priority: 1
Cycle ticks: 10 10.000 ms
□ Start tick (modulo): 0

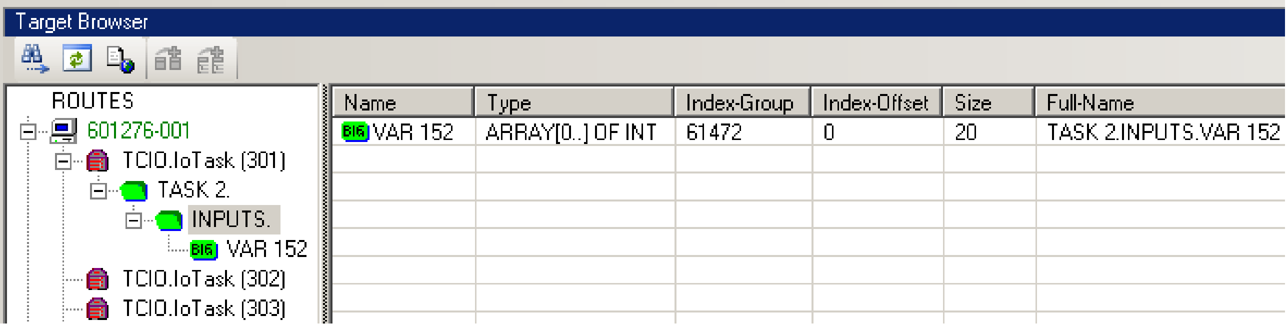
Options
□ I/O at task begin
□ Disable
☑ Create symbols
□ Include external symbols

Fig. 157: Settings in the additional task

Step 2: Configuration in the Scope2

So that the linking works, an array variable with the channel data of the respective terminal/box must be present in the system manager; i.e. each oversampling data package must be present in an array. This array variable must be defined and created manually; see above [▶ 136].

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



Target Browser

ROUTES
- 601276-001
  - TCIO.IoTask (301)
    - TASK 2.
      - INPUTS.
        - VAR 152
  - TCIO.IoTask (302)
  - TCIO.IoTask (303)

Name	Type	Index-Group	Index-Offset	Size	Full-Name
VAR 152	ARRAY[0..] OF INT	61472	0	20	TASK 2.INPUTS.VAR 152

Fig. 158: Variable browser up to the array VAR152

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

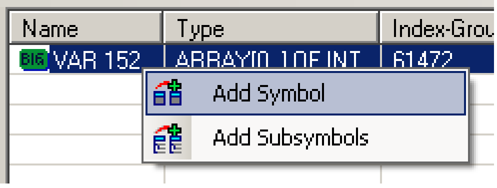
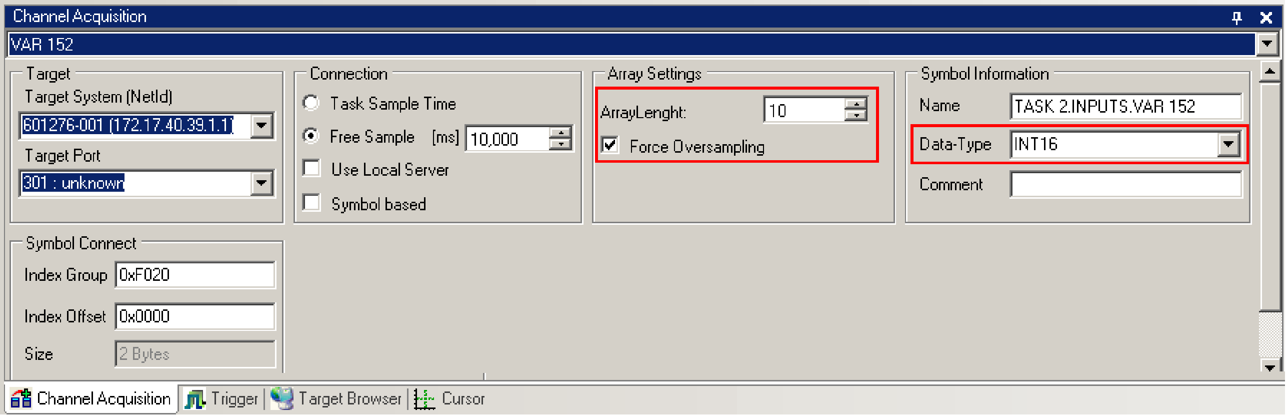


Fig. 159: AddSymbol on the array

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



Channel Acquisition

VAR 152

Target
Target System (NetId): 601276-001 (172.17.40.39.1.1)
Target Port: 301 : unknown

Connection
○ Task Sample Time
● Free Sample [ms] 10,000
□ Use Local Server
□ Symbol based

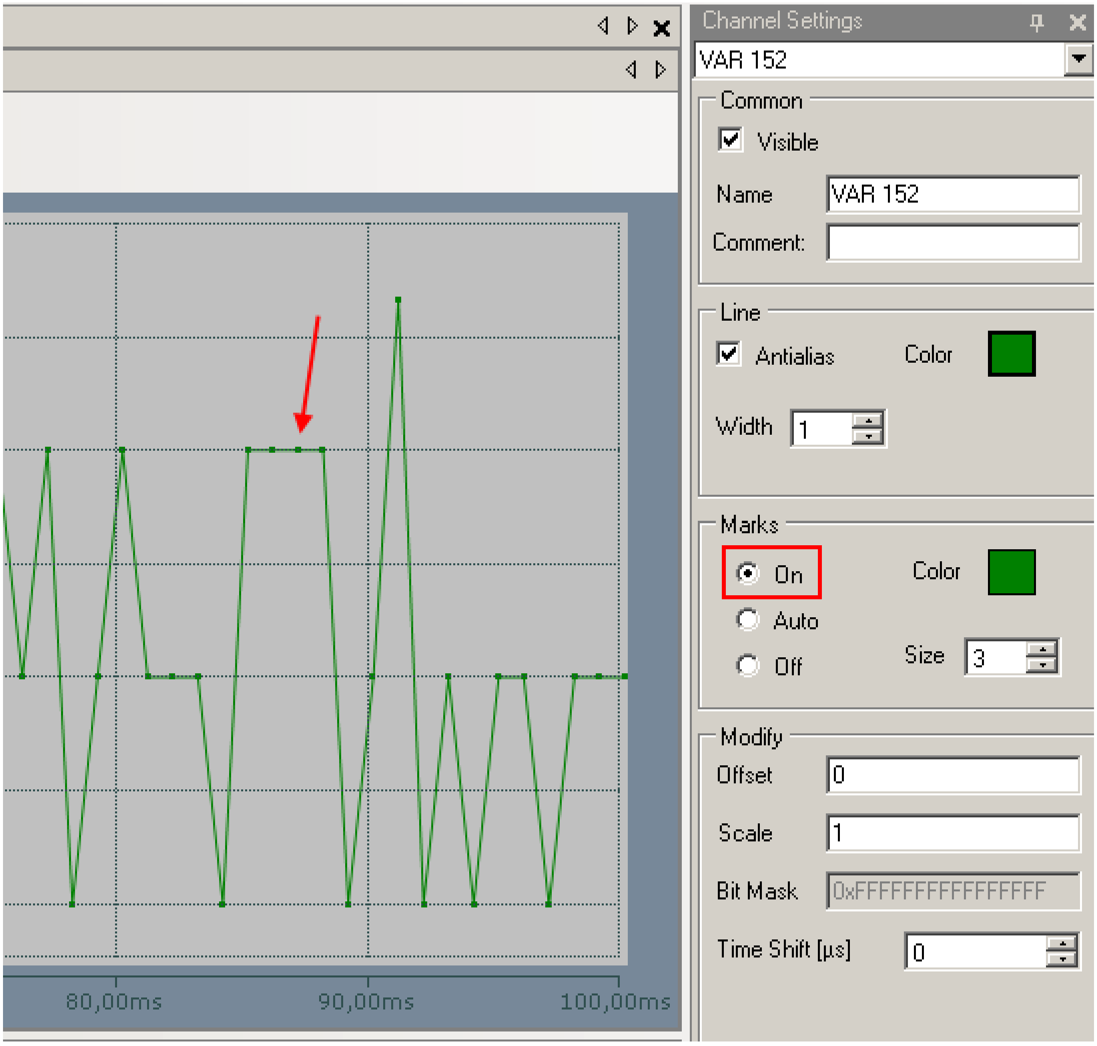
Array Settings
ArrayLenght: 10
☑ Force Oversampling

Symbol Information
Name: TASK 2.INPUTS.VAR 152
Data-Type: INT16
Comment:

Symbol Connect
Index Group: 0xF020
Index Offset: 0x0000
Size: 2 Bytes

Fig. 160: Channel settings

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



Channel Settings

VAR 152

Common

☑ Visible

Name: VAR 152

Comment:

Line

☑ Antialias

Color 🟩

Width: 1

Marks

● On

○ Auto

○ Off

Color 🟩

Size: 3

Modify

Offset: 0

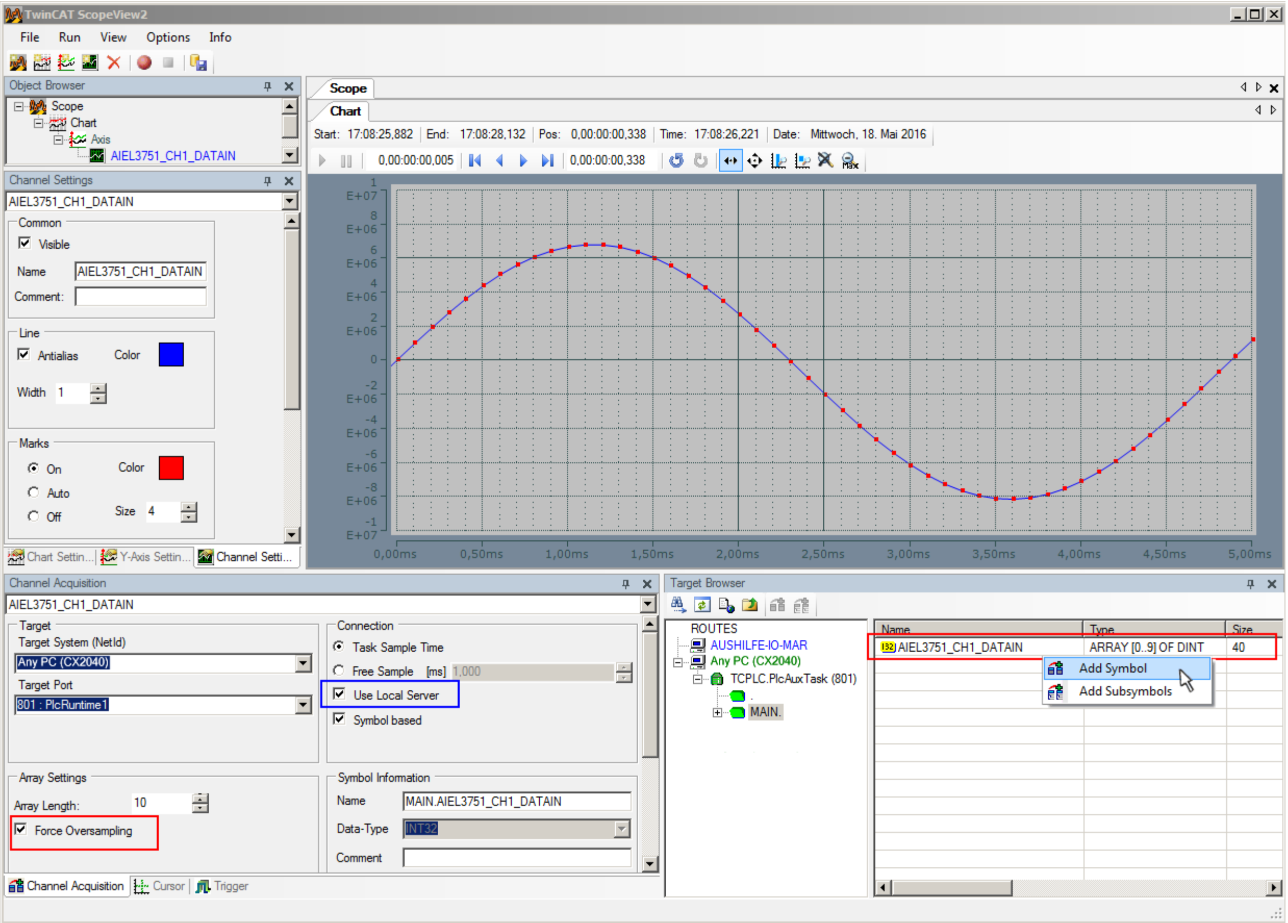
Scale: 1

Bit Mask: 0xFFFFFFFFFFFFFFFF

Time Shift [µs]: 0

Fig. 161: Activation of the marks

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



Start: 17:08:25,882 End: 17:08:28,132 Pos: 0,00:00:00,338 Time: 17:08:26,221 Date: Mittwoch, 18. Mai 2016

Channel Settings: AIEL3751_CH1_DATAIN

Common: ☑ Visible; Name: AIEL3751_CH1_DATAIN; Comment:

Line: ☑ Antialias; Color: 🟦; Width: 1

Marks: ● On ○ Auto ○ Off; Color: 🟥; Size: 4

Channel Acquisition: AIEL3751_CH1_DATAIN

Target: Target System (NetId): Any PC (CX2040); Target Port: 801 : PlcRuntime1

Connection: ● Task Sample Time ○ Free Sample [ms] 1.000; ☑ Use Local Server; ☑ Symbol based

Array Settings: Array Length: 10; ☑ Force Oversampling

Symbol Information: Name: MAIN.AIEL3751_CH1_DATAIN; Data-Type: INT32; Comment:

Name	Type	Size
AIEL3751_CH1_DATAIN	ARRAY [0..9] OF DINT	40

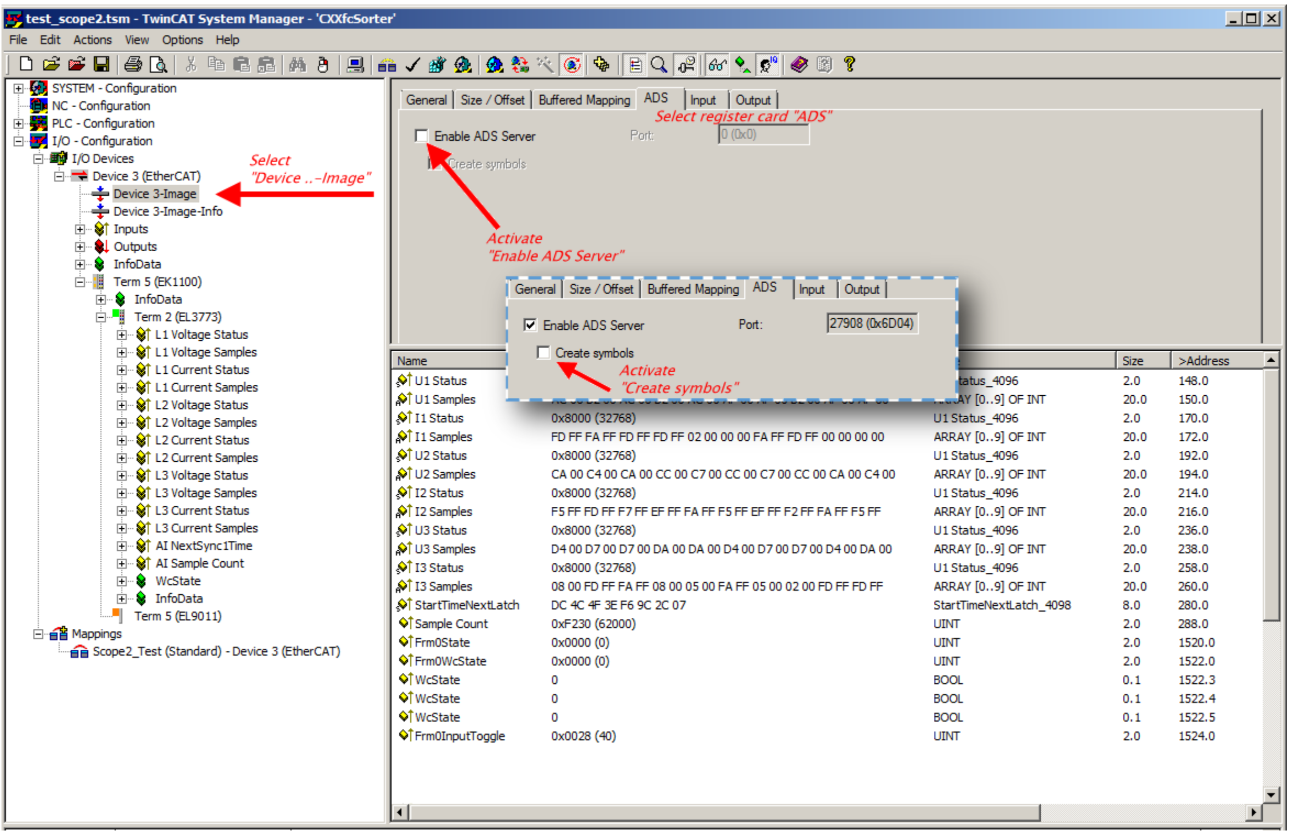
Fig. 162: Illustration of a 10 x oversampling variable of the EL3751 by the Scope2

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

Proceeding with TwinCAT 2/ alternatively via ADS

In former TwinCAT 2 versions (or a lower revision as specified in the table [▶ 127] above) the oversampling PDO of the respective oversampling able terminal/box can be made visible for the Scope2 by activation of the ADS server.

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



□ Enable ADS Server Port: 0 (0x0)

□ Create symbols

☑ Enable ADS Server Port: 27908 (0x6D04)

□ Create symbols

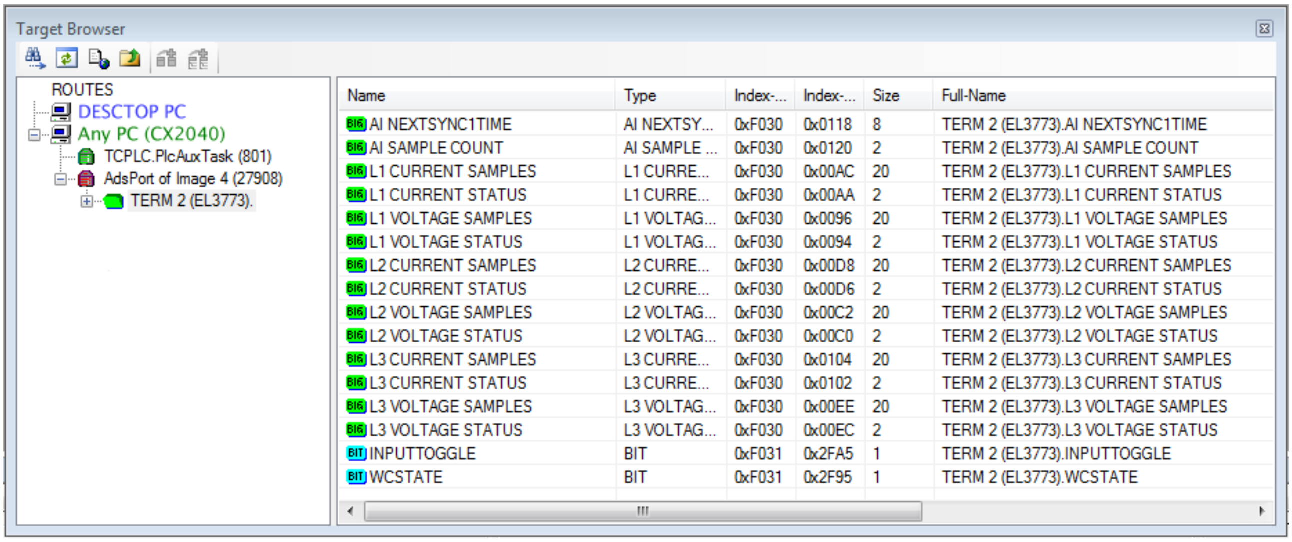
Name			Size	>Address
U1 Status		tatus_4096	2.0	148.0
U1 Samples		AY [0..9] OF INT	20.0	150.0
I1 Status	0x8000 (32768)	U1 Status_4096	2.0	170.0
I1 Samples	FD FF FA FF FD FF FD FF 02 00 00 00 FA FF FD FF 00 00 00 00	ARRAY [0..9] OF INT	20.0	172.0
U2 Status	0x8000 (32768)	U1 Status_4096	2.0	192.0
U2 Samples	CA 00 C4 00 CA 00 CC 00 C7 00 CC 00 C7 00 CC 00 CA 00 C4 00	ARRAY [0..9] OF INT	20.0	194.0
I2 Status	0x8000 (32768)	U1 Status_4096	2.0	214.0
I2 Samples	F5 FF FD FF F7 FF EF FF FA FF F5 FF EF FF F2 FF FA FF F5 FF	ARRAY [0..9] OF INT	20.0	216.0
U3 Status	0x8000 (32768)	U1 Status_4096	2.0	236.0
U3 Samples	D4 00 D7 00 D7 00 DA 00 DA 00 D4 00 D7 00 D7 00 D4 00 DA 00	ARRAY [0..9] OF INT	20.0	238.0
I3 Status	0x8000 (32768)	U1 Status_4096	2.0	258.0
I3 Samples	08 00 FD FF FA FF 08 00 05 00 FA FF 05 00 02 00 FD FF FD FF	ARRAY [0..9] OF INT	20.0	260.0
StartTimeNextLatch	DC 4C 4F 3E F6 9C 2C 07	StartTimeNextLatch_4098	8.0	280.0
Sample Count	0xF230 (62000)	UINT	2.0	288.0
Frm0State	0x0000 (0)	UINT	2.0	1520.0
Frm0WcState	0x0000 (0)	UINT	2.0	1522.0
WcState	0	BOOL	0.1	1522.3
WcState	0	BOOL	0.1	1522.4
WcState	0	BOOL	0.1	1522.5
Frm0InputToggle	0x0028 (40)	UINT	2.0	1524.0

Fig. 163: Activation of the ADS server of the EtherCAT Device (TwinCAT 2)

The activation of the ADS server have to be carried out by selection of the “Device – Image” on the left sided configuration tree:
„I/O – Configuration → I/O Devices → Device .. (EtherCAT) → Device .. – Image“.

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

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

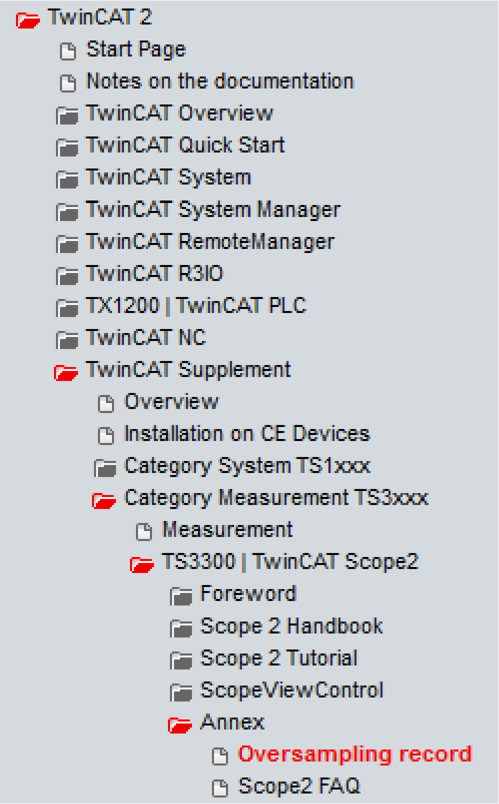


Name	Type	Index-...	Index-...	Size	Full-Name
AI NEXTSYNC1TIME	AI NEXTSY...	0xF030	0x0118	8	TERM 2 (EL3773).AI NEXTSYNC1TIME
AI SAMPLE COUNT	AI SAMPLE ...	0xF030	0x0120	2	TERM 2 (EL3773).AI SAMPLE COUNT
L1 CURRENT SAMPLES	L1 CURRE...	0xF030	0x00AC	20	TERM 2 (EL3773).L1 CURRENT SAMPLES
L1 CURRENT STATUS	L1 CURRE...	0xF030	0x00AA	2	TERM 2 (EL3773).L1 CURRENT STATUS
L1 VOLTAGE SAMPLES	L1 VOLTAG...	0xF030	0x0096	20	TERM 2 (EL3773).L1 VOLTAGE SAMPLES
L1 VOLTAGE STATUS	L1 VOLTAG...	0xF030	0x0094	2	TERM 2 (EL3773).L1 VOLTAGE STATUS
L2 CURRENT SAMPLES	L2 CURRE...	0xF030	0x00D8	20	TERM 2 (EL3773).L2 CURRENT SAMPLES
L2 CURRENT STATUS	L2 CURRE...	0xF030	0x00D6	2	TERM 2 (EL3773).L2 CURRENT STATUS
L2 VOLTAGE SAMPLES	L2 VOLTAG...	0xF030	0x00C2	20	TERM 2 (EL3773).L2 VOLTAGE SAMPLES
L2 VOLTAGE STATUS	L2 VOLTAG...	0xF030	0x00C0	2	TERM 2 (EL3773).L2 VOLTAGE STATUS
L3 CURRENT SAMPLES	L3 CURRE...	0xF030	0x0104	20	TERM 2 (EL3773).L3 CURRENT SAMPLES
L3 CURRENT STATUS	L3 CURRE...	0xF030	0x0102	2	TERM 2 (EL3773).L3 CURRENT STATUS
L3 VOLTAGE SAMPLES	L3 VOLTAG...	0xF030	0x00EE	20	TERM 2 (EL3773).L3 VOLTAGE SAMPLES
L3 VOLTAGE STATUS	L3 VOLTAG...	0xF030	0x00EC	2	TERM 2 (EL3773).L3 VOLTAGE STATUS
INPUTTOGGLE	BIT	0xF031	0x2FA5	1	TERM 2 (EL3773).INPUTTOGGLE
WCSTATE	BIT	0xF031	0x2F95	1	TERM 2 (EL3773).WCSTATE

Fig. 164: Direct access of the Scope2 to the terminal's PDOs

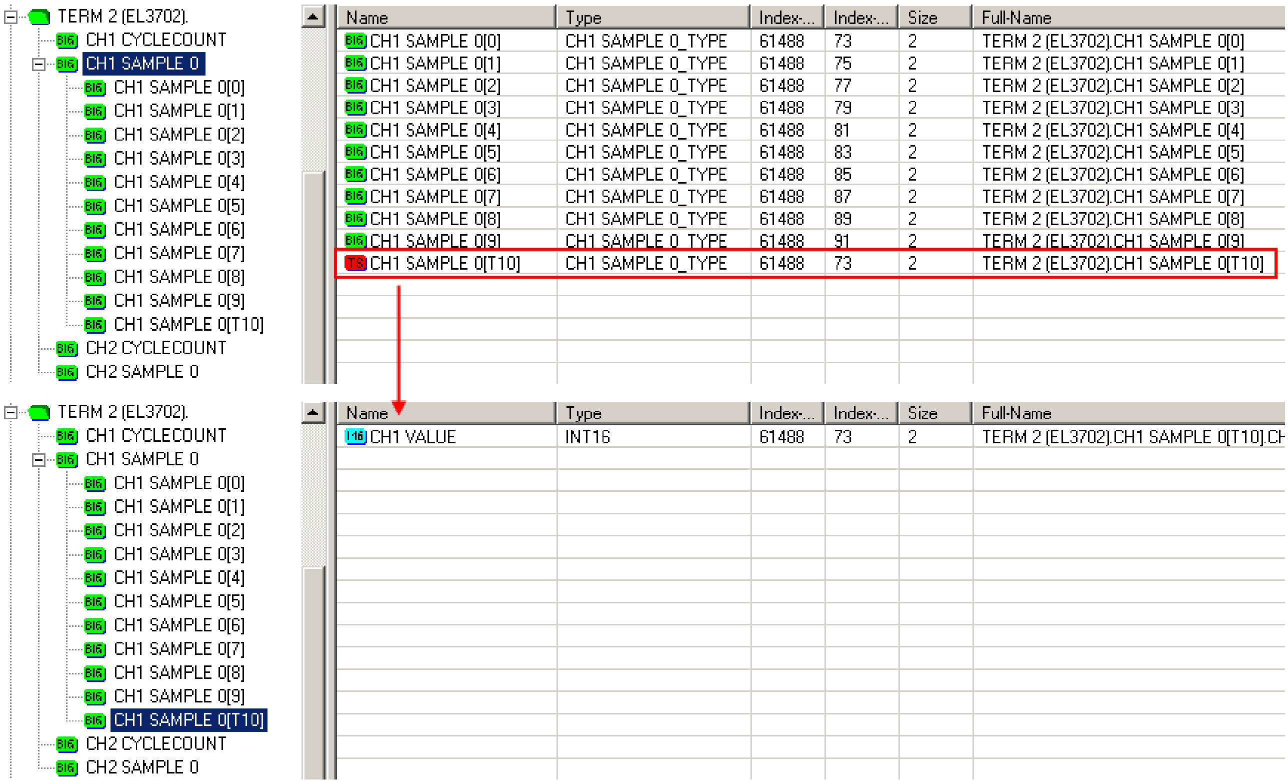
Also see Beckhoff Information System

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



- TwinCAT 2
  - Start Page
  - Notes on the documentation
  - TwinCAT Overview
  - TwinCAT Quick Start
  - TwinCAT System
  - TwinCAT System Manager
  - TwinCAT RemoteManager
  - TwinCAT R3IO
  - TX1200 | TwinCAT PLC
  - TwinCAT NC
  - TwinCAT Supplement
    - Overview
    - Installation on CE Devices
    - Category System TS1xxx
    - Category Measurement TS3xxx
      - Measurement
      - TS3300 | TwinCAT Scope2
        - Foreword
        - Scope 2 Handbook
        - Scope 2 Tutorial
        - ScopeViewControl
        - Annex
          - Oversampling record
          - Scope2 FAQ

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



- TERM 2 (EL3702).
  - CH1 CYCLECOUNT
  - CH1 SAMPLE 0
    - CH1 SAMPLE 0[0]
    - CH1 SAMPLE 0[1]
    - CH1 SAMPLE 0[2]
    - CH1 SAMPLE 0[3]
    - CH1 SAMPLE 0[4]
    - CH1 SAMPLE 0[5]
    - CH1 SAMPLE 0[6]
    - CH1 SAMPLE 0[7]
    - CH1 SAMPLE 0[8]
    - CH1 SAMPLE 0[9]
    - CH1 SAMPLE 0[T10]
  - CH2 CYCLECOUNT
  - CH2 SAMPLE 0

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

- TERM 2 (EL3702).
  - CH1 CYCLECOUNT
  - CH1 SAMPLE 0
    - CH1 SAMPLE 0[0]
    - CH1 SAMPLE 0[1]
    - CH1 SAMPLE 0[2]
    - CH1 SAMPLE 0[3]
    - CH1 SAMPLE 0[4]
    - CH1 SAMPLE 0[5]
    - CH1 SAMPLE 0[6]
    - CH1 SAMPLE 0[7]
    - CH1 SAMPLE 0[8]
    - CH1 SAMPLE 0[9]
    - CH1 SAMPLE 0[T10]
  - CH2 CYCLECOUNT
  - CH2 SAMPLE 0

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

Fig. 165: Automatically calculated array variable (red) in the Scope2

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

5.4.2 Commissioning

The data flow through of the terminal's channel is recorded below. The elements involved are described in detail in this chapter.

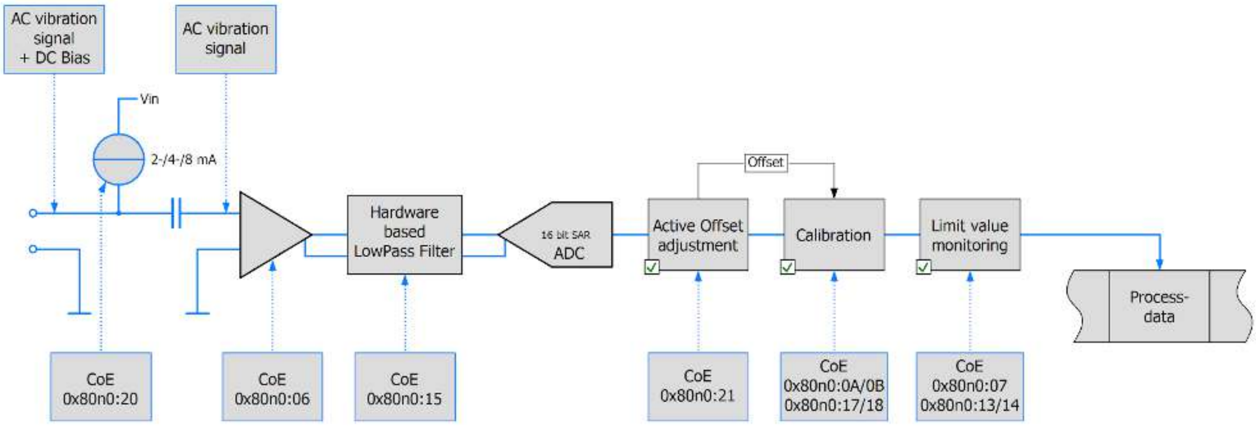


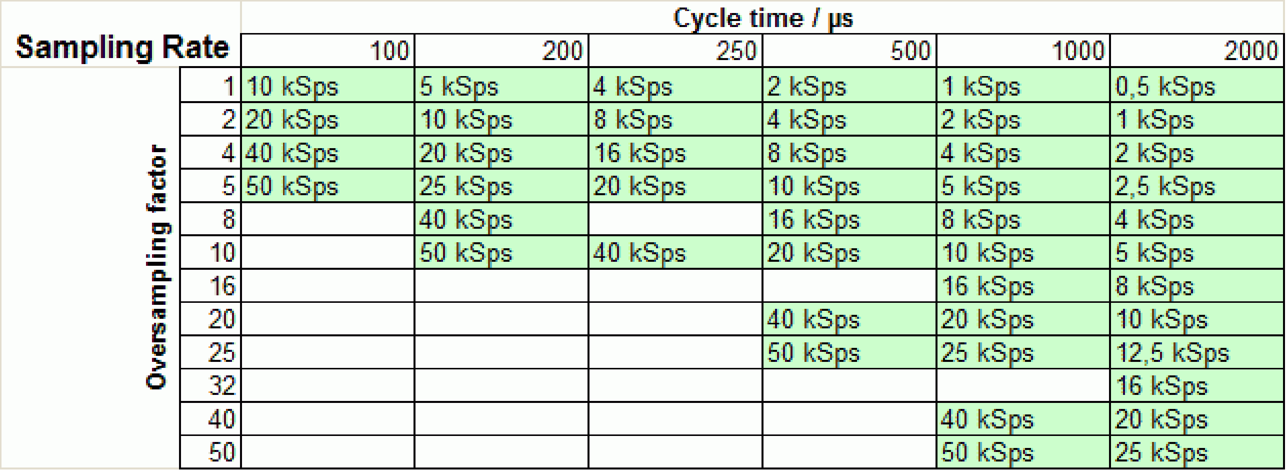
Fig. 166: Data flow of a channel

Note: The sampling rate in the ADC and the low-pass filter can be set independently of each other. For example, the filter can be set to half the sampling rate or to the next possible setting if required for the application.

Determination of the desired sampling rate

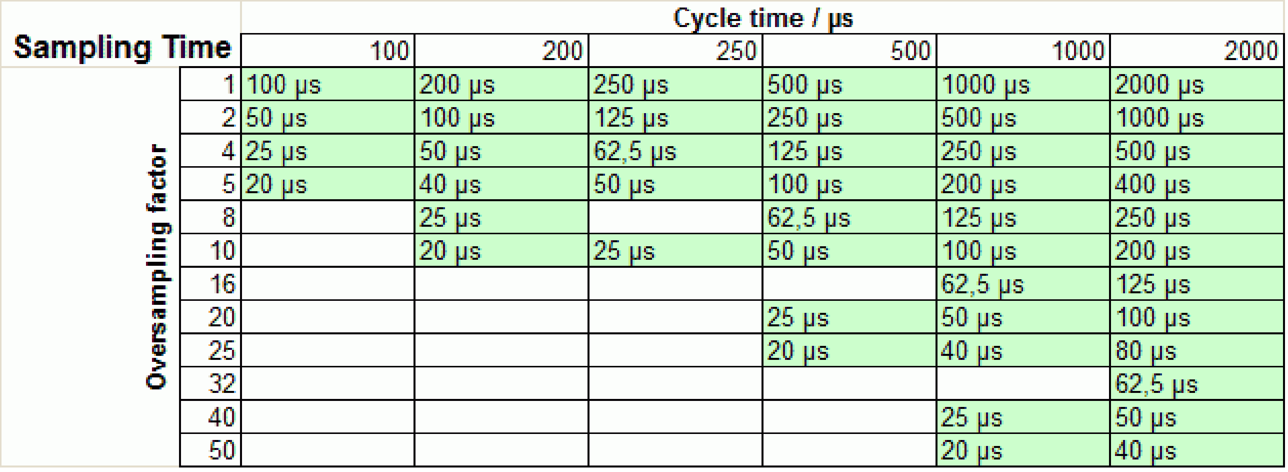
The necessary/desired sampling rate results from the selected cycle time and the set oversampling factor. Automatic setting of the terminal/box by the selection of the sampling rate alone is not possible.

Maximum values: 50-fold oversampling, 50 kSP/s, cycle time 10 ms



Sampling Rate		Cycle time / µs					
		100	200	250	500	1000	2000
Oversampling factor	1	10 kSps	5 kSps	4 kSps	2 kSps	1 kSps	0,5 kSps
	2	20 kSps	10 kSps	8 kSps	4 kSps	2 kSps	1 kSps
	4	40 kSps	20 kSps	16 kSps	8 kSps	4 kSps	2 kSps
	5	50 kSps	25 kSps	20 kSps	10 kSps	5 kSps	2,5 kSps
	8		40 kSps		16 kSps	8 kSps	4 kSps
	10		50 kSps	40 kSps	20 kSps	10 kSps	5 kSps
	16					16 kSps	8 kSps
	20				40 kSps	20 kSps	10 kSps
	25				50 kSps	25 kSps	12,5 kSps
	32						16 kSps
	40					40 kSps	20 kSps
	50					50 kSps	25 kSps

Fig. 167: Sampling rates in relation to cycle time and oversampling



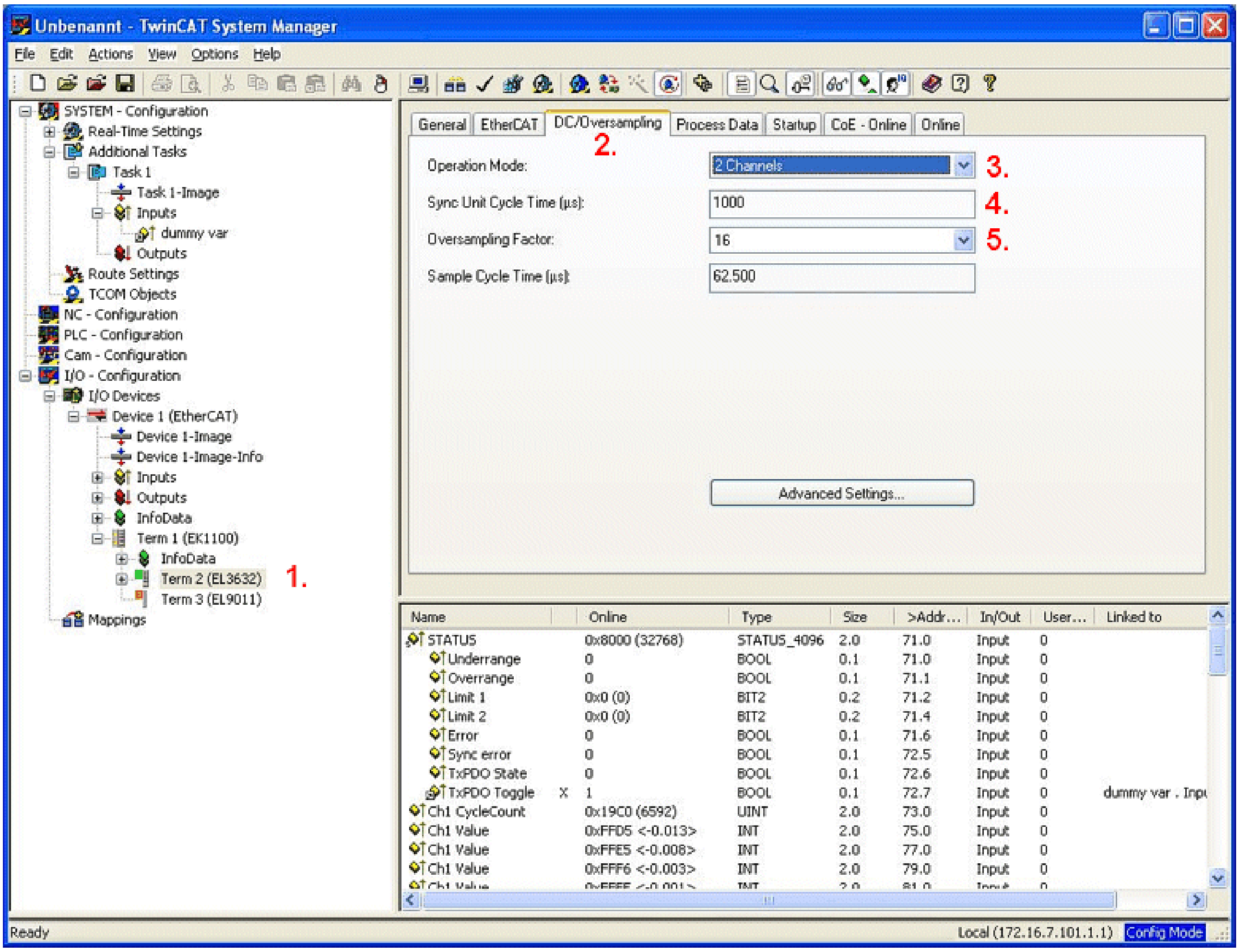
Sampling Time	Cycle time / µs 100	200	250	500	1000	2000
Oversampling factor 1	100 µs	200 µs	250 µs	500 µs	1000 µs	2000 µs
2	50 µs	100 µs	125 µs	250 µs	500 µs	1000 µs
4	25 µs	50 µs	62,5 µs	125 µs	250 µs	500 µs
5	20 µs	40 µs	50 µs	100 µs	200 µs	400 µs
8		25 µs		62,5 µs	125 µs	250 µs
10		20 µs	25 µs	50 µs	100 µs	200 µs
16					62,5 µs	125 µs
20				25 µs	50 µs	100 µs
25				20 µs	40 µs	80 µs
32						62,5 µs
40					25 µs	50 µs
50					20 µs	40 µs

Fig. 168: Sampling times in relation to cycle times and oversampling

Configurations that demand sampling times not divisible by 500 ns are not supported.

Setting the sampling rate

1. Select the terminal/box in the TwinCAT tree
2. Select the “DC/Oversampling” tab
3. Select the operating mode (1/2-channel)
4. “Sync Unit Cycle Time” is indicated; on the basis of the above table…
5. Select the oversampling factor. The “Sample Cycle Time (µs)” indicates the reciprocal value of the sampling rate. The SM automatically activates all process data entries thereafter.



Operation Mode: 2 Channels

Sync Unit Cycle Time (µs): 1000

Oversampling Factor: 16

Sample Cycle Time (µs): 62.500

Name	Online	Type	Size	>Addr...	In/Out	User...	Linked to
STATUS	0x8000 (32768)	STATUS_4096	2.0	71.0	Input	0	
Underrange	0	BOOL	0.1	71.0	Input	0	
Overrange	0	BOOL	0.1	71.1	Input	0	
Limit 1	0x0 (0)	BIT2	0.2	71.2	Input	0	
Limit 2	0x0 (0)	BIT2	0.2	71.4	Input	0	
Error	0	BOOL	0.1	71.6	Input	0	
Sync error	0	BOOL	0.1	72.5	Input	0	
TxPDO State	0	BOOL	0.1	72.6	Input	0	
TxPDO Toggle X	1	BOOL	0.1	72.7	Input	0	dummy var . Inpu
Ch1 CycleCount	0x19C0 (6592)	UINT	2.0	73.0	Input	0	
Ch1 Value	0xFFD5 <-0.013>	INT	2.0	75.0	Input	0	
Ch1 Value	0xFFE5 <-0.008>	INT	2.0	77.0	Input	0	
Ch1 Value	0xFFF6 <-0.003>	INT	2.0	79.0	Input	0	
Ch1 Value	0xFFFF <-0.001>	INT	2.0	81.0	Input	0	

Fig. 169: Setting the sampling rate in TwinCAT

Loading the configuration data (ESI) from the terminal

If the online description is used, the DC/Oversampling dialog is not displayed in the TwinCAT System Manager.
To use the dialog, copy the XML Description into the designated TwinCAT/Io/EtherCAT folder before the TwinCAT System Manager is started.

Application with external masters

The oversampling function can also be activated manually: The oversampling factor should be specified depending on the required sampling rate and cycle time.

For each required channel the status word and the corresponding number of samples have to be entered in object 0x1C13. Activate PDOs "Next Sync1 Time" and/or "Sample Counter", if necessary. To this end, initially set subindex 0 to 0 and at the end to the number of entered values.

The sync interrupts should be parameterized as follows: Sync0: CycleTime/Oversampling Factor, set Enable; Sync1 Cycle Unit Cycle, set Enable.

The master must support Distributed Clocks.

Selection of the process data

No longer necessary with TwinCAT.

Filter

Each channel has a parameterizable 5th order filter with Butterworth characteristic with upstream and downstream anti-aliasing filters that are parameterized automatically. The whole filter stage is based on hardware. There are no software filters (except for the active offset setting described below). When the limit frequency is set to 10 Hz (0), an additional amplification factor 20 is activated automatically.

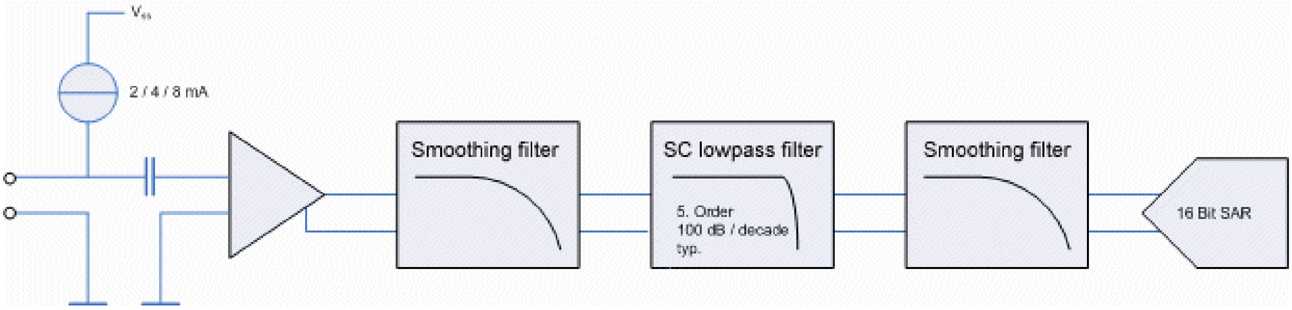


Fig. 170: Filter structure

The analog input filters can be set via the CoE objects 0x80n0:15 [▶ 151] (channel 1/2). It is not possible to switch off the filters.

The characteristic of the filter stage is adjustable:
0: 10 Hz, Gain 20
1: 100 Hz
2: 500 Hz
3: 1000 Hz
4: 5000 Hz
5: 10000 Hz
6: 25000 Hz
7: 2000 Hz (from firmware [▶ 203] 11)

Setting the filters

The filters must be configured separately for both channels.
Setting the filters will interrupt the conversion of new data for a short time.

Active offset setting

The function "Active offset adjustment" calculates the long-term average of the values. The calculated offset value is used instead of the set "user" and "vendor calibration offset" entries. At least one function, "user" or "vendor calibration", must be activated

The calculation of the average value is configurable:
Level 1: b = 1/4096
Level 2: b = 1/8192
Level 3: b = 1/16384
Level 4: b = 1/32768
Level 5: b = 1/65536
Level 6: b = 1/131072 (128 k)
Level 7: b = 1/262144 (256 k)
Level 8: b = 1/524288 (512 k).

Calibration

The input values can be calibrated by means of manufacturer or user values:

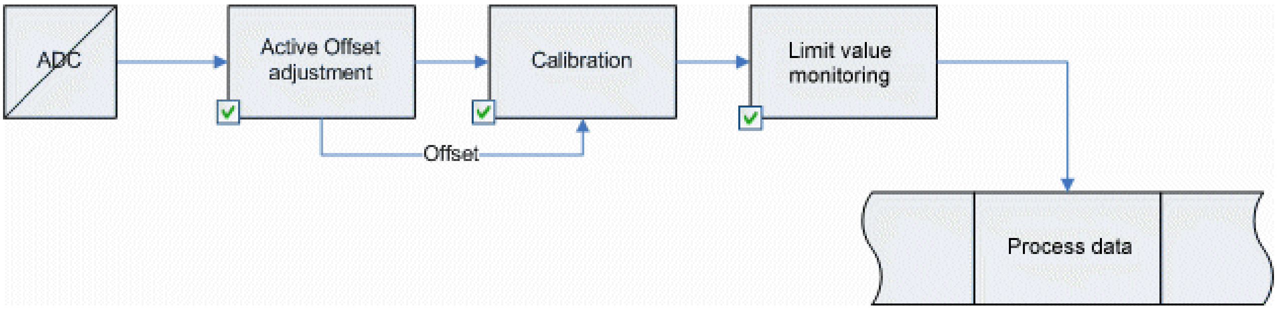


Fig. 5: Data flow

Vendor calibration, index 0x80n0:0B [▸ 151]

The vendor calibration is enabled via index 0x80n0:0B [▸ 151]. Parameterization takes place via the indices

- 0x80nF:01 [▸ 151] offset (manufacturer compensation)
- 0x80nF:02 [▸ 151] gain (manufacturer compensation)

$Y_H = (X_{ADC} - B_K) * A_K$	Measured value following manufacturer calibration (corresponds to X_{ADC} if index 0x80n0:0B [▸ 151] is inactive)

User calibration, index 0x80n0:0A [▸ 151]

The user calibration is enabled via index 0x80n0:0A [▸ 151]. Parameterization takes place via the indices

- 0x80n0:17 [▸ 151] User calibration offset
- 0x80n0:18 [▸ 151] User gain compensation

$Y_A = (Y_H - B_W) * A_W$	Measured value following user calibration (corresponds to Y_H if index 0x80n0:0A [▸ 151] is inactive)

Active offset adjustment

If the function "Active offset adjustment" is active the offset values are not used. Instead, a dynamically calculated offset is subtracted.

Example interpretation

Sample: A sensor with a sensitivity S of 100 mV / g (10.2 mV/(m/s^2)) is connected to a synchronized EL3632/EP3632 (15-bit resolution + sign, +/- 5 V). In the process data an amplitude of 1507 is measured.

a = Y_A * 5 V / (2^15 * S) Conversion of process data value Y_A to acceleration a.
a = 1507 * 5 V / (2^15 * 0.1 V/*g*)
a = 2.3 *g*
a = 2.3 *g* * 9.81 (m/s^2) / *g*
a = 22.5 m/s^2

Y_A = 2^15 / 5 V * S * a Conversion of acceleration g to process data value Y_A.
Y_A = 2^15 / 5 V * 0.1 V/g * 2.3 *g*
Y_A = 1507

Sensor connection

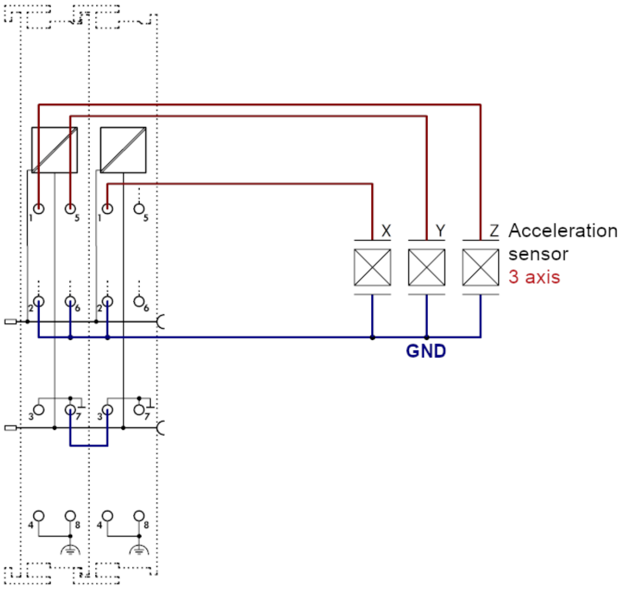
The supply current for the sensors is configurable. With 8 mA the vertical (standard) installation position of the terminal must be ensured. The smallest possible supply current should be set, depending on the sensor and cable length.

After switching on the 24 V supply voltage or connecting the sensor, a leakage current forms due to the input capacity on the high-pass filter. This current is based on the physical properties of electrolytic capacitors and is technically impossible to prevent. This current stabilizes at a constant value within a few minutes, and during the measurement it generates a constant offset of typically a few mV within the specified tolerance range. If this offset should prove to be disturbing when analyzing the measurement, it can be permanently and automatically subtracted out by activating the "Active offset adjustment" (object 0x80n0:21 [▶ 151]).

A shielded (simple or multiple) sensor cable must be used. The shield should be connected directly at the shield connections of the terminal.

The red error LED comes on and the error bit is set in the event of an open circuit or if no sensor is connected. If only the first channel is activated, the red LED for the second channel is disabled in SAFEOP and OP state.

Multi-channel IEPE sensors with a common GND can be connected to the EL3632 if the GND and AGND connection points are connected via an external bridge:



Unused inputs

Unused inputs must not be short-circuited.

Measuring error

Measuring error < ±0,5% (DC; relative to full scale value), taking into account the Butterworth characteristic.

5.4.2.1 CoE object description

5.4.2.1.1 Profile-specific and parameterization objects

EtherCAT XML Device Description

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

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab [▸ 106] (double-click on the respective object) or via the Process Data tab [▸ 103](allocation of PDOs). Please note the following general CoE notes [▸ 30] 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

Restore object

Index 1011 Restore default parameters

Index (hex)	Name	Meaning	Data type	Flags	Default
1011:0	Restore default parameters [▸ 215]	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})

Configuration data

Index 80n0 AI settings (for n = 0: channel 1; n = 1: channel 2)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	AI Settings	Maximum subindex	UINT8	RO	0x20 (32$_{\text{dec}}$)
80n0:07	Enable limit	Activate limit evaluation (deviates from other EL terminals!)	BOOLEAN	RW	0x0 (FALSE)
80n0:0A	Enable user calibration	Enabling of the user calibration	BOOLEAN	RW	0x0 (FALSE
80n0:0B	Enable vendor calibration	Enabling of the vendor calibration	BOOLEAN	RW	0x1 (TRUE)
80n0:13	Limit 1	Upper limit value	INT16	RW	0x0000 (0$_{\text{dec}}$)
80n0:14	Limit 2	Lower limit value	INT16	RW	0x0000 (0$_{\text{dec}}$)
80n0:15	Filter settings	Filter [▶ 146] settings 0: 10 Hz, Gain 20 1: 100 Hz 2: 500 Hz 3: 1000 Hz 4: 5000 Hz 5: 10000 Hz 6: 25000 Hz 7: 2000 Hz (from firmware [▶ 203] 11)	ENUM	RW	10000 Hz (5)
80n0:17	User calibration offset	User calibration offset	INT16	RW	0
80n0:18	User calibration gain	User calibration gain factor	INT16	RW	16384
80n0:20	Supply current	Sensor current setting 0: 2 mA 1: 4 mA 2: 8 mA	ENUM	RW	2 mA (0)
80n0:21	Active offset adjustment	Automatic offset calculation [▶ 147] 0: Disabled 1: Level 1 2: Level 2 3: Level 3 4: Level 4 5: Level 5 6: Level 6 7: Level 7 8: Level 8	ENUM	RW	Disabled (0)

Index 80n8 AI advanced settings (for n = 0: channel 1; n = 1: channel 2)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n8:0	AI Advanced Settings	Maximum subindex	UINT8	RO	0x06 (6$_{\text{dec}}$)
80n8:06	Input Amplifier	permitted values:	BIT2	RW	0x00 (0$_{\text{dec}}$)
		0: Preset (setting via object 0x80n0:15 [▶ 151])			
		1: ON (switch-on of the analog amplifier x20 irrespective of filter settings; gain factor of 20 is not changeable)			
		2: OFF (switch-off of the analog amplifier irrespective of the filter settings)			

Index 80nF AI Vendor data (for n = 0: channel 1; n = 1: channel 2)

Index (hex)	Name	Meaning	Data type	Flags	Default
80nF:0	AI vendor data	Maximum subindex	UINT8	RO	0x04 (4$_{\text{dec}}$)
80nF:01	Calibration offset (gain 1)	Offset (vendor calibration), gain 1	INT16	RW	0x0000 (0$_{\text{dec}}$)
80nF:02	Calibration gain (gain 1)	Gain (vendor calibration), gain 1	INT16	RW	0x0000 (0$_{\text{dec}}$)
80nF:03	Calibration offset (gain 20)	Offset (vendor calibration), gain 20	INT16	RW	0x0000 (0$_{\text{dec}}$)
80nF:04	Calibration gain (gain 20)	Gain (vendor calibration), gain 20	INT16	RW	0x0000 (0$_{\text{dec}}$)

Command object

Index FB00 command

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	DCM Command	Command interface	UINT8	RO	0x03 (3_{dec})
FB00:01	Request	0x8000: Software reset, hardware is re-initialized with the current CoE configuration (this otherwise happens only during the transition to INIT)	OCTET-STRING[2]	RW	{0}
FB00:02	Status	0x8000: 0x01 if a reset has taken place	UINT8	RO	0x00 (0_{dec})
FB00:03	Response	0x8000: not used	OCTET-STRING[4]	RO	{0}

Input data

Index 60n0 Status (for n = 0: channel 1; n = 1: channel 2)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	SAI Inputs	Maximum subindex	UINT8	RO	0x10 (16_{dec})
60n0:01	Underrange	Indicates that the electrical measuring range is undershot	BOOLEAN	RO	0x00 (0_{dec})
60n0:02	Overrange	Indicates that the electrical measuring range is exceeded	BOOLEAN	RO	0x00 (0_{dec})
60n0:03	Limit 1	1: one or more values smaller than or equal to Limit 2 2: one or more values greater than or equal to Limit 1 3: 1 and 2 both true	BIT2	RO	0x00 (0_{dec})
60n0:07	Error	Measuring error	BOOLEAN	RO	0x00 (0_{dec})
60n0:0E	Sync error	Synchronization error	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	A new measured value is available (if the toggle bit was changed). Status bits may be changed independent from the toggle bit.	BOOLEAN	RO	0x00 (0_{dec})

Index 60n1 Samples (for n = 0: channel 1; n = 1: channel 2)

Index (hex)	Name	Meaning	Data type	Flags	Default
60n1:0	Samples	Maximum subindex	UINT8	RO	0x32 (50_{dec})
60n1:01	Subindex 001	1. Sample	UINT16	RO P	0x0000 (0_{dec})
...	...	...	...	...	...
60n1:32	Subindex 050	50. Sample	UINT16	RO P	0x0000 (0_{dec})

Index 6020 Next Sync 1 Time

Index (hex)	Name	Meaning	Data type	Flags	Default
6020:0	Next Sync 1 Time	Maximum subindex	UINT8	RO	0x01 (1_{dec})
6020:01	Start time next latch	DC timestamp of the next data set	UINT64	RO P	0x00 00 00 00 00 00 00 00 (0_{dec})

Index 6021 Sample Count

Index (hex)	Name	Meaning	Data type	Flags	Default
6021:0	Sample Count	Maximum subindex	UINT8	RO	0x01 (1_{dec})
6021:01	Sample Count	Sample counter (incremented with each ADC value)	UINT16	RO P	0x0000 (0_{dec})

Information / diagnosis data (device-specific)

Index 10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Maximum subindex	UINT8	RO	0x15 (21$_{dec}$)
10F3:01	Maximum Messages	Maximum number of stored messages A maximum of 16 messages can be stored	UINT8	RO	0x00 (0$_{dec}$)
10F3:02	Newest Message	Subindex of the latest message	UINT8	RO	0x00 (0$_{dec}$)
10F3:03	Newest Acknowledged Message	Subindex of the last confirmed message	UINT8	RW	0x00 (0$_{dec}$)
10F3:04	New Messages Available	Indicates that a new message is available	BOOLEAN	RO	0x00 (0$_{dec}$)
10F3:05	Flags	not used	UINT16	RW	0x0000 (0$_{dec}$)
10F3:06	Diagnosis Message 001	Message 1	OCTET-STRING[20]	RO	{0}
...	...	..	...	...	...
10F3:15	Diagnosis Message 016	Message 16	OCTET-STRING[20]	RO	{0}

Index 10F8 Actual Time Stamp

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

5.4.2.1.2 Standard objects and PDO mapping

EtherCAT XML Device Description

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

Parameterization via the CoE list (CAN over EtherCAT)

The EtherCAT device is parameterized via the CoE-Online tab [▶ 106] (double-click on the respective object) or via the Process Data tab [▶ 103](allocation of PDOs). Please note the following general CoE notes [▶ 30] 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

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	0x012C1389 (19665801$_{dec}$)

Index 1008 Device name

Index (hex)	Name	Meaning	Data type	Flags	Default
1008:0	Device name	Device name of the EtherCAT slave	STRING	RO	EL3632

Index 1009 Hardware version

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

Index 100A Software version

Index (hex)	Name	Meaning	Data type	Flags	Default
100A:0	Software version	Firmware version of the EtherCAT slave	STRING	RO	05

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	0x0E303052 (238039122_{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	0x00110000 (1114112_{dec})
1018:04	Serial number	Serial number of the EtherCAT slave; the low byte (bit 0-7) of the low word contains the year of production, the high byte (bit 8-15) of the low word contains the week of production, the high word (bit 16-31) is 0	UINT32	RO	0x00000000 (0_{dec})

Index 10F0 Backup parameter handling

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

Index 1A00 Analog Input TxPDO-MapStatus Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A00:0	Analog Input TxPDO-MapStatus Ch.1	PDO Mapping TxPDO 1	UINT8	RO	0x09 (9_{dec})
1A00:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (AI Inputs), entry 0x01 (Underrange))	UINT32	RO	0x6000:01, 1
1A00:02	SubIndex 002	2. PDO Mapping entry (object 0x6000 (AI Inputs), entry 0x02 (Overrange))	UINT32	RO	0x6000:02, 1
1A00:03	SubIndex 003	3. PDO Mapping entry (object 0x6000 (AI Inputs), entry 0x03 (Limit 1))	UINT32	RO	0x6000:03, 2
1A00:04	SubIndex 004	4. PDO Mapping entry (object 0x6000 (AI Inputs), entry 0x05 (Limit 2))	UINT32	RO	0x0000:00, 2
1A00:05	SubIndex 005	5. PDO Mapping entry (2 bits align)	UINT32	RO	0x6000:07, 1
1A00:06	SubIndex 006	6. PDO Mapping entry (6 bits align)	UINT32	RO	0x0000:00, 6
1A00:07	SubIndex 007	7. PDO Mapping entry (object 0x1800 (AI TxPDO-Par Standard Ch.1), entry 0x07 (TxPDO State))	UINT32	RO	0x1C33:20, 1
1A00:08	SubIndex 008	8. PDO Mapping entry (object 0x1800, entry 0x09)	UINT32	RO	0x1800:07, 1
1A00:09	SubIndex 009	9. PDO Mapping entry (object 0x6000 (AI Inputs), entry 0x11 (Value))	UINT32	RO	0x1800:09, 1

Index 1A01 Analog Input TxPDO-MapSamples 1 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A01:0	Analog Input TxPDO-MapSamples 1 Ch.1	PDO Mapping TxPDO 2	UINT8	RO	0x01 (1_{dec})
1A01:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (AI Inputs), entry 0x11 (Value))	UINT32	RO	0x6001:01, 16

Index 1A02 Analog Input TxPDO-MapSamples 2 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A02:0	Analog Input TxPDO-MapSamples 2 Ch.1	PDO Mapping TxPDO 3	UINT8	RO	0x01 (1_{dec})
1A02:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Sample 2), entry 0x01 (Value))	UINT32	RO	0x6001:02, 16

Index 1A03 Analog Input TxPDO-MapSamples 3 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A03:0	Analog Input TxPDO-MapSamples 3 Ch.1	PDO Mapping TxPDO 4	UINT8	RO	0x01 (1_{dec})
1A03:01	SubIndex 001	1. PDO Mapping entry (object 0x6003 (Sample 3), entry 0x01 (Value))	UINT32	RO	0x6001:03, 16

Index 1A04 Analog Input TxPDO-MapSamples 4 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A04:0	Analog Input TxPDO-MapSamples 4 Ch.1	PDO Mapping TxPDO 5	UINT8	RO	0x01 (1_{dec})
1A04:01	SubIndex 001	1. PDO Mapping entry (object 0x6004 (Sample 4), entry 0x01 (Value))	UINT32	RO	0x6001:04, 16

Index 1A05 Analog Input TxPDO-MapSamples 5 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A05:0	Analog Input TxPDO-MapSamples 5 Ch.1	PDO Mapping TxPDO 6	UINT8	RO	0x01 (1_{dec})
1A05:01	SubIndex 001	1. PDO Mapping entry (object 0x6005 (Sample 5), entry 0x01 (Value))	UINT32	RO	0x6001:05, 16

Index 1A06 Analog Input TxPDO-MapSamples 6 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A06:0	Analog Input TxPDO-MapSamples 6 Ch.1	PDO Mapping TxPDO 7	UINT8	RO	0x01 (1_{dec})
1A06:01	SubIndex 001	1. PDO Mapping entry (object 0x6006 (Sample 6), entry 0x01 (Value))	UINT32	RO	0x6001:06, 16

Index 1A07 Analog Input TxPDO-MapSamples 7 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A07:0	Analog Input TxPDO-MapSamples 7 Ch.1	PDO Mapping TxPDO 8	UINT8	RO	0x01 (1_{dec})
1A07:01	SubIndex 001	1. PDO Mapping entry (object 0x6007 (Sample 7), entry 0x01 (Value))	UINT32	RO	0x6001:07, 16

Index 1A08 Analog Input TxPDO-MapSamples 8 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A08:0	Analog Input TxPDO-MapSamples 8 Ch.1	PDO Mapping TxPDO 9	UINT8	RO	0x01 (1_{dec})
1A08:01	SubIndex 001	1. PDO Mapping entry (object 0x6008 (Sample 8), entry 0x01 (Value))	UINT32	RO	0x6001:08, 16

Index 1A09 Analog Input TxPDO-MapSamples 9 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A09:0	Analog Input TxPDO-MapSamples 9 Ch.1	PDO Mapping TxPDO 10	UINT8	RO	0x01 (1_{dec})
1A09:01	SubIndex 001	1. PDO Mapping entry (object 0x6009 (Sample 9), entry 0x01 (Value))	UINT32	RO	0x6001:09, 16

Index 1A0A Analog Input TxPDO-MapSamples 10 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0A:0	Analog Input TxPDO-MapSamples 10 Ch.1	PDO Mapping TxPDO 11	UINT8	RO	0x01 (1_{dec})
1A0A:01	SubIndex 001	1. PDO Mapping entry (object 0x600A (Sample 10), entry 0x01 (Value))	UINT32	RO	0x6001:0A, 16

Index 1A0B Analog Input TxPDO-MapSamples 11 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0B:0	Analog Input TxPDO-MapSamples 11 Ch.1	PDO Mapping TxPDO 12	UINT8	RO	0x01 (1_{dec})
1A0B:01	SubIndex 001	1. PDO Mapping entry (object 0x600B (Sample 11), entry 0x01 (Value))	UINT32	RO	0x6001:0B, 16

Index 1A0C Analog Input TxPDO-MapSamples 12 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0C:0	Analog Input TxPDO-MapSamples 12 Ch.1	PDO Mapping TxPDO 13	UINT8	RO	0x01 (1_{dec})
1A0C:01	SubIndex 001	1. PDO Mapping entry (object 0x600C (Sample 12), entry 0x01 (Value))	UINT32	RO	0x6001:0C, 16

Index 1A0D Analog Input TxPDO-MapSamples 13 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0D:0	Analog Input TxPDO-MapSamples 13 Ch.1	PDO Mapping TxPDO 14	UINT8	RO	0x01 (1_{dec})
1A0D:01	SubIndex 001	1. PDO Mapping entry (object 0x600D (Sample 13), entry 0x01 (Value))	UINT32	RO	0x6001:0D, 16

Index 1A0E Analog Input TxPDO-MapSamples 14 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0E:0	Analog Input TxPDO-MapSamples 14 Ch.1	PDO Mapping TxPDO 15	UINT8	RO	0x01 (1_{dec})
1A0E:01	SubIndex 001	1. PDO Mapping entry (object 0x600E (Sample 14), entry 0x01 (Value))	UINT32	RO	0x6001:0E, 16

Index 1A0F Analog Input TxPDO-MapSamples 15 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A0F:0	Analog Input TxPDO-MapSamples 15 Ch.1	PDO Mapping TxPDO 16	UINT8	RO	0x01 (1_{dec})
1A0F:01	SubIndex 001	1. PDO Mapping entry (object 0x600F (Sample 15), entry 0x01 (Value))	UINT32	RO	0x6001:0F, 16

Index 1A10 Analog Input TxPDO-MapSamples 16 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A10:0	Analog Input TxPDO-MapSamples 16 Ch.1	PDO Mapping TxPDO 17	UINT8	RO	0x01 (1_{dec})
1A10:01	SubIndex 001	1. PDO Mapping entry (object 0x6000 (Sample 0), entry 0x01 (Underrange))	UINT32	RO	0x6001:10, 16

Index 1A11 Analog Input TxPDO-MapSamples 17 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A11:0	Analog Input TxPDO-MapSamples 17 Ch.1	PDO Mapping TxPDO 18	UINT8	RO	0x01 (1_{dec})
1A11:01	SubIndex 001	1. PDO Mapping entry (object 0x6001 (Sample 1), entry 0x01 (Value))	UINT32	RO	0x6001:11, 16

Index 1A12 Analog Input TxPDO-MapSamples 18 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A12:0	Analog Input TxPDO-MapSamples 18 Ch.1	PDO Mapping TxPDO 19	UINT8	RO	0x01 (1_{dec})
1A12:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Sample 2), entry 0x01 (Value))	UINT32	RO	0x6001:12, 16

Index 1A13 Analog Input TxPDO-MapSamples 19 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A13:0	Analog Input TxPDO-MapSamples 19 Ch.1	PDO Mapping TxPDO 20	UINT8	RO	0x01 (1_{dec})
1A13:01	SubIndex 001	1. PDO Mapping entry (object 0x6003 (Sample 3), entry 0x01 (Value))	UINT32	RO	0x6001:13, 16

Index 1A14 Analog Input TxPDO-MapSamples 20 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A14:0	Analog Input TxPDO-MapSamples 20 Ch.1	PDO Mapping TxPDO 21	UINT8	RO	0x01 (1_{dec})
1A14:01	SubIndex 001	1. PDO Mapping entry (object 0x6010 (Sample 16), entry 0x01 (Value))	UINT32	RO	0x6001:14, 16

Index 1A15 Analog Input TxPDO-MapSamples 21 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A15:0	Analog Input TxPDO-MapSamples 21 Ch.1	PDO Mapping TxPDO 22	UINT8	RO	0x01 (1_{dec})
1A15:01	SubIndex 001	1. PDO Mapping entry (object 0x6011 (Sample 17), entry 0x01 (Value))	UINT32	RO	0x6001:15, 16

Index 1A16 Analog Input TxPDO-MapSamples 22 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A16:0	Analog Input TxPDO-MapSamples 22 Ch.1	PDO Mapping TxPDO 23	UINT8	RO	0x01 (1_{dec})
1A16:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x15 ())	UINT32	RO	0x6001:16, 16

Index 1A17 Analog Input TxPDO-MapSamples 23 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A17:0	Analog Input TxPDO-MapSamples 23 Ch.1	PDO Mapping TxPDO 24	UINT8	RO	0x01 (1_{dec})
1A17:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x16 ())	UINT32	RO	0x6001:17, 16

Index 1A18 Analog Input TxPDO-MapSamples 24 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A18:0	Analog Input TxPDO-MapSamples 24 Ch.1	PDO Mapping TxPDO 25	UINT8	RO	0x01 (1_{dec})
1A18:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x17 ())	UINT32	RO	0x6001:18, 16

Index 1A19 Analog Input TxPDO-MapSamples 25 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A19:0	Analog Input TxPDO-MapSamples 25 Ch.1	PDO Mapping TxPDO 26	UINT8	RO	0x01 (1_{dec})
1A19:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x18 ())	UINT32	RO	0x6001:19, 16

Index 1A1A Analog Input TxPDO-MapSamples 26 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A1A:0	Analog Input TxPDO-MapSamples 26 Ch.1	PDO Mapping TxPDO 27	UINT8	RO	0x01 (1_{dec})
1A1A:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x19 ())	UINT32	RO	0x6001:1A, 16

Index 1A1B Analog Input TxPDO-MapSamples 27 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A1B:0	Analog Input TxPDO-MapSamples 27 Ch.1	PDO Mapping TxPDO 28	UINT8	RO	0x01 (1_{dec})
1A1B:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x1A ())	UINT32	RO	0x6001:1B, 16

Index 1A1C Analog Input TxPDO-MapSamples 28 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A1C:0	Analog Input TxPDO-MapSamples 28 Ch.1	PDO Mapping TxPDO 29	UINT8	RO	0x01 (1_{dec})
1A1C:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x1B ())	UINT32	RO	0x6001:1C, 16

Index 1A1D Analog Input TxPDO-MapSamples 29 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A1D:0	Analog Input TxPDO-MapSamples 29 Ch.1	PDO Mapping TxPDO 30	UINT8	RO	0x01 (1_{dec})
1A1D:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x1C ())	UINT32	RO	0x6001:1D, 16

Index 1A1E Analog Input TxPDO-MapSamples 30 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A1E:0	Analog Input TxPDO-MapSamples 30 Ch.1	PDO Mapping TxPDO 31	UINT8	RO	0x01 (1_{dec})
1A1E:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x1D ())	UINT32	RO	0x6001:1E, 16

Index 1A1F Analog Input TxPDO-MapSamples 31 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A1F:0	Analog Input TxPDO-MapSamples 31 Ch.1	PDO Mapping TxPDO 32	UINT8	RO	0x01 (1_{dec})
1A1F:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x1E ())	UINT32	RO	0x6001:1F, 16

Index 1A20 Analog Input TxPDO-MapSamples 32 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A20:0	Analog Input TxPDO-MapSamples 32 Ch.1	PDO Mapping TxPDO 33	UINT8	RO	0x01 (1_{dec})
1A20:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x1F ())	UINT32	RO	0x6001:20, 16

Index 1A21 Analog Input TxPDO-MapSamples 33 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A21:0	Analog Input TxPDO-MapSamples 33 Ch.1	PDO Mapping TxPDO 34	UINT8	RO	0x01 (1_{dec})
1A21:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x20 ())	UINT32	RO	0x6001:21, 16

Index 1A22 Analog Input TxPDO-MapSamples 34 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A22:0	Analog Input TxPDO-MapSamples 34 Ch.1	PDO Mapping TxPDO 35	UINT8	RO	0x01 (1_{dec})
1A22:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x21 ())	UINT32	RO	0x6001:22, 16

Index 1A23 Analog Input TxPDO-MapSamples 35 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A23:0	Analog Input TxPDO-MapSamples 35 Ch.1	PDO Mapping TxPDO 36	UINT8	RO	0x01 (1_{dec})
1A23:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x22 ())	UINT32	RO	0x6001:23, 16

Index 1A24 Analog Input TxPDO-MapSamples 36 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A24:0	Analog Input TxPDO-MapSamples 36 Ch.1	PDO Mapping TxPDO 37	UINT8	RO	0x01 (1_{dec})
1A24:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x23 ())	UINT32	RO	0x6001:24, 16

Index 1A25 Analog Input TxPDO-MapSamples 37 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A25:0	Analog Input TxPDO-MapSamples 37 Ch.1	PDO Mapping TxPDO 38	UINT8	RO	0x01 (1_{dec})
1A25:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x24 ())	UINT32	RO	0x6001:25, 16

Index 1A26 Analog Input TxPDO-MapSamples 38 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A26:0	Analog Input TxPDO-MapSamples 38 Ch.1	PDO Mapping TxPDO 39	UINT8	RO	0x01 (1_{dec})
1A26:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x25 ())	UINT32	RO	0x6001:26, 16

Index 1A27 Analog Input TxPDO-MapSamples 39 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A27:0	Analog Input TxPDO-MapSamples 39 Ch.1	PDO Mapping TxPDO 40	UINT8	RO	0x01 (1_{dec})
1A27:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x26 ())	UINT32	RO	0x6001:27, 16

Index 1A28 Analog Input TxPDO-MapSamples 40 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A28:0	Analog Input TxPDO-MapSamples 40 Ch.1	PDO Mapping TxPDO 41	UINT8	RO	0x01 (1_{dec})
1A28:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x27 ())	UINT32	RO	0x6001:28, 16

Index 1A29 Analog Input TxPDO-MapSamples 41 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A29:0	Analog Input TxPDO-MapSamples 41 Ch.1	PDO Mapping TxPDO 42	UINT8	RO	0x01 (1_{dec})
1A29:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x28 ())	UINT32	RO	0x6001:29, 16

Index 1A2A Analog Input TxPDO-MapSamples 42 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A2A:0	Analog Input TxPDO-MapSamples 42 Ch.1	PDO Mapping TxPDO 43	UINT8	RO	0x01 (1_{dec})
1A2A:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x29 ())	UINT32	RO	0x6001:2A, 16

Index 1A2B Analog Input TxPDO-MapSamples 43 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A2B:0	Analog Input TxPDO-MapSamples 43 Ch.1	PDO Mapping TxPDO 44	UINT8	RO	0x01 (1_{dec})
1A2B:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x2A ())	UINT32	RO	0x6001:2B, 16

Index 1A2C Analog Input TxPDO-MapSamples 44 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A2C:0	Analog Input TxPDO-MapSamples 44 Ch.1	PDO Mapping TxPDO 45	UINT8	RO	0x01 (1_{dec})
1A2C:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x2B ())	UINT32	RO	0x6001:2C, 16

Index 1A2D Analog Input TxPDO-MapSamples 45 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A2D:0	Analog Input TxPDO-MapSamples 45 Ch.1	PDO Mapping TxPDO 46	UINT8	RO	0x01 (1_{dec})
1A2D:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x2C ())	UINT32	RO	0x6001:2D, 16

Index 1A2E Analog Input TxPDO-MapSamples 46 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A2E:0	Analog Input TxPDO-MapSamples 46 Ch.1	PDO Mapping TxPDO 47	UINT8	RO	0x01 (1_{dec})
1A2E:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x2D ())	UINT32	RO	0x6001:2E, 16

Index 1A2F Analog Input TxPDO-MapSamples 47 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A2F:0	Analog Input TxPDO-MapSamples 47 Ch.1	PDO Mapping TxPDO 48	UINT8	RO	0x01 (1_{dec})
1A2F:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x2E ())	UINT32	RO	0x6001:2F, 16

Index 1A30 Analog Input TxPDO-MapSamples 48 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A30:0	Analog Input TxPDO-MapSamples 48 Ch.1	PDO Mapping TxPDO 49	UINT8	RO	0x01 (1_{dec})
1A30:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x2F ())	UINT32	RO	0x6001:30, 16

Index 1A31 Analog Input TxPDO-MapSamples 49 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A31:0	Analog Input TxPDO-MapSamples 49 Ch.1	PDO Mapping TxPDO 50	UINT8	RO	0x01 (1_{dec})
1A31:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x30 ())	UINT32	RO	0x6001:31, 16

Index 1A32 Analog Input TxPDO-MapSamples 50 Ch.1

Index (hex)	Name	Meaning	Data type	Flags	Default
1A32:0	Analog Input TxPDO-MapSamples 50 Ch.1	PDO Mapping TxPDO 51	UINT8	RO	0x01 (1_{dec})
1A32:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x31 ())	UINT32	RO	0x6001:32, 16

Index 1A40 Analog Input TxPDO-MapStatus Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A40:0	Analog Input TxPDO-MapStatus Ch.2	PDO Mapping TxPDO 65	UINT8	RO	0x09 (9_{dec})
1A40:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0B ())	UINT32	RO	0x6010:01, 1
1A40:02	SubIndex 002	2. PDO Mapping entry (object 0x6010 (Status), entry 0x02 (Overrange))	UINT32	RO	0x6010:02, 1
1A40:03	SubIndex 003	3. PDO Mapping entry (object 0x6010 (Status), entry 0x03 (Limit 1))	UINT32	RO	0x6010:03, 2
1A40:04	SubIndex 004	4. PDO Mapping entry (object 0x6010 (Status), entry 0x05 (Limit 2))	UINT32	RO	0x0000:00, 2
1A40:05	SubIndex 005	5. PDO Mapping entry (object 0x6010 (Status), entry 0x07 (Error))	UINT32	RO	0x6010:07, 1
1A40:06	SubIndex 006	6. PDO Mapping entry (6 bits align)	UINT32	RO	0x0000:00, 6
1A40:07	SubIndex 007	7. PDO Mapping entry (object 0x1C33 (SM input parameter), entry 0x20 (Sync error))	UINT32	RO	0x1C33:20, 1
1A40:08	SubIndex 008	8. PDO Mapping entry (object 0x1840, entry 0x07)	UINT32	RO	0x1840:07, 1
1A40:09	SubIndex 009	9. PDO Mapping entry (object 0x1840, entry 0x09)	UINT32	RO	0x1840:09, 1

Index 1A41 Analog Input TxPDO-MapSamples 1 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A41:0	Analog Input TxPDO-MapSamples 1 Ch.2	PDO Mapping TxPDO 66	UINT8	RO	0x01 (1_{dec})
1A41:01	SubIndex 001	1. PDO Mapping entry (object 0x6002 (Samples), entry 0x40 ())	UINT32	RO	0x6011:01, 16

Index 1A42 Analog Input TxPDO-MapSamples 2 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A42:0	Analog Input TxPDO-MapSamples 2 Ch.2	PDO Mapping TxPDO 67	UINT8	RO	0x01 (1_{dec})
1A42:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x02 ())	UINT32	RO	0x6011:02, 16

Index 1A43 Analog Input TxPDO-MapSamples 3 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A43:0	Analog Input TxPDO-MapSamples 3 Ch.2	PDO Mapping TxPDO 68	UINT8	RO	0x01 (1_{dec})
1A43:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x03 ())	UINT32	RO	0x6011:03, 16

Index 1A44 Analog Input TxPDO-MapSamples 4 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A44:0	Analog Input TxPDO-MapSamples 4 Ch.2	PDO Mapping TxPDO 69	UINT8	RO	0x01 (1_{dec})
1A44:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x04 ())	UINT32	RO	0x6011:04, 16

Index 1A45 Analog Input TxPDO-MapSamples 5 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A45:0	Analog Input TxPDO-MapSamples 5 Ch.2	PDO Mapping TxPDO 70	UINT8	RO	0x01 (1_{dec})
1A45:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x05 ())	UINT32	RO	0x6011:05, 16

Index 1A46 Analog Input TxPDO-MapSamples 6 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A46:0	Analog Input TxPDO-MapSamples 6 Ch.2	PDO Mapping TxPDO 71	UINT8	RO	0x01 (1_{dec})
1A46:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x06 ())	UINT32	RO	0x6011:06, 16

Index 1A47 Analog Input TxPDO-MapSamples 7 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A47:0	Analog Input TxPDO-MapSamples 7 Ch.2	PDO Mapping TxPDO 72	UINT8	RO	0x01 (1_{dec})
1A47:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x07 ())	UINT32	RO	0x6011:07, 16

Index 1A48 Analog Input TxPDO-MapSamples 8 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A48:0	Analog Input TxPDO-MapSamples 8 Ch.2	PDO Mapping TxPDO 73	UINT8	RO	0x01 (1_{dec})
1A48:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x08 ())	UINT32	RO	0x6011:08, 16

Index 1A49 Analog Input TxPDO-MapSamples 9 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A49:0	Analog Input TxPDO-MapSamples 9 Ch.2	PDO Mapping TxPDO 74	UINT8	RO	0x01 (1_{dec})
1A49:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x09 ())	UINT32	RO	0x6011:09, 16

Index 1A4A Analog Input TxPDO-MapSamples 10 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A4A:0	Analog Input TxPDO-MapSamples 10 Ch.2	PDO Mapping TxPDO 75	UINT8	RO	0x01 (1_{dec})
1A4A:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0A ())	UINT32	RO	0x6011:0A, 16

Index 1A4B Analog Input TxPDO-MapSamples 11 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A4B:0	Analog Input TxPDO-MapSamples 11 Ch.2	PDO Mapping TxPDO 76	UINT8	RO	0x01 (1_{dec})
1A4B:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0B ())	UINT32	RO	0x6011:0B, 16

Index 1A4C Analog Input TxPDO-MapSamples 12 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A4C:0	Analog Input TxPDO-MapSamples 12 Ch.2	PDO Mapping TxPDO 77	UINT8	RO	0x01 (1_{dec})
1A4C:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0C ())	UINT32	RO	0x6011:0C, 16

Index 1A4D Analog Input TxPDO-MapSamples 13 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A4D:0	Analog Input TxPDO-MapSamples 13 Ch.2	PDO Mapping TxPDO 78	UINT8	RO	0x01 (1_{dec})
1A4D:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0D ())	UINT32	RO	0x6011:0D, 16

Index 1A4E Analog Input TxPDO-MapSamples 14 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A4E:0	Analog Input TxPDO-MapSamples 14 Ch.2	PDO Mapping TxPDO 79	UINT8	RO	0x01 (1_{dec})
1A4E:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0E ())	UINT32	RO	0x6011:0E, 16

Index 1A4F Analog Input TxPDO-MapSamples 15 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A4F:0	Analog Input TxPDO-MapSamples 15 Ch.2	PDO Mapping TxPDO 80	UINT8	RO	0x01 (1_{dec})
1A4F:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0F ())	UINT32	RO	0x6011:0F, 16

Index 1A50 Analog Input TxPDO-MapSamples 16 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A50:0	Analog Input TxPDO-MapSamples 16 Ch.2	PDO Mapping TxPDO 81	UINT8	RO	0x01 (1_{dec})
1A50:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0F ())	UINT32	RO	0x6011:10, 16

Index 1A51 Analog Input TxPDO-MapSamples 17 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A51:0	Analog Input TxPDO-MapSamples 17 Ch.2	PDO Mapping TxPDO 82	UINT8	RO	0x01 (1_{dec})
1A51:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x10 ())	UINT32	RO	0x6011:11, 16

Index 1A52 Analog Input TxPDO-MapSamples 18 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A52:0	Analog Input TxPDO-MapSamples 18 Ch.2	PDO Mapping TxPDO 83	UINT8	RO	0x01 (1_{dec})
1A52:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x11 ())	UINT32	RO	0x6011:12, 16

Index 1A53 Analog Input TxPDO-MapSamples 19 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A53:0	Analog Input TxPDO-MapSamples 19 Ch.2	PDO Mapping TxPDO 84	UINT8	RO	0x01 (1_{dec})
1A53:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x12 ())	UINT32	RO	0x6011:13, 16

Index 1A54 Analog Input TxPDO-MapSamples 20 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A54:0	Analog Input TxPDO-MapSamples 20 Ch.2	PDO Mapping TxPDO 85	UINT8	RO	0x01 (1_{dec})
1A54:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x13 ())	UINT32	RO	0x6011:14, 16

Index 1A55 Analog Input TxPDO-MapSamples 21 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A55:0	Analog Input TxPDO-MapSamples 21 Ch.2	PDO Mapping TxPDO 86	UINT8	RO	0x01 (1_{dec})
1A55:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x14 ())	UINT32	RO	0x6011:15, 16

Index 1A56 Analog Input TxPDO-MapSamples 22 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A56:0	Analog Input TxPDO-MapSamples 22 Ch.2	PDO Mapping TxPDO 87	UINT8	RO	0x01 (1_{dec})
1A56:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x05 ())	UINT32	RO	0x6011:16, 16

Index 1A57 Analog Input TxPDO-MapSamples 23 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A57:0	Analog Input TxPDO-MapSamples 23 Ch.2	PDO Mapping TxPDO 88	UINT8	RO	0x01 (1_{dec})
1A57:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x06 ())	UINT32	RO	0x6011:17, 16

Index 1A58 Analog Input TxPDO-MapSamples 24 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A58:0	Analog Input TxPDO-MapSamples 24 Ch.2	PDO Mapping TxPDO 89	UINT8	RO	0x01 (1_{dec})
1A58:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x07 ())	UINT32	RO	0x6011:18, 16

Index 1A59 Analog Input TxPDO-MapSamples 25 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A59:0	Analog Input TxPDO-MapSamples 25 Ch.2	PDO Mapping TxPDO 90	UINT8	RO	0x01 (1_{dec})
1A59:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x08 ())	UINT32	RO	0x6011:19, 16

Index 1A5A Analog Input TxPDO-MapSamples 26 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A5A:0	Analog Input TxPDO-MapSamples 26 Ch.2	PDO Mapping TxPDO 91	UINT8	RO	0x01 (1_{dec})
1A5A:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x09 ())	UINT32	RO	0x6011:1A, 16

Index 1A5B Analog Input TxPDO-MapSamples 27 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A5B:0	Analog Input TxPDO-MapSamples 27 Ch.2	PDO Mapping TxPDO 92	UINT8	RO	0x01 (1_{dec})
1A5B:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0A ())	UINT32	RO	0x6011:1B, 16

Index 1A5C Analog Input TxPDO-MapSamples 28 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A5C:0	Analog Input TxPDO-MapSamples 28 Ch.2	PDO Mapping TxPDO 93	UINT8	RO	0x01 (1_{dec})
1A5C:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0B ())	UINT32	RO	0x6011:1C, 16

Index 1A5D Analog Input TxPDO-MapSamples 29 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A5D:0	Analog Input TxPDO-MapSamples 29 Ch.2	PDO Mapping TxPDO 94	UINT8	RO	0x01 (1_{dec})
1A5D:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0C ())	UINT32	RO	0x6011:1D, 16

Index 1A5E Analog Input TxPDO-MapSamples 30 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A5E:0	Analog Input TxPDO-MapSamples 30 Ch.2	PDO Mapping TxPDO 95	UINT8	RO	0x01 (1_{dec})
1A5E:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0D ())	UINT32	RO	0x6011:1E, 16

Index 1A5F Analog Input TxPDO-MapSamples 31 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A5F:0	Analog Input TxPDO-MapSamples 31 Ch.2	PDO Mapping TxPDO 96	UINT8	RO	0x01 (1_{dec})
1A5F:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0E ())	UINT32	RO	0x6011:1F, 16

Index 1A60 Analog Input TxPDO-MapSamples 32 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A60:0	Analog Input TxPDO-MapSamples 32 Ch.2	PDO Mapping TxPDO 97	UINT8	RO	0x01 (1_{dec})
1A60:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x0F ())	UINT32	RO	0x6011:20, 16

Index 1A61 Analog Input TxPDO-MapSamples 33 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A61:0	Analog Input TxPDO-MapSamples 33 Ch.2	PDO Mapping TxPDO 98	UINT8	RO	0x01 (1_{dec})
1A61:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x10 ())	UINT32	RO	0x6011:21, 16

Index 1A62 Analog Input TxPDO-MapSamples 34 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A62:0	Analog Input TxPDO-MapSamples 34 Ch.2	PDO Mapping TxPDO 99	UINT8	RO	0x01 (1_{dec})
1A62:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x11 ())	UINT32	RO	0x6011:22, 16

Index 1A63 Analog Input TxPDO-MapSamples 35 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A63:0	Analog Input TxPDO-MapSamples 35 Ch.2	PDO Mapping TxPDO 100	UINT8	RO	0x01 (1_{dec})
1A63:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x12 ())	UINT32	RO	0x6011:23, 16

Index 1A64 Analog Input TxPDO-MapSamples 36 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A64:0	Analog Input TxPDO-MapSamples 36 Ch.2	PDO Mapping TxPDO 101	UINT8	RO	0x01 (1_{dec})
1A64:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x13 ())	UINT32	RO	0x6011:24, 16

Index 1A65 Analog Input TxPDO-MapSamples 37 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A65:0	Analog Input TxPDO-MapSamples 37 Ch.2	PDO Mapping TxPDO 102	UINT8	RO	0x01 (1_{dec})
1A65:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x14 ())	UINT32	RO	0x6011:25, 16

Index 1A66 Analog Input TxPDO-MapSamples 38 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A66:0	Analog Input TxPDO-MapSamples 38 Ch.2	PDO Mapping TxPDO 103	UINT8	RO	0x01 (1_{dec})
1A66:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x15 ())	UINT32	RO	0x6011:26, 16

Index 1A67 Analog Input TxPDO-MapSamples 39 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A67:0	Analog Input TxPDO-MapSamples 39 Ch.2	PDO Mapping TxPDO 104	UINT8	RO	0x01 (1_{dec})
1A67:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x16 ())	UINT32	RO	0x6011:27, 16

Index 1A68 Analog Input TxPDO-MapSamples 40 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A68:0	Analog Input TxPDO-MapSamples 40 Ch.2	PDO Mapping TxPDO 105	UINT8	RO	0x01 (1_{dec})
1A68:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x17 ())	UINT32	RO	0x6011:28, 16

Index 1A69 Analog Input TxPDO-MapSamples 41 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A69:0	Analog Input TxPDO-MapSamples 41 Ch.2	PDO Mapping TxPDO 106	UINT8	RO	0x01 (1_{dec})
1A69:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x18 ())	UINT32	RO	0x6011:29, 16

Index 1A6A Analog Input TxPDO-MapSamples 42 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A6A:0	Analog Input TxPDO-MapSamples 42 Ch.2	PDO Mapping TxPDO 107	UINT8	RO	0x01 (1_{dec})
1A6A:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x19 ())	UINT32	RO	0x6011:2A, 16

Index 1A6B Analog Input TxPDO-MapSamples 43 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A6B:0	Analog Input TxPDO-MapSamples 43 Ch.2	PDO Mapping TxPDO 108	UINT8	RO	0x01 (1_{dec})
1A6B:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x1A ())	UINT32	RO	0x6011:2B, 16

Index 1A6C Analog Input TxPDO-MapSamples 44 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A6C:0	Analog Input TxPDO-MapSamples 44 Ch.2	PDO Mapping TxPDO 109	UINT8	RO	0x01 (1_{dec})
1A6C:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x1B ())	UINT32	RO	0x6011:2C, 16

Index 1A6D Analog Input TxPDO-MapSamples 45 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A6D:0	Analog Input TxPDO-MapSamples 45 Ch.2	PDO Mapping TxPDO 110	UINT8	RO	0x01 (1_{dec})
1A6D:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x1C ())	UINT32	RO	0x6011:2D, 16

Index 1A6E Analog Input TxPDO-MapSamples 46 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A6E:0	Analog Input TxPDO-MapSamples 46 Ch.2	PDO Mapping TxPDO 111	UINT8	RO	0x01 (1_{dec})
1A6E:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x1D ())	UINT32	RO	0x6011:2E, 16

Index 1A6F Analog Input TxPDO-MapSamples 47 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A6F:0	Analog Input TxPDO-MapSamples 47 Ch.2	PDO Mapping TxPDO 112	UINT8	RO	0x01 (1_{dec})
1A6F:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x1E ())	UINT32	RO	0x6011:2F, 16

Index 1A70 Analog Input TxPDO-MapSamples 48 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A70:0	Analog Input TxPDO-MapSamples 48 Ch.2	PDO Mapping TxPDO 113	UINT8	RO	0x01 (1_{dec})
1A70:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x1F ())	UINT32	RO	0x6011:30, 16

Index 1A71 Analog Input TxPDO-MapSamples 49 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A71:0	Analog Input TxPDO-MapSamples 49 Ch.2	PDO Mapping TxPDO 114	UINT8	RO	0x01 (1_{dec})
1A71:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x20 ())	UINT32	RO	0x6011:31, 16

Index 1A72 Analog Input TxPDO-MapSamples 50 Ch.2

Index (hex)	Name	Meaning	Data type	Flags	Default
1A72:0	Analog Input TxPDO-MapSamples 50 Ch.2	PDO Mapping TxPDO 115	UINT8	RO	0x01 (1_{dec})
1A72:01	SubIndex 001	1. PDO Mapping entry (object 0x6011 (Samples), entry 0x32 ())	UINT32	RO	0x6011:32, 16

Index 1A80 Analog Input Timestamp TxPDO-Map NextSync1Time

Index (hex)	Name	Meaning	Data type	Flags	Default
1A80:0	Analog Input Timestamp TxPDO-Map NextSync1Time	PDO Mapping TxPDO 129	UINT8	RO	0x01 (1_{dec})
1A80:01	SubIndex 001	1. PDO Mapping entry (object 0x6012 (Samples), entry 0x2F ())	UINT32	RO	0x6020:01, 64

Index 1A81 Analog Input Timestamp TxPDO-Map Cycle Count

Index (hex)	Name	Meaning	Data type	Flags	Default
1A81:0	Analog Input Timestamp TxPDO-Map Cycle Count	PDO Mapping TxPDO 130	UINT8	RO	0x01 (1_{dec})
1A81:01	SubIndex 001	1. PDO Mapping entry (object 0x6021 (CycleCount), entry 0x01 (Cycle Count))	UINT32	RO	0x6021:01, 16

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	0x00 (0_{dec})

Index 1C13 TxPDO assign

Index (hex)	Name	Meaning	Data type	Flags	Default
1C13:0	TxPDO assign	PDO Assign Inputs	UINT8	RW	0x02 (2_{dec})
1C13:01	Subindex 001	1. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A80 (6784_{dec})
1C13:02	Subindex 002	2. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x1A81 (6785_{dec})
1C13:03	Subindex 003	3. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})
...	...	...	...	...	..
1C13:68	Subindex 104	104. allocated TxPDO (contains the index of the associated TxPDO mapping object)	UINT16	RW	0x0000 (0_{dec})

Index 1C33 SM input parameter

Index (hex)	Name	Meaning	Data type	Flags	Default
1C33:0	SM input parameter	Synchronization parameters for the inputs	UINT8	RO	0x20 (32$_{dec}$)
1C33:01	Sync mode	Current synchronization mode: • 0: Free Run • 1: Synchron with SM 3 Event (no outputs available) • 2: DC - Synchron with SYNC0 Event • 3: DC - Synchron with SYNC1 Event • 34: Synchron with SM 2 Event (outputs available)	UINT16	RW	0x0022 (34$_{dec}$)
1C33:02	Cycle time	as 0x1C32:02	UINT32	RW	0x000F4240 (1000000$_{dec}$)
1C33:03	Shift time	Time between SYNC0 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0$_{dec}$)
1C33:04	Sync modes supported	Supported synchronization modes: • 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 or 0x1C33:08)	UINT16	RO	0xC007 (49159$_{dec}$)
1C33:05	Minimum cycle time	as 0x1C32:05	UINT32	RO	0x00002710 (10000$_{dec}$)
1C33:06	Calc and copy time	Time between reading of the inputs and availability of the inputs for the master (in ns, only DC mode)	UINT32	RO	0x00000000 (0$_{dec}$)
1C33:07	Minimum delay time		UINT32	RO	0x00000000 (0$_{dec}$)
1C33:08	Command	as 0x1C32:08	UINT16	RW	0x0000 (0$_{dec}$)
1C33:09	Maximum delay time	Time between SYNC1 event and reading of the inputs (in ns, only DC mode)	UINT32	RO	0x00000000 (0$_{dec}$)
1C33:0B	SM event missed counter	as 0x1C32:11	UINT16	RO	0x0000 (0$_{dec}$)
1C33:0C	Cycle exceeded counter	as 0x1C32:12	UINT16	RO	0x0000 (0$_{dec}$)
1C33:0D	Shift too short counter	as 0x1C32:13	UINT16	RO	0x0000 (0$_{dec}$)
1C33:20	Sync error	as 0x1C32:32	BOOLEAN	RO	0x00 (0$_{dec}$)

Index F000 Modular device profile

Index (hex)	Name	Meaning	Data type	Flags	Default
F000:0	Modular device profile	General information for the modular device profile	UINT8	RO	0x02 (2$_{dec}$)
F000:01	Module index distance	Index distance of the objects of the individual channels	UINT16	RO	0x0010 (16$_{dec}$)
F000:02	Maximum number of modules	Number of channels	UINT16	RO	0x0003 (3$_{dec}$)

Index F008 Code word

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

Index F010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list	Maximum subindex	UINT8	RW	0x03 (3_{dec})
F010:01	SubIndex 001	Profile AI	UINT32	RW	0x0000012C (300_{dec})
F010:02	SubIndex 002	Profile AI	UINT32	RW	0x0000012C (300_{dec})
F010:03	SubIndex 003	Profile AI	UINT32	RW	0x0000012C (300_{dec})

5.4.3 Update of the terminal

Information on updating the terminal description can be found in the chapter Firmware-Update [▶ 204].

5.5 Notices on analog specifications

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

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

5.5.1 Full scale value (FSV)

An I/O device with an analog input measures over a nominal measuring range that is limited by an upper and a lower limit (initial value and end value); these can usually be taken from the device designation.
The range between the two limits is called the measuring span and corresponds to the equation (end value - initial value). Analogous to pointing devices this is the measuring scale (see IEC 61131) or also the dynamic range.

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

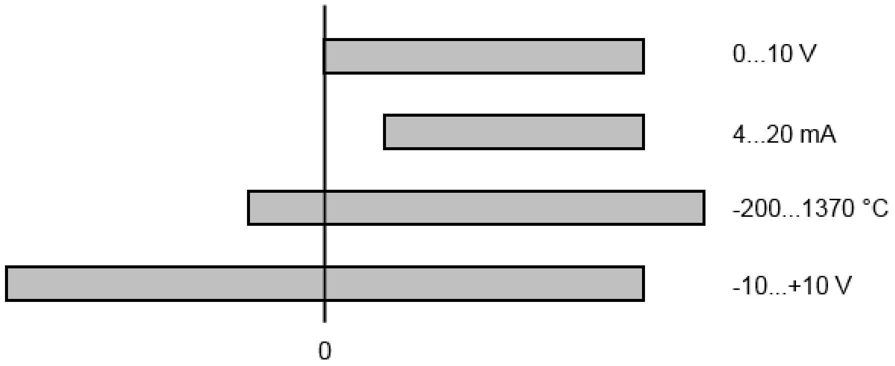


Fig. 171: Full scale value, measuring span

For the above **examples** this means:

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

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

5.5.2 Measuring error/ measurement deviation

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

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

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

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

The maximum deviation can also be specified directly.

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

Lower measuring error

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

This applies to analog output devices.

5.5.3 Temperature coefficient tK [ppm/K]

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

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

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

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

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

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

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

5.5.4 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.

5.5.5 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.

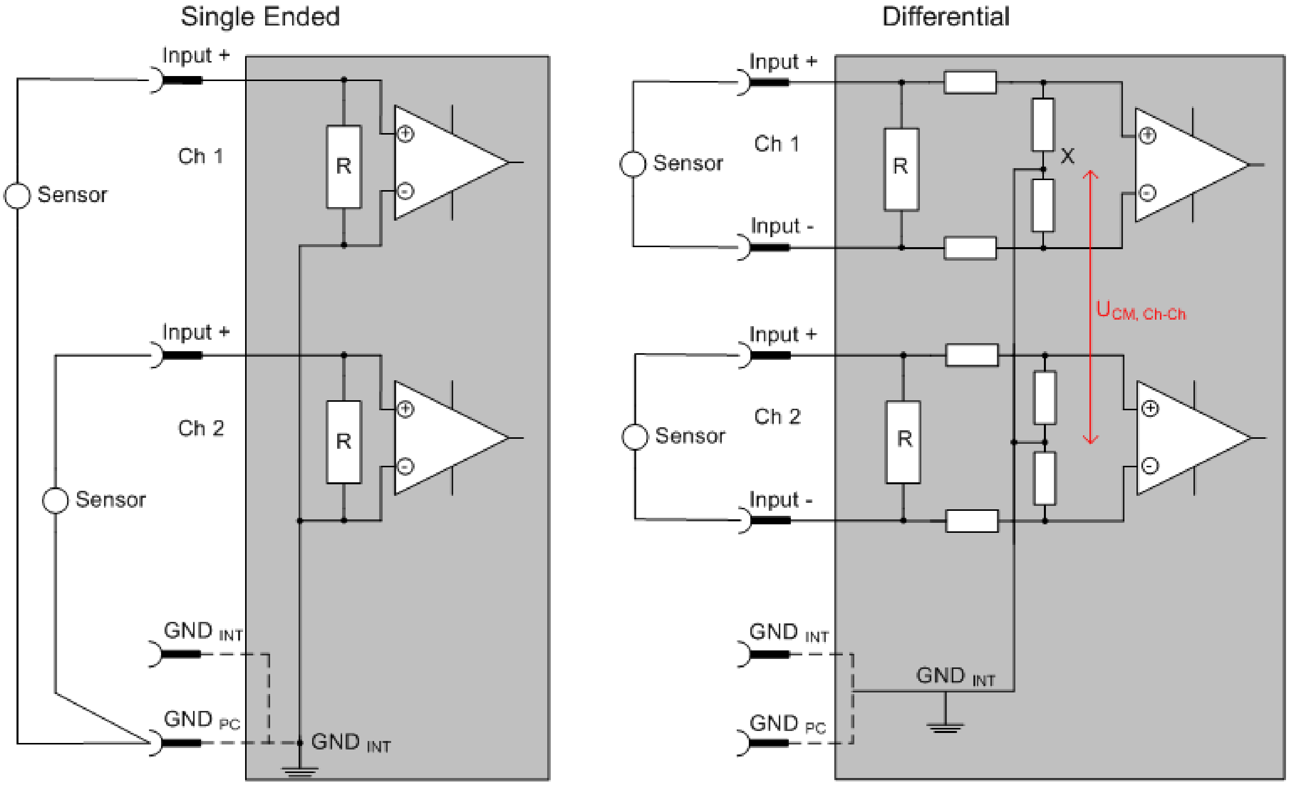


Fig. 172: SE and DIFF module as 2-channel version

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

The basic rule:

- Analog measurements always take the form of voltage measurements between two potential points. For voltage measurements a large R is used, in order to ensure a high impedance. For current measurements a small R is used as shunt. If the purpose is resistance measurement, corresponding considerations are applied.

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

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

Explanation

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

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

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

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

Self-supplied sensors

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

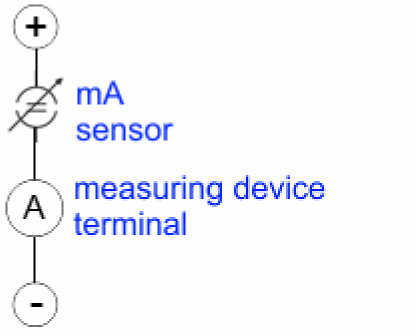


Fig. 173: 2-wire connection

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

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

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

Externally supplied sensors

- 3- and 4-wire connection see Fig. *Connection of externally supplied sensors*, cf. IEC60381-1
- the sensor draws the energy/operating voltage for its own operation from 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/ boxes (and related product groups), in this case the polarity/direction of current have to be observed.

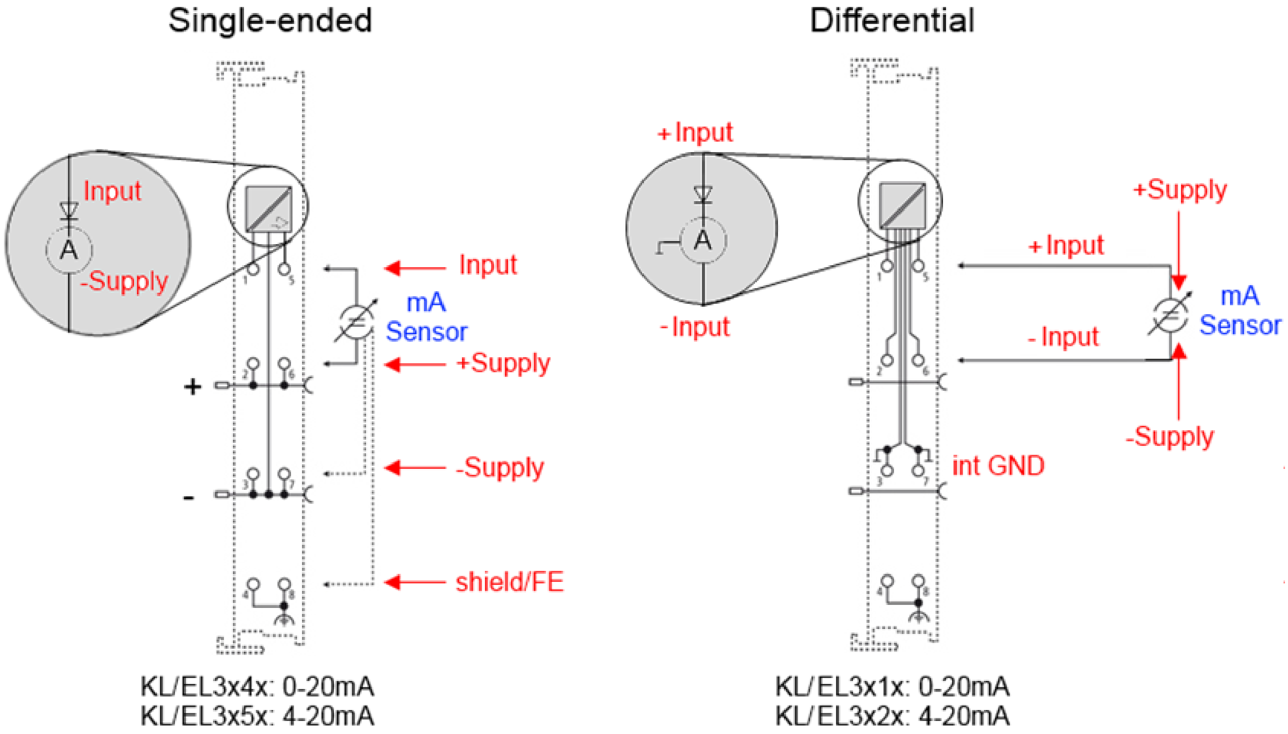


Fig. 174: Connection of externally supplied sensors

Classification of Beckhoff Terminals/ Boxes - Beckhoff 0/4-20 mA Terminals/ Boxes (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	

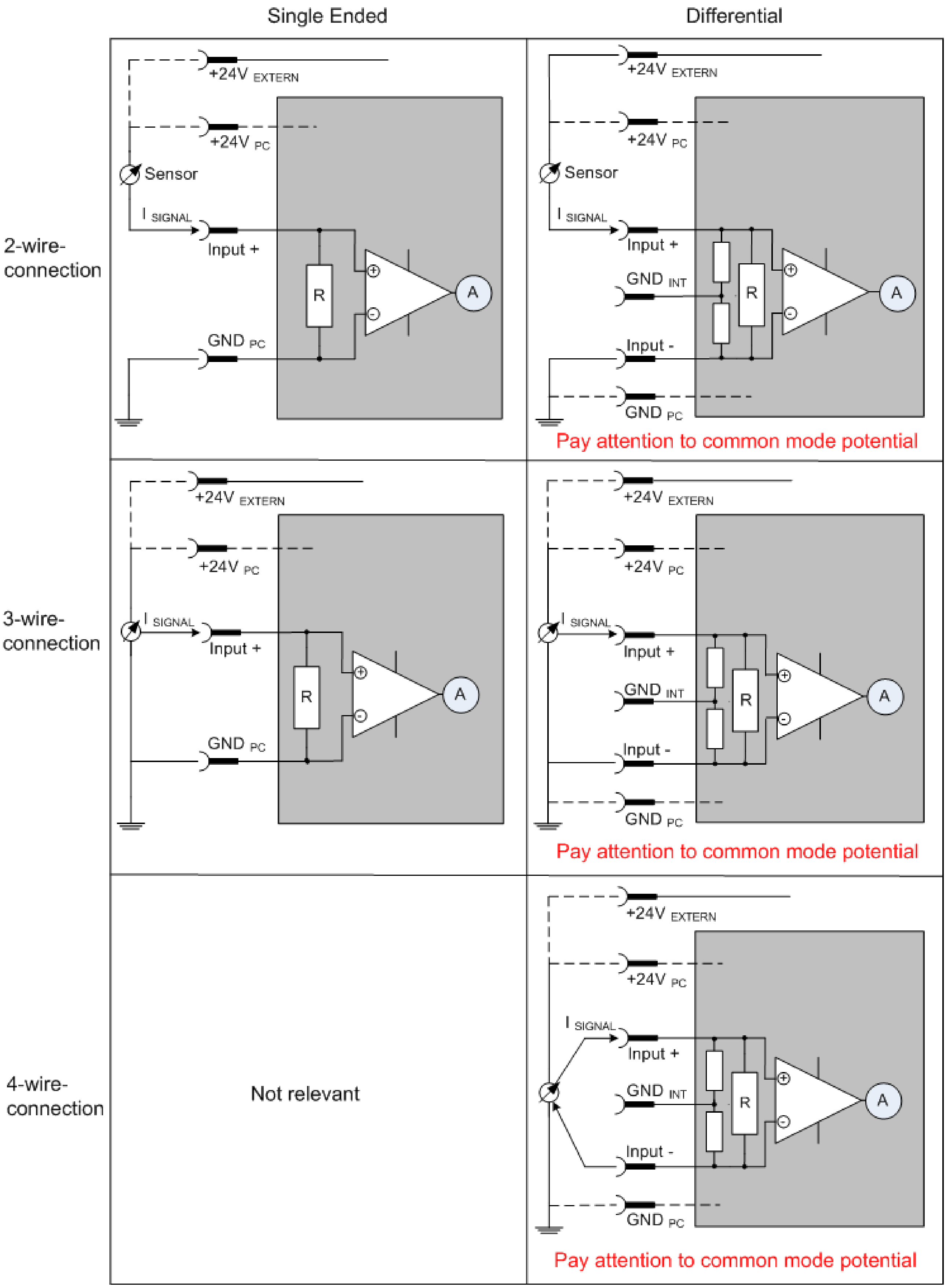


Fig. 175: 2-, 3- and 4-wire connection at single-ended and differential inputs

5.5.6 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.

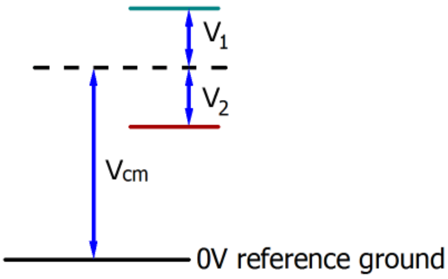


Fig. 176: Common-mode voltage (V_{cm})

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

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

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

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

Reference ground samples for Beckhoff IO devices:

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

5.5.7 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

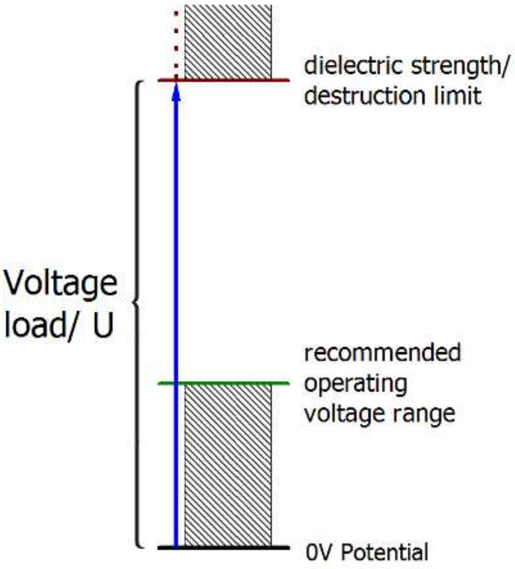


Fig. 177: Recommended operating voltage range

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)

5.5.8 Temporal aspects of analog/digital conversion

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

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

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

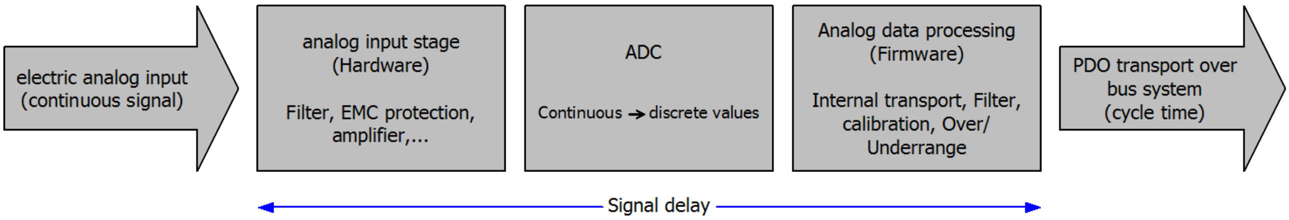

electric analog input (continuous signal)
analog input stage (Hardware)
Filter, EMC protection, amplifier,...
ADC
Continuous → discrete values
Analog data processing (Firmware)
Internal transport, Filter, calibration, Over/ Underrange
PDO transport over bus system (cycle time)
Signal delay

Fig. 178: Signal processing analog input

Two aspects are crucial from a user perspective:

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

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

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

1. Minimum conversion time [ms, µs]

This is the reciprocal value of the maximum **sampling rate** [sps, samples per second]:
Indicates how often the analog channel makes a newly detected process data value available for collection by the fieldbus. Whether the fieldbus (EtherCAT, K-bus) fetches the value with the same speed (i.e. synchronous), or more quickly (if the AI channel operates in slow FreeRun mode) or more slowly (e.g. with oversampling), is then a question of the fieldbus setting and which modes the AI device supports.
For EtherCAT devices the so-called toggle bit indicates (by toggling) for the diagnostic PDOs when a newly determined analog value is available.
Accordingly, a maximum conversion time, i.e. a smallest sampling rate supported by the AI device, can be specified.
Corresponds to IEC 61131-2, section 7.10.2 2, “Sampling repeat time”

2. Typical signal delay

Corresponds to IEC 61131-2, section 7.10.2 1, “Sampling duration”. From this perspective it includes all internal hardware and firmware components, but not “external” delay components from the fieldbus or the controller (TwinCAT).
This delay is particularly relevant for absolute time considerations, if AI channels also provide a time stamp that corresponds to the amplitude value – which can be assumed to match the physically prevailing amplitude value at the time.
Due to the frequency-dependent signal delay time, a dedicated value can only be specified for a given signal. The value also depends on potentially variable filter settings of the channel.
A typical characterization in the device documentation may be:

2.1 Signal delay (step response)

Keywords: Settling time
The square wave signal can be generated externally with a frequency generator (note impedance!)
The 90 % limit is used as detection threshold.
The signal delay [ms, µs] is then the time interval between the (ideal) electrical square wave signal and the time at which the analog process value has reached the 90 % amplitude.

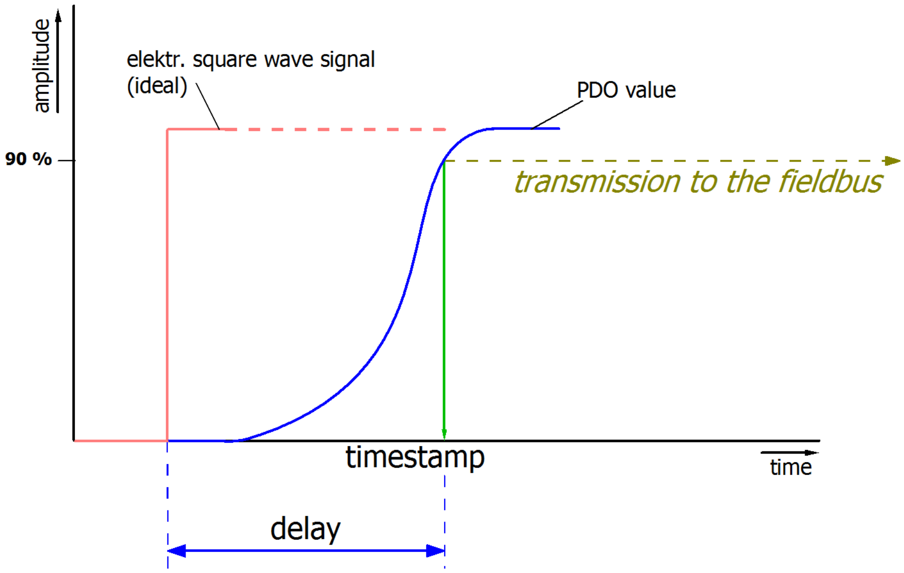


Fig. 179: Diagram signal delay (step response)

2.2 Signal delay (linear)

Keyword: Group delay
Describes the delay of a signal with constant frequency
A test signal can be generated externally with a frequency generator, e.g. as sawtooth or sine. A simultaneous square wave signal would be used as reference.
The signal delay [ms, µs] is then the interval between the applied electrical signal with a particular amplitude and the moment at which the analog process value reaches the same value.
A meaningful range must be selected for the test frequency, e.g. 1/20 of the maximum sampling rate.

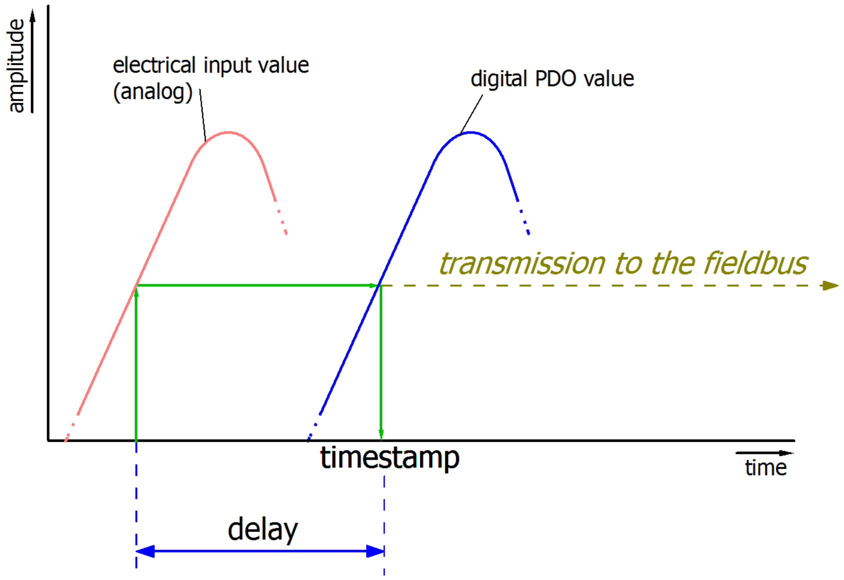


Fig. 180: Diagram signal delay (linear)

3. Additional Information

May be provided in the specification, e.g.

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

5.5.9 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 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.

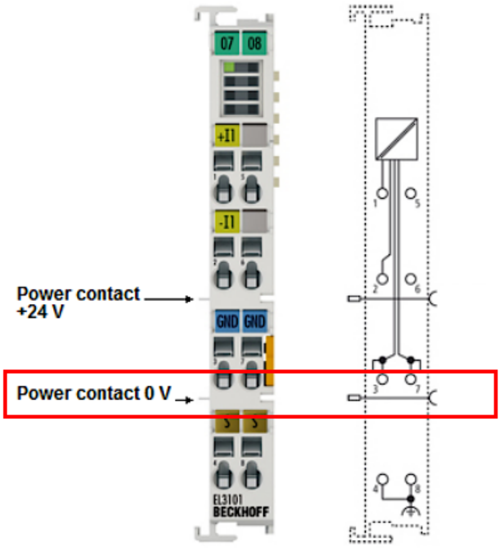
PGND, AGND

- 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: ⊥;

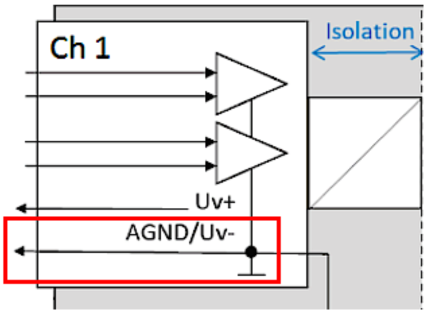
  hardly used any more, but actually means earth potential: ⏚.
- There may be several PGND/AGND networks in a system that are electrically isolated from one another.
- If a device has several AGNDs, due to isolation by channel, these can be numbered: AGND1, AGND2, …
- PGND
  - also called: GND_{PC}, 0 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 [▶ 173]). 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:



.

5.5.10 Sampling type: Simultaneous vs. multiplexed

Analog inputs and outputs in Beckhoff devices can operate in 2 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** *)
- 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. 181: 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.

*) 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:

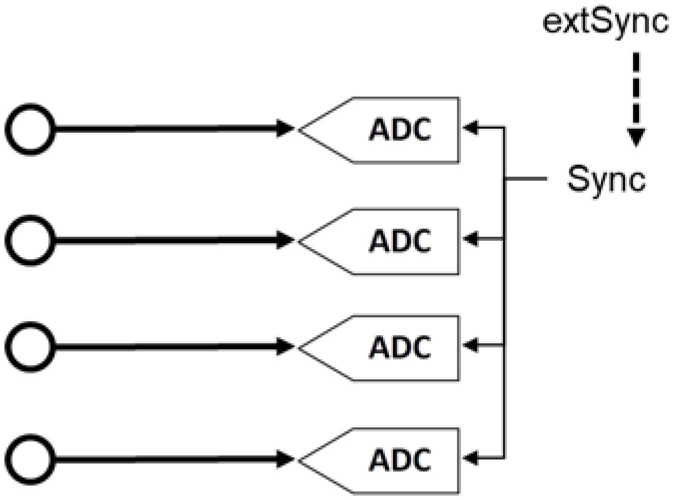


Fig. 182: Schematic representation simultaneous sampling with 4 ADC converters

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.

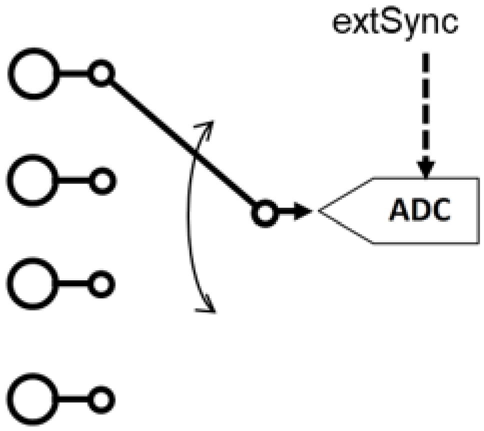


Fig. 183: Schematic representation of multiplex sampling with an ADC converter

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.

5.6 Application example

Use of ScopeView2 with oversampling

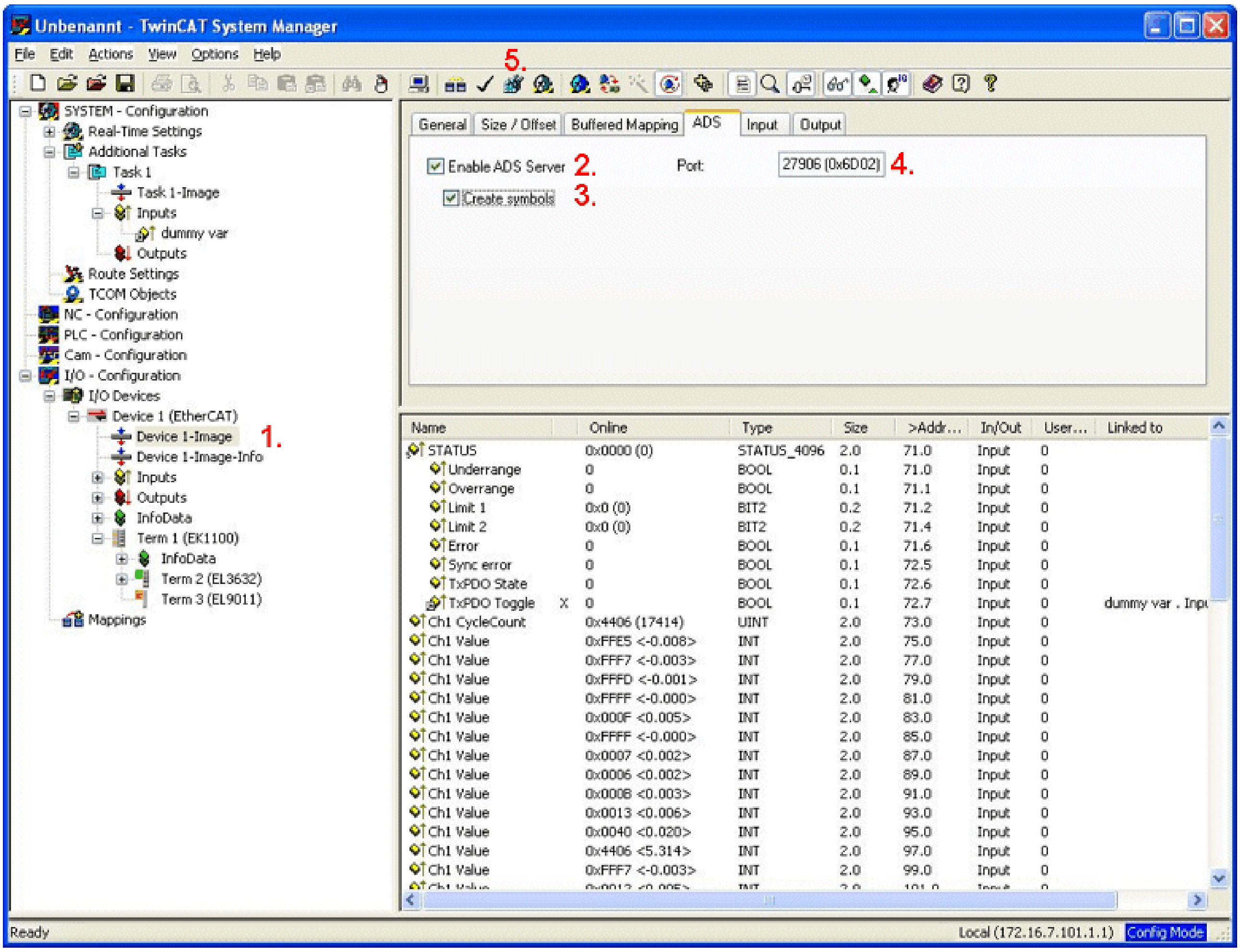
Preparation

- Use TwinCAT 2.11, build 1549 or higher
- Use current ScopeView2

Activation of oversampling

The following steps must be performed (fig. Activation of ScopeView2):

1. Select the process image of the EtherCAT master (in this case: “Device 1 - process image”)
2. Select “Enable ADS Server” in the “ADS” tab
3. Select “Create symbols”
4. Note the “Port” (usually 27905)
5. Adopt configuration (note: PLC restart is necessary!)



☑ Enable ADS Server

Port: 27906 (0x6D02)

☑ Create symbols

Name		Online	Type	Size	>Addr...	In/Out	User...	Linked to
STATUS		0x0000 (0)	STATUS_4096	2.0	71.0	Input	0	
Underrange		0	BOOL	0.1	71.0	Input	0	
Overrange		0	BOOL	0.1	71.1	Input	0	
Limit 1		0x0 (0)	BIT2	0.2	71.2	Input	0	
Limit 2		0x0 (0)	BIT2	0.2	71.4	Input	0	
Error		0	BOOL	0.1	71.6	Input	0	
Sync error		0	BOOL	0.1	72.5	Input	0	
TxPDO State		0	BOOL	0.1	72.6	Input	0	
TxPDO Toggle	X	0	BOOL	0.1	72.7	Input	0	dummy var . Inpu
Ch1 CycleCount		0x4406 (17414)	UINT	2.0	73.0	Input	0	
Ch1 Value		0xFFE5 <-0.008>	INT	2.0	75.0	Input	0	
Ch1 Value		0xFFF7 <-0.003>	INT	2.0	77.0	Input	0	
Ch1 Value		0xFFFD <-0.001>	INT	2.0	79.0	Input	0	
Ch1 Value		0xFFFF <-0.000>	INT	2.0	81.0	Input	0	
Ch1 Value		0x000F <0.005>	INT	2.0	83.0	Input	0	
Ch1 Value		0xFFFF <-0.000>	INT	2.0	85.0	Input	0	
Ch1 Value		0x0007 <0.002>	INT	2.0	87.0	Input	0	
Ch1 Value		0x0006 <0.002>	INT	2.0	89.0	Input	0	
Ch1 Value		0x0008 <0.003>	INT	2.0	91.0	Input	0	
Ch1 Value		0x0013 <0.006>	INT	2.0	93.0	Input	0	
Ch1 Value		0x0040 <0.020>	INT	2.0	95.0	Input	0	
Ch1 Value		0x4406 <5.314>	INT	2.0	97.0	Input	0	
Ch1 Value		0xFFF7 <-0.003>	INT	2.0	99.0	Input	0	

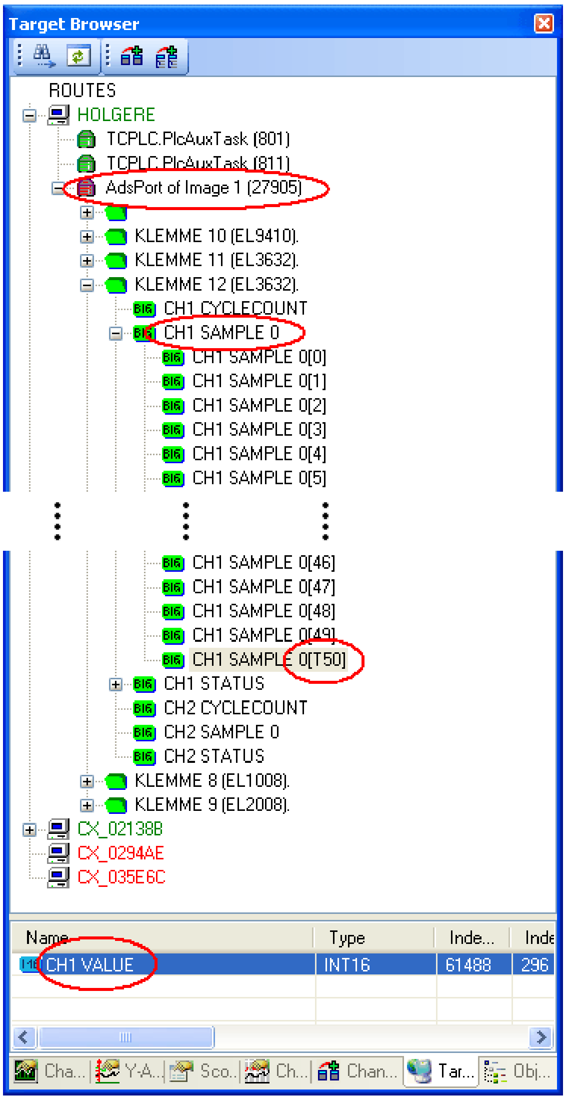
Fig. 184: Activation of ScopeView2

Use of the oversampling variable

The following steps must be performed (fig. *Target Browser*):

- Select ScopeView2 “View” - > “Target Browser”
- Select “Enable Server Ports” (button at top left)
- Enter the noted port (see fig. *Activation of ScopeView2*) in the field at the bottom left
- Confirm “Add” and “OK”
- A new entry, “AdsPort of Image 1”, appears in the Target Browser; select it and display the tree with [+]
- Select the variable “CH1 SAMPLE 0” or “CH2 SAMPLE 0” for the terminal used and display the tree with [+]

The last entry in the tree contains the oversampling variable (can be recognized by the index T+Oversampling factor)
A double-click on this variable displays its characteristics. A double-click on the variable with the blue box inserts it into the object browser.



Name	Type	Inde...	Inde
CH1 VALUE	INT16	61488	296

Fig. 185: Target Browser

Creating a task

An additional task must be used if no PLC is active: TwinCAT System Manager: “SYSTEM - Configuration” - > additional task - > rights mouse button “Add task” - > rights mouse button “Insert variable”. Create variable(s) and link with a terminal. The meaning and size of the variables are not relevant.

6 Error handling and diagnostics

6.1 Error descriptions and troubleshooting

Error Codes

Error Index 0x6000:07 [▶ 152]	Underange Index 0x6000:01 [▶ 152]	Overrange Index 0x6000:02 [▶ 152]	TxPDO State Index 0x6000:0F [▶ 152]	Sync Error Index 0x6000:0E [▶ 152]	Error description	Remedy
1	1				Measurement is below range	Reduce the input level, change the gain (filter settings)
1		1			Measuring range exceeded	Reduce the input level, change the gain (filter settings)
1					General measuring error	e.g. open circuit
				1	Synchronization error	Jitter of master too high, distributed clocks switched off

The error LED lights up only in case of open circuit.

Troubleshooting

The dialog for setting the sampling rate is missing

The dialog for setting the sampling rate is missing. The TwinCAT System Manager uses the “online description” of the terminal. The note to use the dialog for the parameterization is only available in the XML description.
Solution: Make sure that the latest XML description is used.

The amplitude is too small

Filter settings incorrect.
Solution: Correct the settings in the CoE in the entries 0x80n0:15 [▶ 151].

The terminal switches to SAFEOP

The terminal switches to SAFEOP. The real-time-settings are not accurate enough.
Solution: Use a PC without mobile chipset/CPU.

7 Appendix

7.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

7.2 Diagnostics – basic principles of diag messages

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

Definition

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

Each DiagMessage consists of

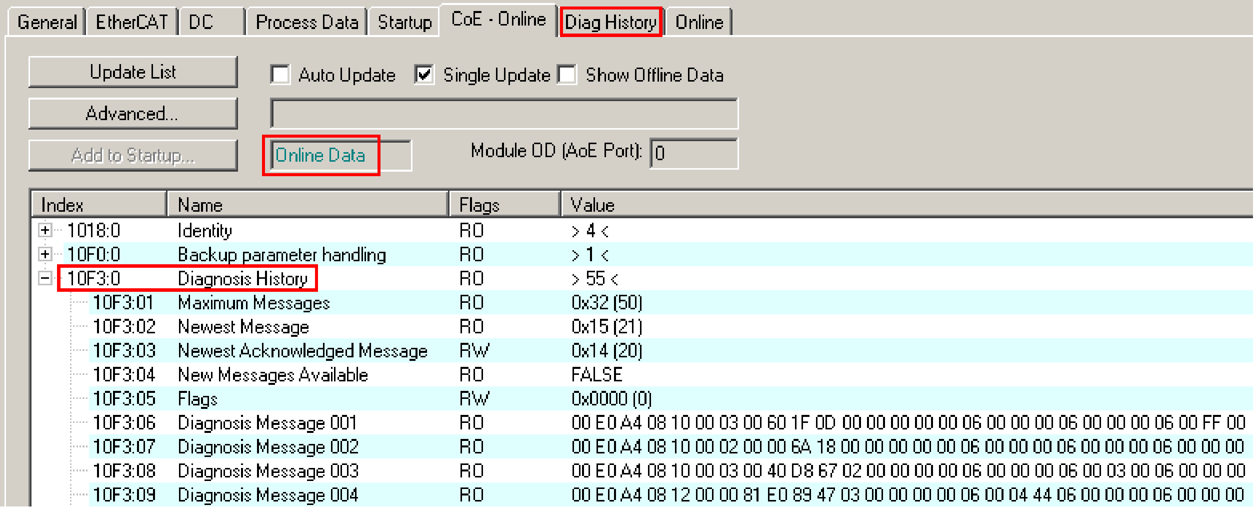
- Diag Code (4-byte)
- Flags (2-byte; info, warning or error)
- Text ID (2-byte; reference to explanatory text from the ESI/XML)
- Timestamp (8-byte, local slave time or 64-bit Distributed Clock time, if available)
- Dynamic parameters added by the firmware

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

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

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

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



☐ Auto Update ☑ Single Update ☐ Show Offline Data

Module OD (AoE Port): 0

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

Fig. 186: DiagMessages in the CoE

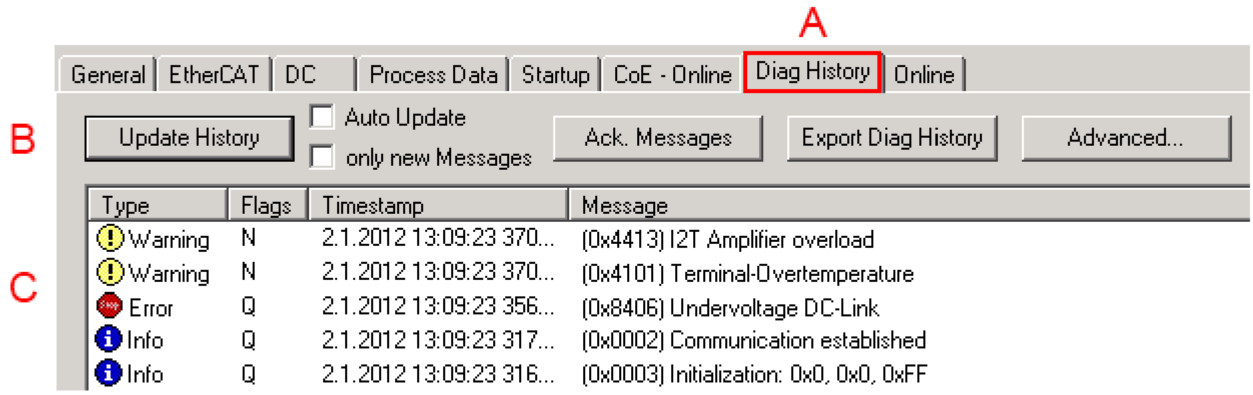
The subindex of the latest *DiagMessage* can be read under 0x10F3:02.

Support for commissioning

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

TwinCAT System Manager implementation

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



Type	Flags	Timestamp	Message
Warning	N	2.1.2012 13:09:23 370...	(0x4413) I2T Amplifier overload
Warning	N	2.1.2012 13:09:23 370...	(0x4101) Terminal-Overtemperature
Error	Q	2.1.2012 13:09:23 356...	(0x8406) Undervoltage DC-Link
Info	Q	2.1.2012 13:09:23 317...	(0x0002) Communication established
Info	Q	2.1.2012 13:09:23 316...	(0x0003) Initialization: 0x0, 0x0, 0xFF

Fig. 187: Implementation of the DiagMessage system in the TwinCAT System Manager

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

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

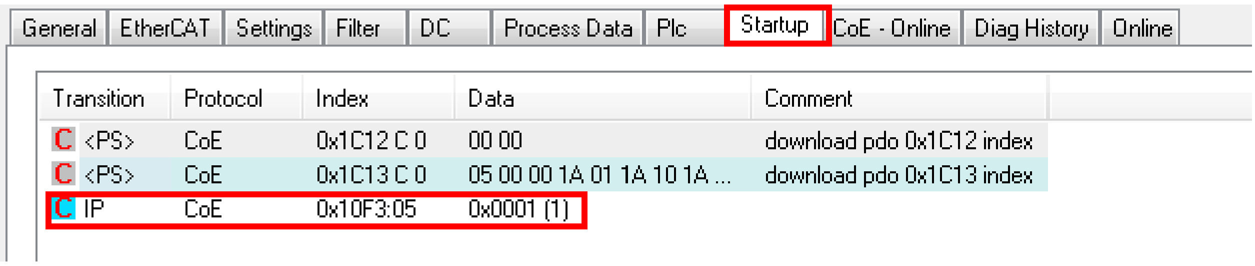
The meanings of the buttons are self-explanatory.

DiagMessages within the ADS Logger/Eventlogger

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

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

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



Transition	Protocol	Index	Data	Comment
C <PS>	CoE	0x1C12 C 0	00 00	download pdo 0x1C12 index
C <PS>	CoE	0x1C13 C 0	05 00 00 1A 01 1A 10 1A ...	download pdo 0x1C13 index
C IP	CoE	0x10F3:05	0x0001 (1)	

Fig. 188: Startup List

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

Reading messages into the PLC

- In preparation -

Interpretation

Time stamp

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

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

Structure of the Text ID

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

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

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

Overview of text IDs

Specific text IDs are listed in the device documentation.

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

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

Text ID	Type	Place	Text Message	Additional comment
0x4000	Warning		Warning: 0x%X, 0x%X, 0x%X	General warning; parameters depend on event. See device documentation for interpretation.
0x4001	Warning	System	Warning: 0x%X, 0x%X, 0x%X	
0x4002	Warning	System	%s: %s Connection Open (IN:%d OUT:%d API:%dms) from %d.%d.%d.%d successful	
0x4003	Warning	System	%s: %s Connection Close (IN:%d OUT:%d) from %d.%d.%d.%d successful	
0x4004	Warning	System	%s: %s Connection (IN:%d OUT:%d) with %d.%d.%d.%d timed out	
0x4005	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Error: %u)	
0x4006	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Input Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4007	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (Output Data Size expected: %d Byte(s) received: %d Byte(s))	
0x4008	Warning	System	%s: %s Connection Open (IN:%d OUT:%d) from %d.%d.%d.%d denied (RPI:%dms not supported -> API:%dms)	
0x4101	Warning	General	Terminal-Overtemperature	Overtemperature. The internal temperature of the terminal exceeds the parameterized warning threshold.
0x4102	Warning	General	Discrepancy in the PDO-Configuration	The selected PDOs do not match the set operating mode. Sample: Drive operates in velocity mode, but the velocity PDO is but not mapped in the PDOs.
0x417F	Warning	General	Warning: 0x%X, 0x%X, 0x%X	
0x428D	Warning	General	Challenge is not Random	
0x4300	Warning	Encoder	Subincrements deactivated: %d, %d	Sub-increments deactivated (despite activated configuration)
0x4301	Warning	Encoder	Encoder-Warning	General encoder error
0x4302	Warning	Encoder	Maximum frequency of the input signal is nearly reached (channel %d)	
0x4303	Warning	Encoder	Limit counter value was reduced because of the PDO configuration (channel %d)	
0x4304	Warning	Encoder	Reset counter value was reduced because of the PDO configuration (channel %d)	
0x4400	Warning	Drive	Drive is not calibrated: %d, %d	Drive is not calibrated
0x4401	Warning	Drive	Starttype not supported: 0x%X, %d	Start type is not supported
0x4402	Warning	Drive	Command rejected: %d, %d	Command rejected
0x4405	Warning	Drive	Invalid modulo subtype: %d, %d	Modulo sub-type invalid
0x4410	Warning	Drive	Target overrun: %d, %d	Target position exceeded
0x4411	Warning	Drive	DC-Link undervoltage (Warning)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x4412	Warning	Drive	DC-Link overvoltage (Warning)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x4413	Warning	Drive	I2T-Model Amplifier overload (Warning)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x4414	Warning	Drive	I2T-Model Motor overload (Warning)	• The motor is being operated outside the parameterized rated values. • The I2T-model of the motor is incorrectly parameterized.

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

Text ID	Type	Place	Text Message	Additional comment
0x8000	Error	System	%s: %s	
0x8001	Error	System	Error: 0x%X, 0x%X, 0x%X	General error; parameters depend on event. See device documentation for interpretation.
0x8002	Error	System	Communication aborted	Communication aborted
0x8003	Error	System	Configuration error: 0x%X, 0x%X, 0x%X	General; parameters depend on event. See device documentation for interpretation.
0x8004	Error	System	%s: Unsuccessful FwdOpen-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8005	Error	System	%s: FwdClose-Request sent to %d.%d.%d.%d (%s)	
0x8006	Error	System	%s: Unsuccessful FwdClose-Response received from %d.%d.%d.%d (%s) (Error: %u)	
0x8007	Error	System	%s: Connection with %d.%d.%d.%d (%s) closed	
0x8100	Error	General	Status word set: 0x%X, %d	Error bit set in the status word
0x8101	Error	General	Operation mode incompatible to PDO interface: 0x%X, %d	Mode of operation incompatible with the PDO interface
0x8102	Error	General	Invalid combination of Inputs and Outputs PDOs	Invalid combination of input and output PDOs
0x8103	Error	General	No variable linkage	No variables linked
0x8104	Error	General	Terminal-Overtemperature	The internal temperature of the terminal exceeds the parameterized error threshold. Activation of the terminal is prevented
0x8105	Error	General	PD-Watchdog	Communication between the fieldbus and the output stage is secured by a Watchdog. The axis is stopped automatically if the fieldbus communication is interrupted. • The EtherCAT connection was interrupted during operation. • The Master was switched to Config mode during operation.
0x8135	Error	General	Cycle time has to be a multiple of 125 µs	The IO or NC cycle time divided by 125 µs does not produce a whole number.
0x8136	Error	General	Configuration error: invalid sampling rate	Configuration error: Invalid sampling rate
0x8137	Error	General	Electronic type plate: CRC error	Content of the external name plate memory invalid.
0x8140	Error	General	Sync Error	Real-time violation
0x8141	Error	General	Sync%X Interrupt lost	Sync%X Interrupt lost
0x8142	Error	General	Sync Interrupt asynchronous	Sync Interrupt asynchronous
0x8143	Error	General	Jitter too big	Jitter limit violation
0x817F	Error	General	Error: 0x%X, 0x%X, 0x%X	
0x8200	Error	Communication	Write access error: %d, %d	Error while writing
0x8201	Error	Communication	No communication to field-side (Auxiliary voltage missing)	• There is no voltage applied to the power contacts. • A firmware update has failed.
0x8281	Error	Communication	Ownership failed: %X	
0x8282	Error	Communication	To many Keys founded	
0x8283	Error	Communication	Key Creation failed: %X	
0x8284	Error	Communication	Key loading failed	
0x8285	Error	Communication	Reading Public Key failed: %X	
0x8286	Error	Communication	Reading Public EK failed: %X	
0x8287	Error	Communication	Reading PCR Value failed: %X	
0x8288	Error	Communication	Reading Certificate EK failed: %X	
0x8289	Error	Communication	Challenge could not be hashed: %X	
0x828A	Error	Communication	Tickstamp Process failed	
0x828B	Error	Communication	PCR Process failed: %X	
0x828C	Error	Communication	Quote Process failed: %X	
0x82FF	Error	Communication	Bootmode not activated	Boot mode not activated
0x8300	Error	Encoder	Set position error: 0x%X, %d	Error while setting the position

Text ID	Type	Place	Text Message	Additional comment
0x8301	Error	Encoder	Encoder increments not configured: 0x%X, %d	Encoder increments not configured
0x8302	Error	Encoder	Encoder error	The amplitude of the resolver is too small
0x8303	Error	Encoder	Encoder power missing (channel %d)	
0x8304	Error	Encoder	Encoder communication error, channel: %X	Encoder communication error
0x8305	Error	Encoder	EnDat2.2 is not supported, channel: %X	EnDat2.2 is not supported
0x8306	Error	Encoder	Delay time, tolerance limit exceeded, 0x%X, channel: %X	Runtime measurement, tolerance exceeded
0x8307	Error	Encoder	Delay time, maximum value exceeded, 0x%X, channel: %X	Runtime measurement, maximum value exceeded
0x8308	Error	Encoder	Unsupported ordering designation, 0x%X, channel: %X (only 02 and 22 is supported)	Wrong EnDat order ID
0x8309	Error	Encoder	Encoder CRC error, channel: %X	Encoder CRC error
0x830A	Error	Encoder	Temperature %X could not be read, channel: %X	Temperature cannot be read
0x830C	Error	Encoder	Encoder Single-Cycle-Data Error, channel. %X	CRC error detected. Check the transmission path and the CRC polynomial
0x830D	Error	Encoder	Encoder Watchdog Error, channel. %X	The sensor has not responded within a predefined time period
0x8310	Error	Encoder	Initialisation error	
0x8311	Error	Encoder	Maximum frequency of the input signal is exceeded (channel %d)	
0x8312	Error	Encoder	Encoder plausibility error (channel %d)	
0x8313	Error	Encoder	Configuration error (channel %d)	
0x8314	Error	Encoder	Synchronisation error	
0x8315	Error	Encoder	Error status input (channel %d)	
0x8400	Error	Drive	Incorrect drive configuration: 0x%X, %d	Drive incorrectly configured
0x8401	Error	Drive	Limiting of calibration velocity: %d, %d	Limitation of the calibration velocity
0x8402	Error	Drive	Emergency stop activated: 0x%X, %d	Emergency stop activated
0x8403	Error	Drive	ADC Error	Error during current measurement in the ADC
0x8404	Error	Drive	Overcurrent	Overcurrent in phase U, V or W
0x8405	Error	Drive	Invalid modulo position: %d	Modulo position invalid
0x8406	Error	Drive	DC-Link undervoltage (Error)	The DC link voltage of the terminal is lower than the parameterized minimum voltage. Activation of the output stage is prevented.
0x8407	Error	Drive	DC-Link overvoltage (Error)	The DC link voltage of the terminal is higher than the parameterized maximum voltage. Activation of the output stage is prevented.
0x8408	Error	Drive	I2T-Model Amplifier overload (Error)	• The amplifier is being operated outside the specification. • The I2T-model of the amplifier is incorrectly parameterized.
0x8409	Error	Drive	I2T-Model motor overload (Error)	• The motor is being operated outside the parameterized rated values. • The I2T-model of the motor is incorrectly parameterized.
0x840A	Error	Drive	Overall current threshold exceeded	Total current exceeded
0x8415	Error	Drive	Invalid modulo factor: %d	Modulo factor invalid
0x8416	Error	Drive	Motor overtemperature	The internal temperature of the motor exceeds the parameterized error threshold. The motor stops immediately. Activation of the output stage is prevented.
0x8417	Error	Drive	Maximum rotating field velocity exceeded	Rotary field speed exceeds the value specified for dual use (EU 1382/2014).
0x841C	Error	Drive	STO while the axis was enabled	An attempt was made to activate the axis, despite the fact that no voltage is present at the STO input.
0x8550	Error	Inputs	Zero crossing phase %X missing	Zero crossing phase %X missing

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

7.3 Firmware compatibility

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

Note

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

NOTE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 204]. 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!

EL3632			
Hardware (HW)	**Firmware**	**Revision no.**	**Release date**
03 – 12*	04	EL3632-0000-0016	2011/10
	05		2011/11
		EL3632-0000-0017	2012/11
	06		2014/09
	07	EL3632-0000-0018	2014/09
	08	EL3632-0000-0019	2014/09
	09		2015/05
	10		2015/08
	11	EL3632-0000-0020	2016/03
	12	EL3632-0000-0021	2016/05
		EL3632-0000-0022	2016/05
	13*		2017/07
		EL3632-0000-0023	2018/09

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

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

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

NOTE

Only use TwinCAT 3 software!

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

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

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

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

Storage locations

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

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

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. 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

NOTE

Risk of damage to the device!

✓ Note the following when downloading new device files

a) Firmware downloads to an EtherCAT device must not be interrupted

b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.

c) The power supply must adequately dimensioned. The signal level must meet the specification.

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

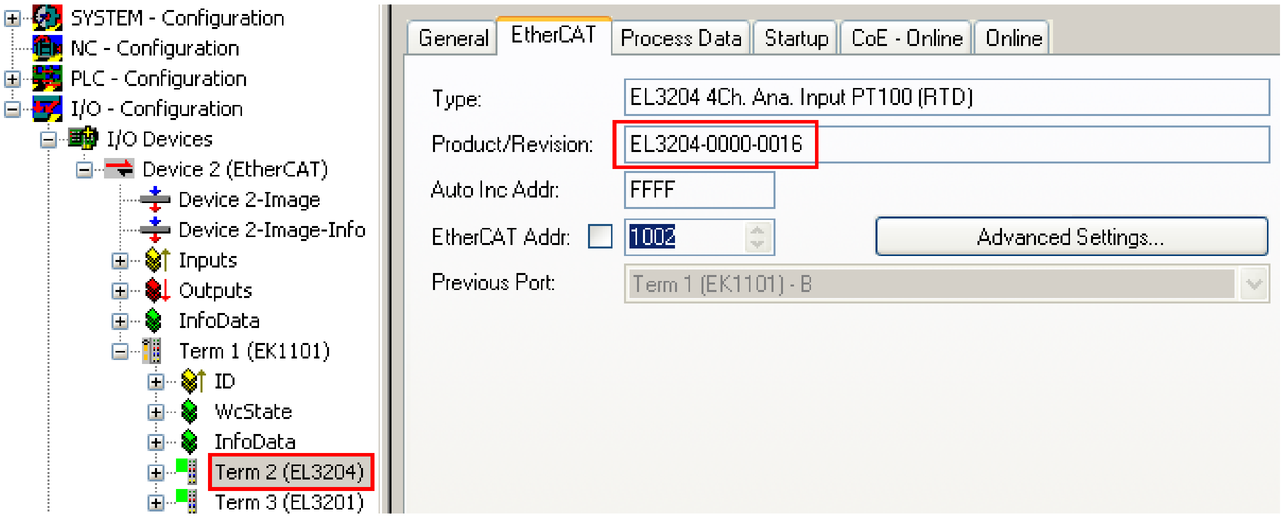
7.4.1 Device description ESI file/XML

NOTE

Attention regarding update of the ESI description/EEPROM

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

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



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. 189: 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:

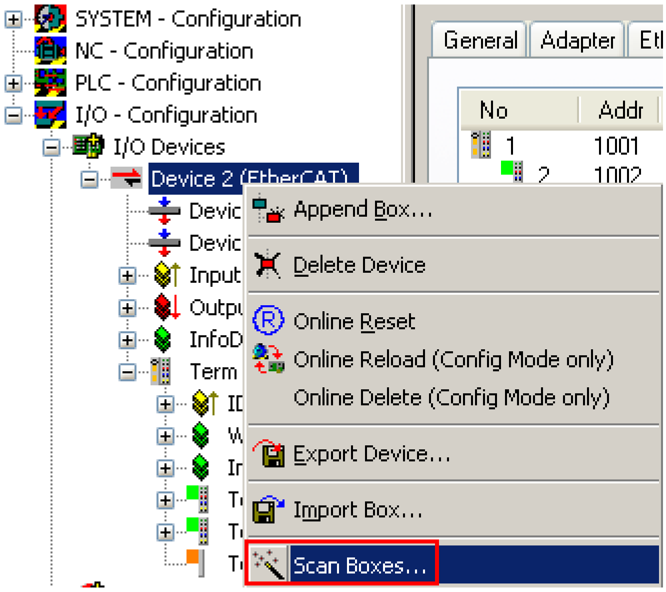
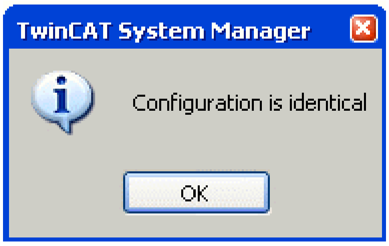


Fig. 190: Scan the subordinate field by right-clicking on the EtherCAT device

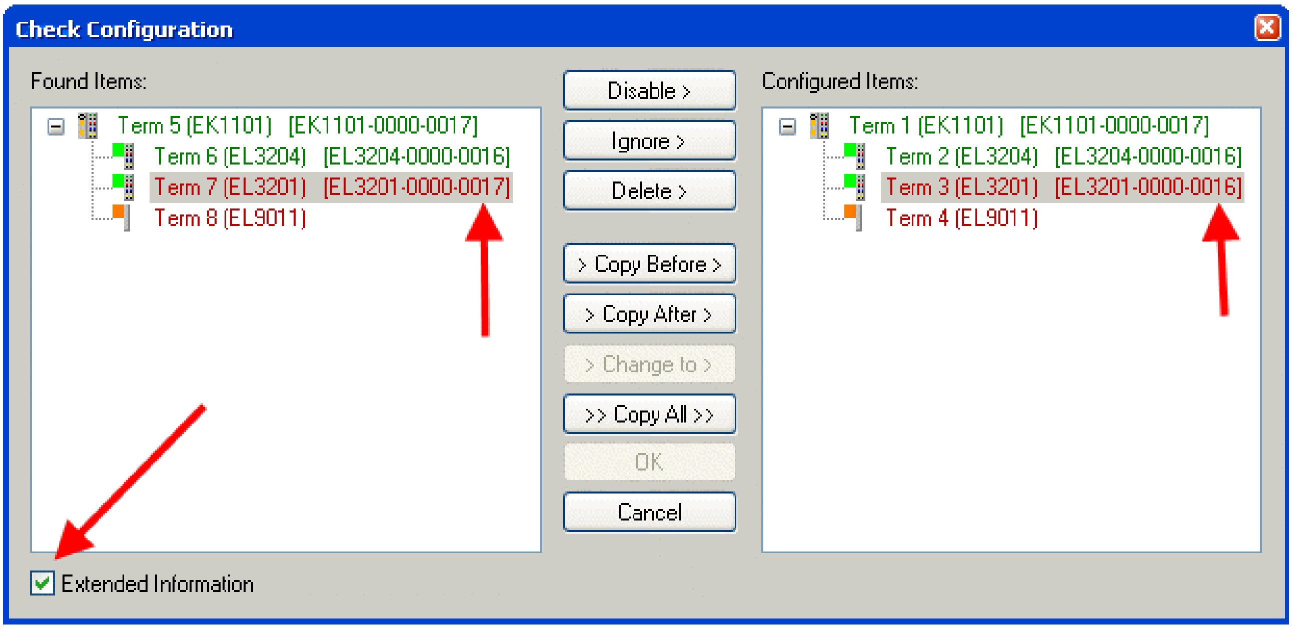
If the found field matches the configured field, the display shows


TwinCAT System Manager

Configuration is identical

Fig. 191: 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)

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

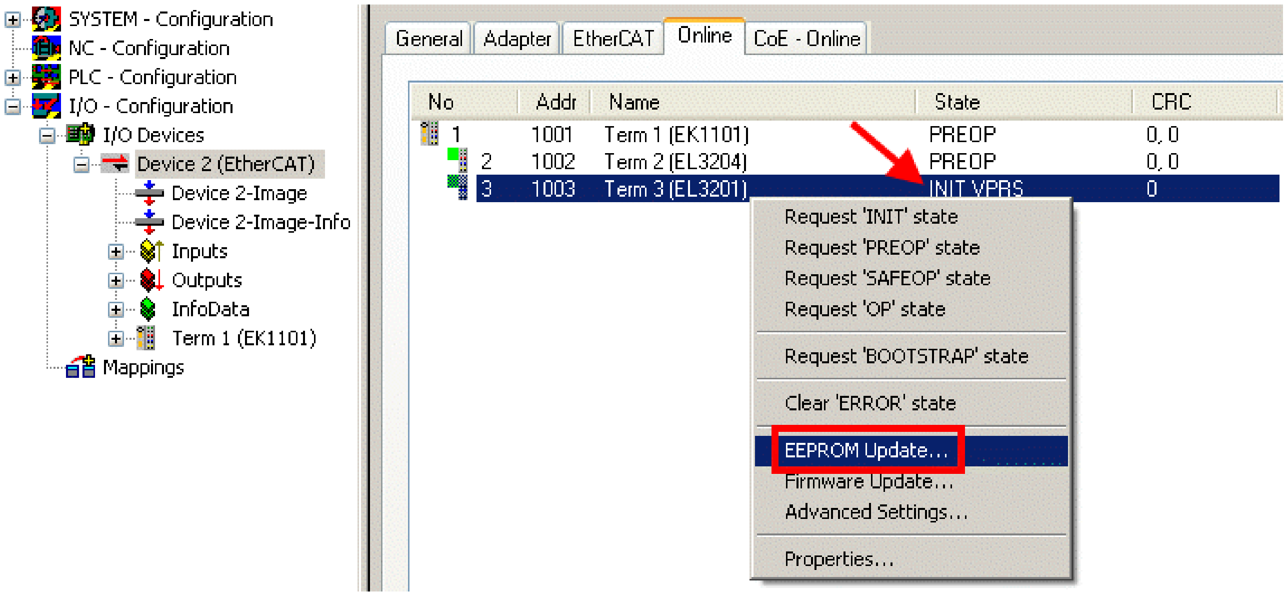
Fig. 192: Change dialog

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. 193: 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.

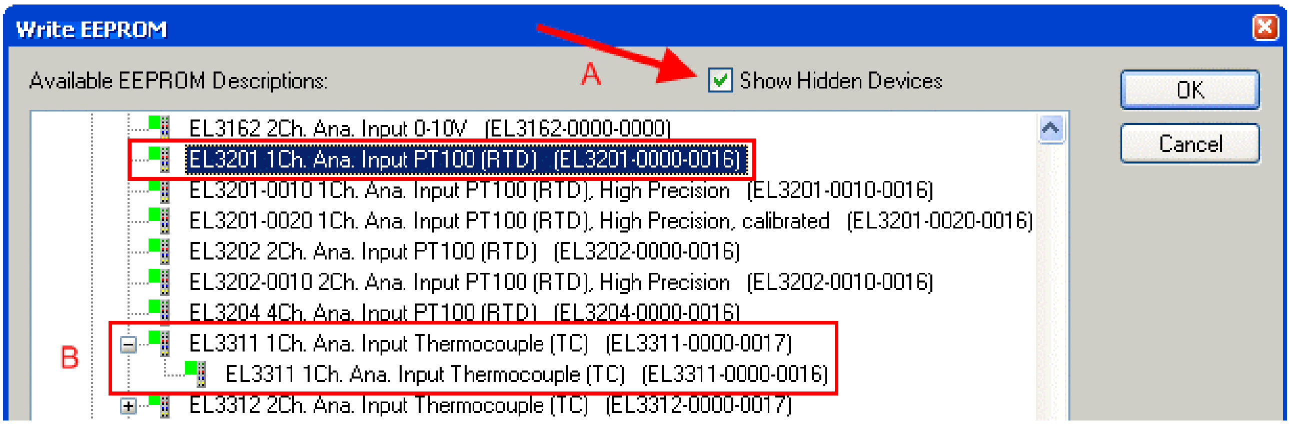

Write EEPROM
Available EEPROM Descriptions:
A
☑ Show Hidden Devices
OK
Cancel
B
EL3162 2Ch. Ana. Input 0-10V (EL3162-0000-0000)
EL3201 1Ch. Ana. Input PT100 (RTD) (EL3201-0000-0016)
EL3201-0010 1Ch. Ana. Input PT100 (RTD), High Precision (EL3201-0010-0016)
EL3201-0020 1Ch. Ana. Input PT100 (RTD), High 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. 194: 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.

7.4.2 Firmware explanation

Determining the firmware version

Determining the version via the System Manager

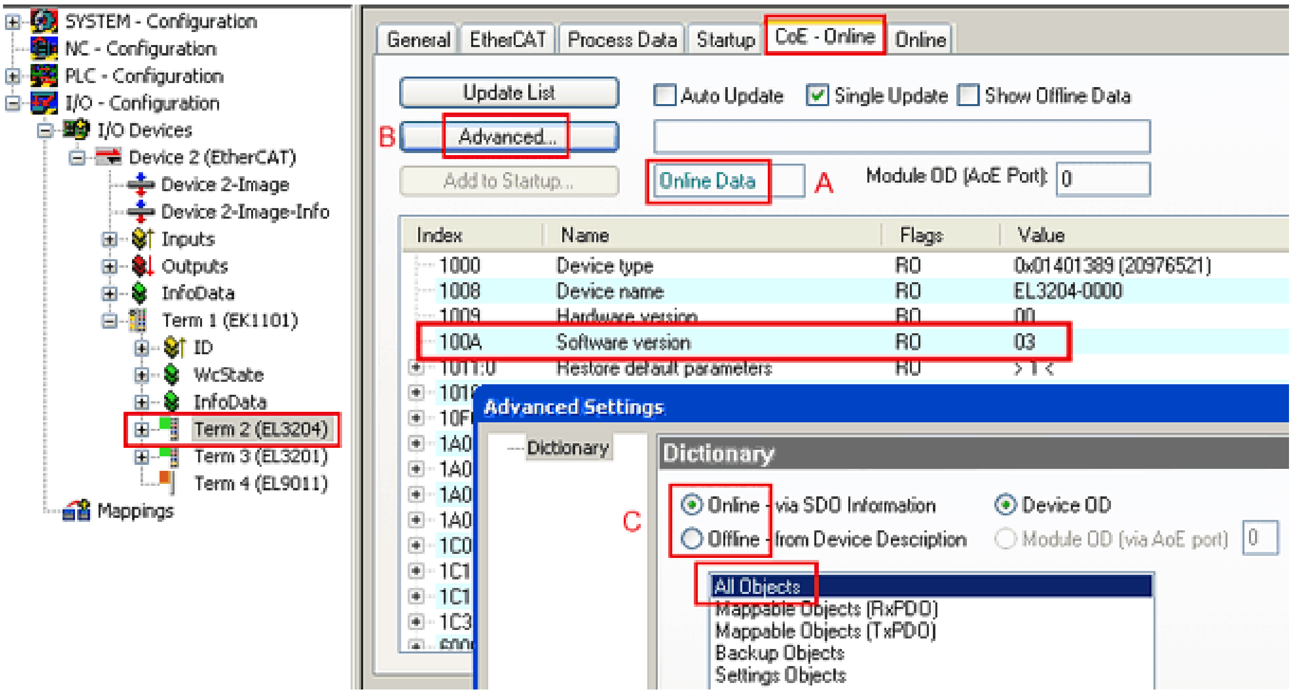
The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

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			

Dictionary

● Online - via SDO Information ● Device OD
○ Offline - from Device Description ○ Module OD (via AoE port) 0

Fig. 195: Display of EL3204 firmware version

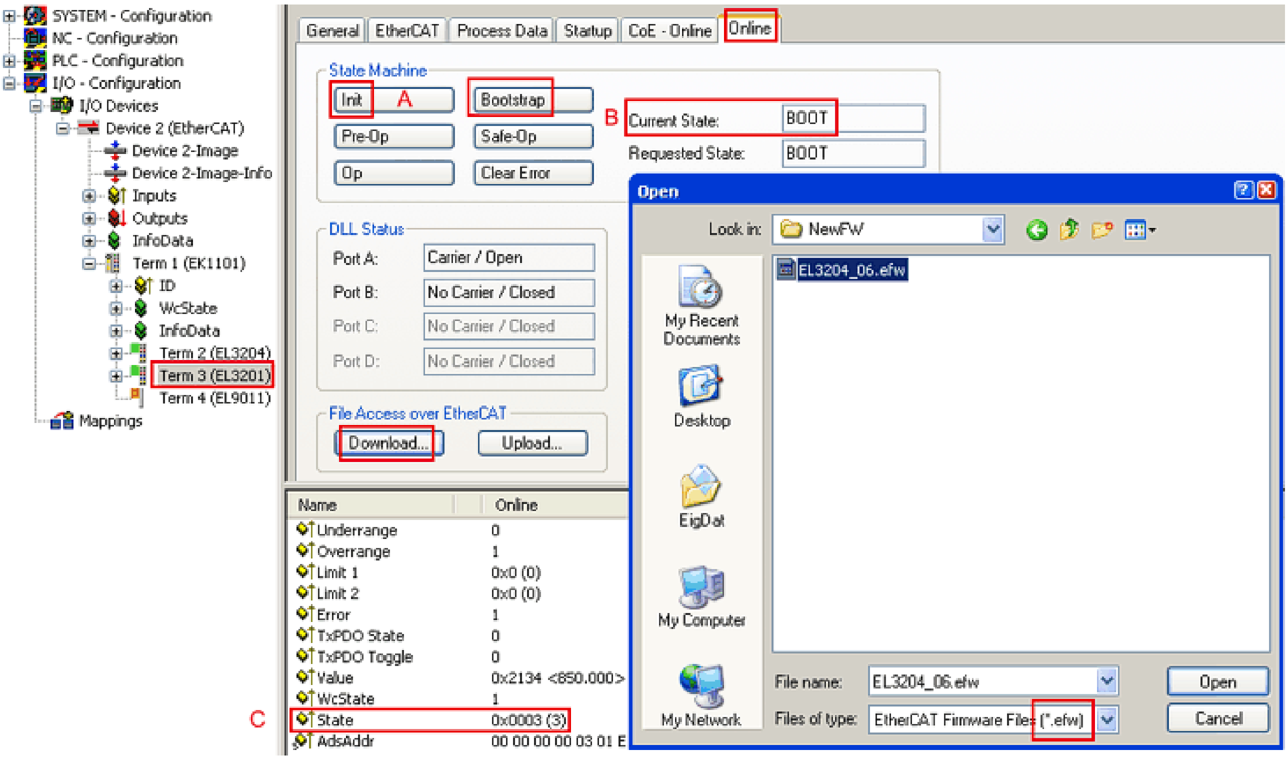
In (A) TwinCAT 2.11 shows that the Online CoE directory is currently displayed. If this is not the case, the Online directory can be loaded via the *Online* option in Advanced Settings (B) and double-clicking on *AllObjects*.

7.4.3 Updating controller firmware *.efw

CoE directory

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.*



SYSTEM - Configuration
NC - Configuration
PLC - Configuration
I/O - Configuration
I/O Devices
Device 2 (EtherCAT)
Device 2-Image
Device 2-Image-Info
Inputs
Outputs
InfoData
Term 1 (EK1101)
ID
WcState
InfoData
Term 2 (EL3204)
Term 3 (EL3201)
Term 4 (EL9011)
Mappings

Current State: BOOT
Requested State: BOOT

Port A: Carrier / Open
Port B: No Carrier / Closed
Port C: No Carrier / Closed
Port D: No Carrier / Closed

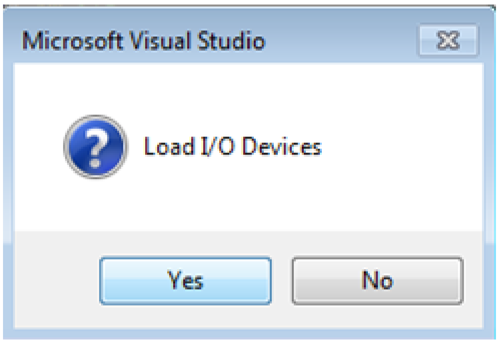
Name	Online
Underrange	0
Overrange	1
Limit 1	0x0 (0)
Limit 2	0x0 (0)
Error	1
TxPDO State	0
TxPDO Toggle	0
Value	0x2134 <850.000>
WcState	1
State	0x0003 (3)
AdsAddr	00 00 00 00 03 01 E

File name: EL3204_06.efw
Files of type: EtherCAT Firmware Files (*.efw)

Fig. 196: Firmware Update

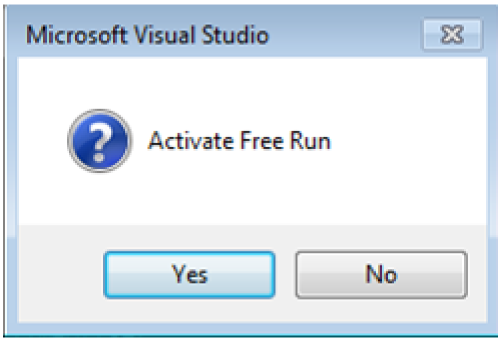
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

- 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

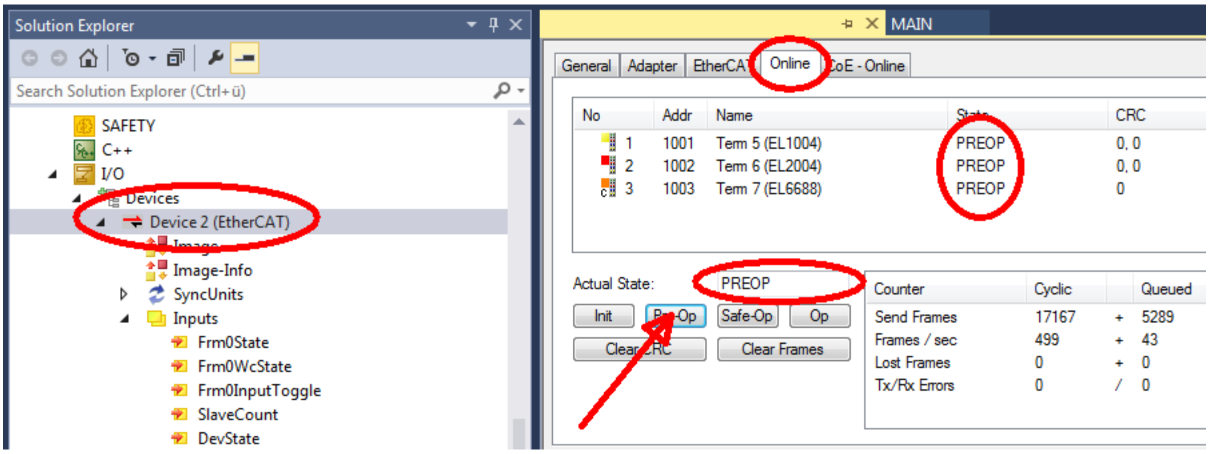
Load I/O Devices



Microsoft Visual Studio

Activate Free Run

- Switch EtherCAT Master to PreOP



SAFETY
C++
I/O
Devices
Device 2 (EtherCAT)
Image
Image-Info
SyncUnits
Inputs
Frm0State
Frm0WcState
Frm0InputToggle
SlaveCount
DevState

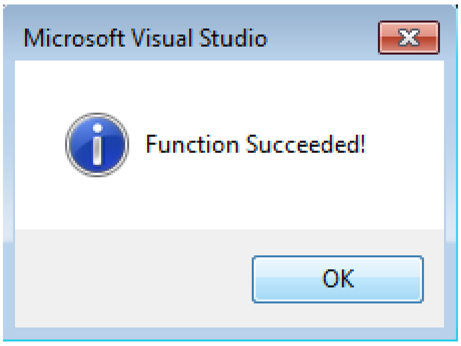
No	Addr	Name	State	CRC
1	1001	Term 5 (EL1004)	PREOP	0, 0
2	1002	Term 6 (EL2004)	PREOP	0, 0
3	1003	Term 7 (EL6688)	PREOP	0

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 pass word will not be neccessary 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.

7.4.4 FPGA firmware *.rbf

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

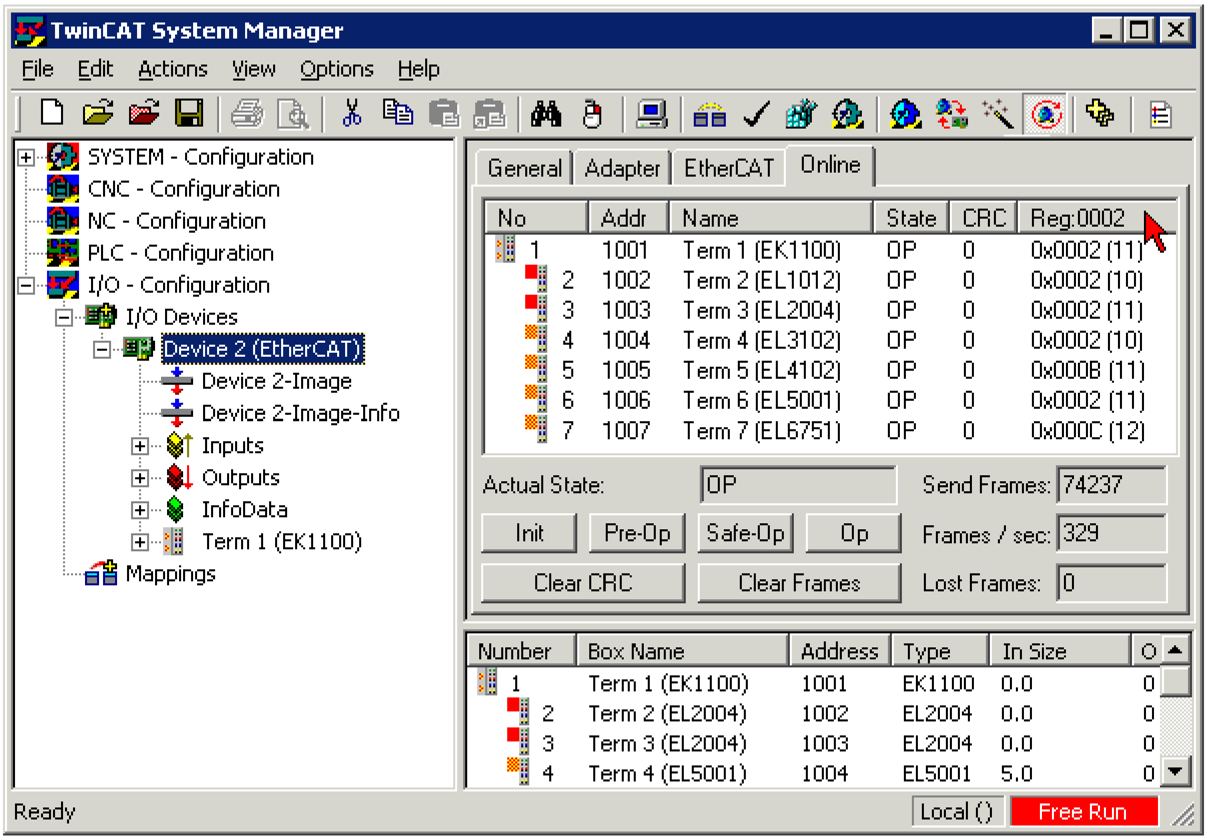
- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

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

Determining the version via the System Manager

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

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



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. 197: FPGA firmware version definition

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

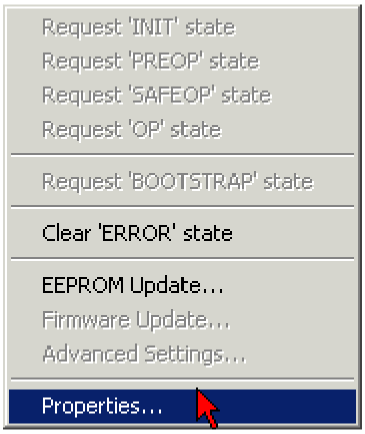
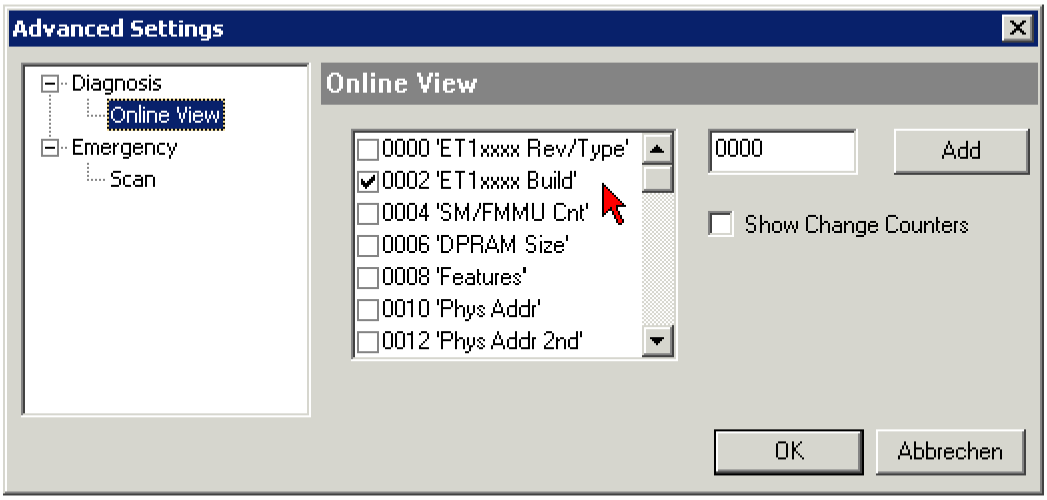


Fig. 198: Context menu *Properties*

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. 199: 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.

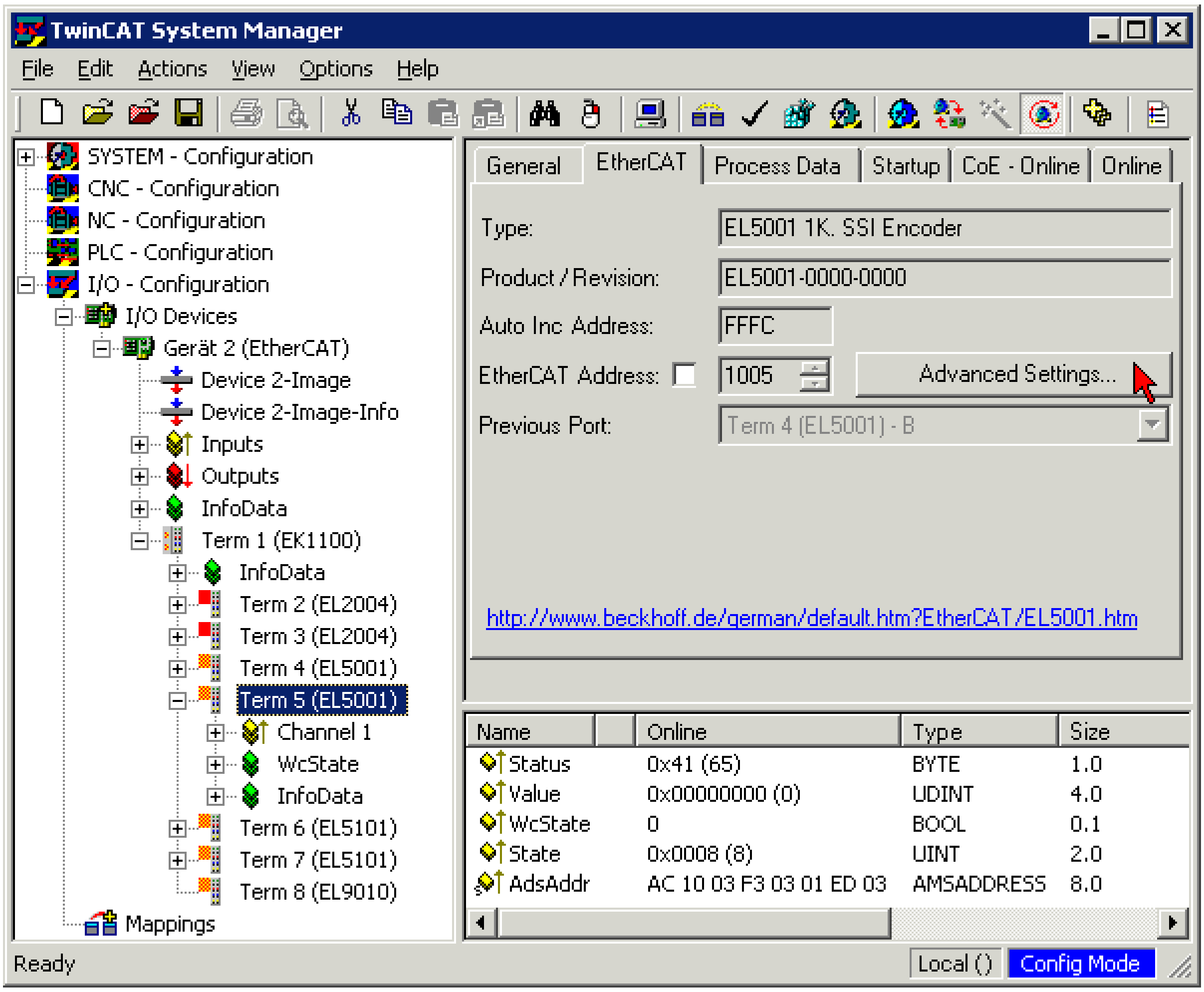
Older firmware versions can only be updated by the manufacturer!

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

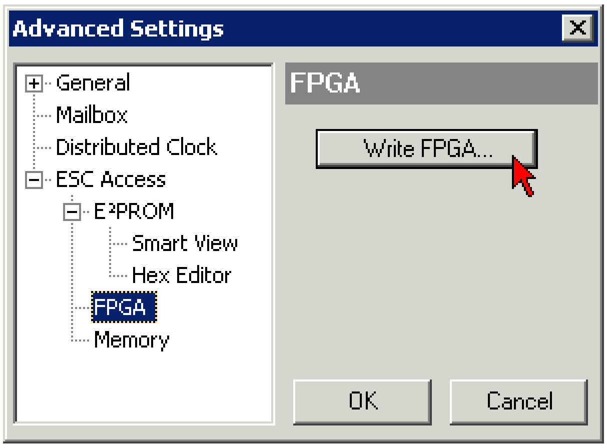
EtherCAT Address: □ 1005

Previous Port: Term 4 (EL5001) - B

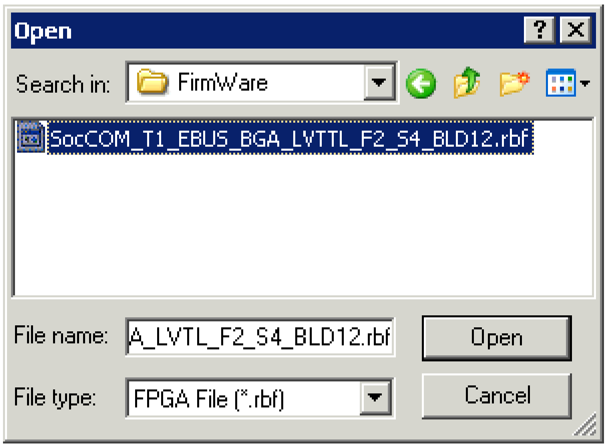
http://www.beckhoff.de/german/default.htm?EtherCAT/EL5001.htm

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:



Search in: FirmWare

SocCOM_T1_EBUS_BGA_LVTTL_F2_S4_BLD12.rbf

File name: A_LVTTL_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

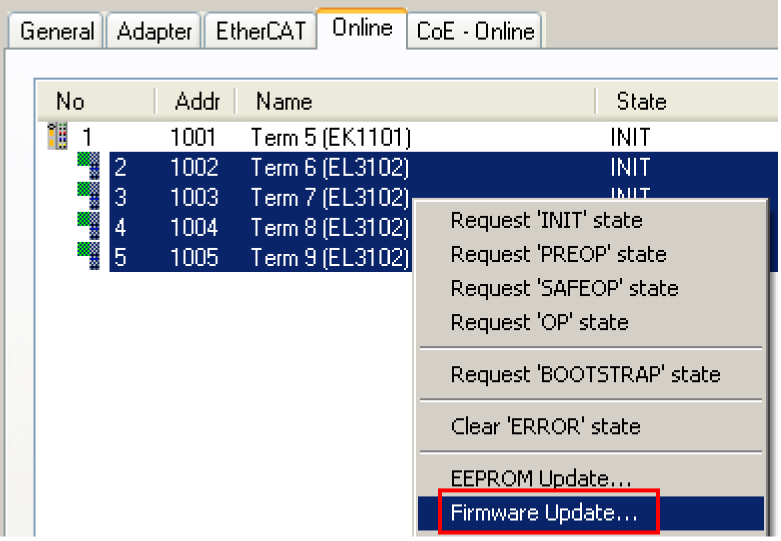
NOTE

Risk of damage to the device!

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

7.4.5 Simultaneous updating of several EtherCAT devices

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



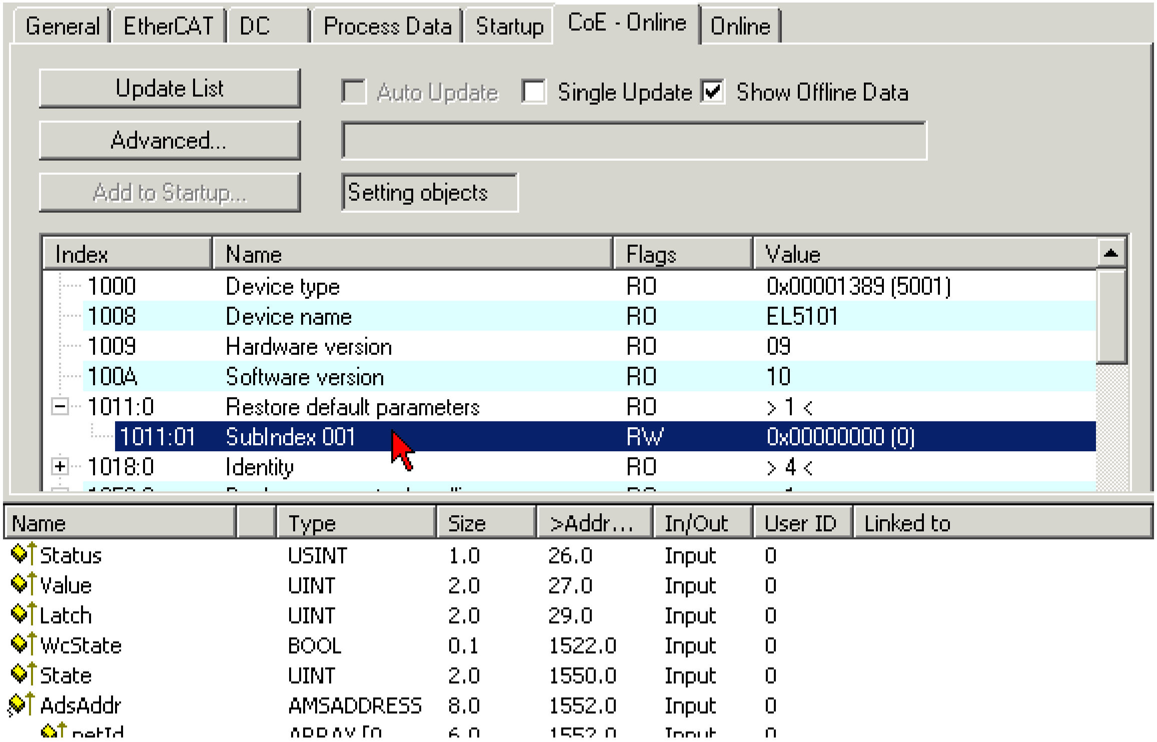
No	Addr	Name	State
1	1001	Term 5 (EK1101)	INIT
2	1002	Term 6 (EL3102)	INIT
3	1003	Term 7 (EL3102)	INIT
4	1004	Term 8 (EL3102)	
5	1005	Term 9 (EL3102)	

Fig. 200: Multiple selection and firmware update

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

7.5 Restoring the delivery state

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



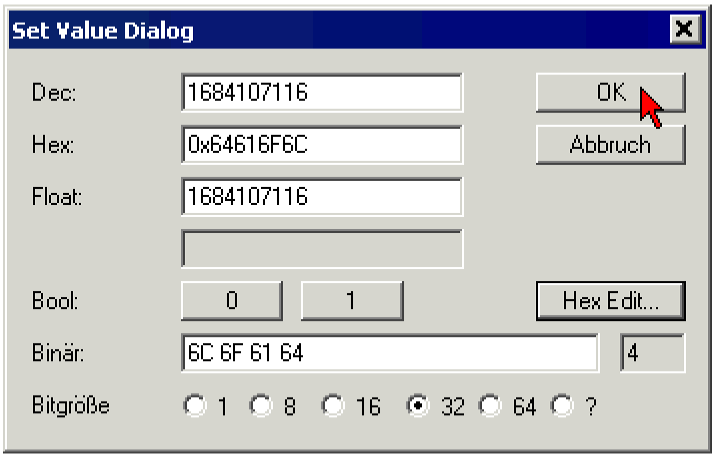
☐ Auto Update ☐ Single Update ☑ Show Offline Data

Index	Name	Flags	Value
1000	Device type	RO	0x00001389 (5001)
1008	Device name	RO	EL5101
1009	Hardware version	RO	09
100A	Software version	RO	10
1011:0	Restore default parameters	RO	> 1 <
1011:01	SubIndex 001	RW	0x00000000 (0)
1018:0	Identity	RO	> 4 <

Name	Type	Size	>Addr...	In/Out	User ID	Linked to
Status	USINT	1.0	26.0	Input	0	
Value	UINT	2.0	27.0	Input	0	
Latch	UINT	2.0	29.0	Input	0	
WcState	BOOL	0.1	1522.0	Input	0	
State	UINT	2.0	1550.0	Input	0	
AdsAddr	AMSADDRESS	8.0	1552.0	Input	0	
netId	ARRAY [0	6.0	1552.0	Input	0	

Fig. 201: Selecting the *Restore default parameters* PDO

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



Dec: 1684107116

Hex: 0x64616F6C

Float: 1684107116

Bool: 0 1

Binär: 6C 6F 61 64 4

Bitgröße ○ 1 ○ 8 ○ 16 ● 32 ○ 64 ○ ?

Fig. 202: Entering a restore value in the Set Value dialog

Alternative restore value

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

7.6 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Beckhoff Support

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

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

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

Beckhoff Service

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

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

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

Beckhoff Headquarters

Beckhoff Automation GmbH & Co. KG

Huelshorstweg 20
33415 Verl
Germany

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

More Information:
www.beckhoff.com/EL3632

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