BECKHOFF New Automation Technology

Short documentation | EN

EPP3504-0023/ ERP3504-0022

4-channel measuring bridge,(SG) full/half/quarter bridge, 24 bit, 10 ksps

2026-05-04 | Version: 1.7

1 Product overview measurement technology box modules

The following box modules were described within this documentation:

Measuring bridge/ SG measurement

EPP3504-0023 [▶ 17] (4-channel analog input, measuring bridge, full/half/quarter bridge, 24 bit, 10 ksps, IP20 connector)

ERP3504-0022 [▶ 17] (4-channel analog input, measuring bridge, full/half/quarter bridge, 24 bit, 10 ksps)

Table of contents

1 Product overview measurement technology box modules 3

2 Foreword 7

2.1 Notes on the documentation 7
2.2 Safety instructions 8
2.3 Documentation Issue Status 9
2.4 Guide through documentation 10
2.5 Version identification of EtherCAT devices 11
2.5.1 General notes on marking 11
2.5.2 Beckhoff Identification Code (BIC) 11
2.5.3 Electronic access to the BIC (eBIC) 13

3 Product group: EtherCAT P Box modules 16

4 Product overview 17

4.1 EPP3504-0023/ ERP3504-0022 - Introduction 17
4.2 EPP3504-0023/ ERP3504-0022 - Technical data 19
4.2.1 EPP3504-0023/ ERP3504-0022 overview measurement ranges 21
4.2.2 Measurement ±10 V 23
4.2.3 Measurement ±80 mV 25
4.2.4 RTD/Pt1000 measurement 27
4.2.5 Potentiometer measurement 34
4.2.6 Measurement SG 1/1 bridge (full bridge) 4/6-wire connection 38
4.2.7 Measurement SG 1/2 bridge (half bridge) 3/5-wire connection 42
4.2.8 Measurement SG 1/4 bridge (quarter-bridge) 2/3-wire connection 47
4.3 Process data interpretation 56
4.4 General Information on Measuring Accuracy/Measurement Uncertainty 57
4.5 Scope of supply 62

5 Commissioning 63

5.1 Notes to short documentation 63
5.2 CoE overview 64
5.2.1 General access to online CoE values 64
5.2.2 0x10E2 Manufacturer-specific Identification Code 69
5.2.3 0x10F3 Diagnosis History 69
5.2.4 0x60n0 PAI Status Ch.[n+1] 69
5.2.5 0x60n1 PAI Samples Ch.[n+1] (24 Bit) 70
5.2.6 0x60n2 PAI Samples Ch.[n+1] (16 Bit) 70
5.2.7 0x60n3 PAI Samples Ch.[n+1] (REAL32) 70
5.2.8 0x60n5 PAI Timestamp Ch.[n+1] 70
5.2.9 0x60n6 PAI Synchronous Oversampling Ch.[n+1] 71
5.2.10 0x70n0 PAI Control Ch.[n+1] 71
5.2.11 0x80n0 PAI Settings Ch.[n+1] 71
5.2.12 0x80n1 PAI Filter 1 Settings Ch.[n+1] 74
5.2.13 0x80n3 PAI Filter 2 Settings Ch.[n+1] 74
5.2.14 0x80n6 PAI Scaler Settings Ch.[n+1] 74
5.2.15 0x80nA PAI Extended Settings Ch.[n+1] 75

5.2.16 0x80nE PAI User Calibration Data Ch.[n+1] ... 75
5.2.17 0x80nF PAI Vendor Calibration Data Ch.[n+1] ... 76
5.2.18 0x90n0 PAI Internal Data Ch.[n+1] ... 77
5.2.19 0x90n2 PAI Info Data Ch.[n+1] ... 77
5.2.20 0x90nF PAI Calibration Dates Ch.[n+1] ... 79
5.2.21 0xF000 Modular device profile ... 84
5.2.22 0xF008 Code word ... 84
5.2.23 0xF009 Password protection ... 84
5.2.24 0xF010 Module list ... 85
5.2.25 0xF083 BTN ... 85
5.2.26 0xF900 PAI Info Data ... 85
5.2.27 0xF912 Filter info ... 85
5.2.28 0xFB00 PAI Command ... 86
5.2.29 0x80n0:01 PAI Settings.Interface ... 87
5.3 Sample programs ... 88
5.3.1 Sample program 1 and 2 (offset/gain) ... 89
5.3.2 Sample program 3 (write LookUp table) ... 94
5.3.3 Sample program 4 (generate LookUp table) ... 95
5.3.4 Sample program 5 (write filter coefficients) ... 96
5.3.5 Sample program 6 (interlacing of measured values) ... 98
5.3.6 Sample program 7 (general decimation in the PLC) ... 102
5.3.7 Sample program 8 (diagnosis messages) ... 107
5.3.8 Sample program 9 (measuring range combination) ... 107
5.3.9 Sample program 10 (FB for real time diagnosis) ... 111
5.3.10 Sample program 11 (scripts for generation and transformation of filter coefficients) ... 114
5.3.11 Sample program 12 (R/W signature of calibration) ... 114
5.3.12 Sample program 13: Reading the BIC from the CoE ... 116

6 Features ... 118

7 Commissioning on EtherCAT Master ... 119
7.1 General Commissioning Instructions for an EtherCAT Slave ... 119
7.2 TwinCAT Quick Start ... 127
7.2.1 TwinCAT 2 ... 129
7.2.2 TwinCAT 3 ... 139
7.3 TwinCAT Development Environment ... 153
7.3.1 Installation of the TwinCAT real-time driver ... 153
7.3.2 Notes regarding ESI device description ... 159
7.3.3 TwinCAT ESI Updater ... 163
7.3.4 Distinction between Online and Offline ... 165
7.3.5 OFFLINE configuration creation ... 165
7.3.6 ONLINE configuration creation ... 171
7.3.7 EtherCAT subscriber configuration ... 179
7.3.8 Import/Export of EtherCAT devices with SCI and XTI ... 189
7.4 EtherCAT basics ... 196
7.5 EtherCAT cabling – wire-bound ... 197
7.6 General notes for setting the watchdog ... 199

7.7 EtherCAT State Machine ... 201
7.8 CoE Interface ... 202
7.9 Distributed Clock ... 208

8 Mounting and wiring ... 209
8.1 Mounting ... 209
8.1.1 Dimensions ... 209
8.1.2 Fixing ... 210
8.1.3 Functional earth (FE) ... 211
8.1.4 Tightening torques for plug connectors ... 212
8.2 Notes regarding connectors and wiring ... 213
8.3 Note - power supply ... 215
8.4 Notes on connection technology EPP3504/ ERP3504 ... 216
8.5 Connectors ... 217
8.6 Accessories ... 218
8.7 Power supply, potential groups ... 219
8.8 Status LEDs ... 221
8.8.1 RUN, Channel 1 ... Channel 4 ... 221
8.8.2 Supply voltage ... 222
8.8.3 EtherCAT ... 222
8.9 Conductor losses ... 223
8.10 Disposal ... 224

9 Appendix ... 225
9.1 Volatility ... 225
9.2 Diagnostics - basic principles of diag messages ... 226
9.3 TcEventLogger and IO ... 233
9.4 EtherCAT AL Status Codes ... 237
9.5 Firmware Update ... 237
9.5.1 Device description ESI file/XML ... 239
9.5.2 Firmware explanation ... 242
9.5.3 Updating controller firmware *.efw ... 242
9.5.4 FPGA firmware *.rbf ... 244
9.5.5 Simultaneous updating of several EtherCAT devices ... 248
9.6 Firmware compatibility ... 249
9.7 Restoring the delivery state ... 250
9.8 Notes on operation ... 251
9.9 Further documentation for I/O components with analog in and outputs ... 252
9.10 Support and Service ... 253
9.11 Reshipment and return ... 253

2 Foreword

NOTICE

Within this short documentation some chapters are only available in a shortened version. For the complete documentation please contact the Beckhoff sales department responsible for you.

2.1 Notes on the documentation

Intended audience

This description is only intended for the use of trained specialists in control and automation engineering who are familiar with the applicable national standards.
It is essential that the documentation and the following notes and explanations are followed when installing and commissioning these components.
The qualified personnel is obliged to always use the currently valid documentation.

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®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered trademarks of and licensed by Beckhoff Automation GmbH.

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

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.

Third-party brands

Trademarks of third parties may be used in this documentation. You can find the trademark notices here:
https://www.beckhoff.com/trademarks

2.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Signal words

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

Personal injury warnings

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

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

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

Warning of damage to property or environment

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

Information on handling the product

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

2.3 Documentation Issue Status

Version	Comment
1.7	• Update to “EPP3504-0023/ ERP3504-0022 - Technical data“: ◦ Basic mechanical properties ◦ Specific data regarding the measurement ranges • Update to subchapter “Appendix“/ “Firmware compatibility“

Firmware and hardware versions

This documentation refers to the firmware and hardware version that was applicable at the time the documentation was written.

The module features are continuously improved and developed further. Modules having earlier production statuses cannot have the same properties as modules with the latest status. However, existing properties are retained and are not changed, so that older modules can always be replaced with new ones.

The firmware and hardware version (delivery state) can be found in the batch number (D-number) printed on the side of the EtherCAT Box.

Syntax of the batch number (D-number)

D: WW YY FF HH	Example with D no. 29 10 02 01:
WW - week of production (calendar week)	29 - week of production 29
YY - year of production	10 - year of production 2010
FF - firmware version	02 - firmware version 02
HH - hardware version	01 - hardware version 01

Also see about this

Firmware compatibility [▶ 249]

2.4 Guide through documentation

NOTICE

Further components of documentation

This documentation describes device-specific content. It is part of the modular documentation concept for Beckhoff I/O components. For the use and safe operation of the device / devices described in this documentation, additional cross-product descriptions are required, which can be found in the following table.

Title	Description
EtherCAT System Documentation (PDF)	• System overview • EtherCAT basics • Cable redundancy • Hot Connect • EtherCAT devices configuration
I/O Configuration in TwinCAT (PDF)	Quick start guide for EtherCAT Box Modules and EtherCAT P Box modules
I/O Analog Manual (PDF)	Notes on I/O components with analog in and outputs
Infrastructure for EtherCAT/Ethernet (PDF)	Technical recommendations and notes for design, implementation and testing
Software Declarations I/O (PDF)	Open source software declarations for Beckhoff I/O components

The documentations can be viewed at and downloaded from the Beckhoff website (www.beckhoff.com) via:

- the “Documentation and Download” area of the respective product page,
- the Download finder,
- the Beckhoff Information System.

If you have any suggestions or proposals for our documentation, please send us an e-mail stating the documentation title and version number to: documentation@beckhoff.com

2.5 Version identification of EtherCAT devices

2.5.1 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. *“EL2872 with revision 0022 and serial number 01200815”*.
- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

2.5.2 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.

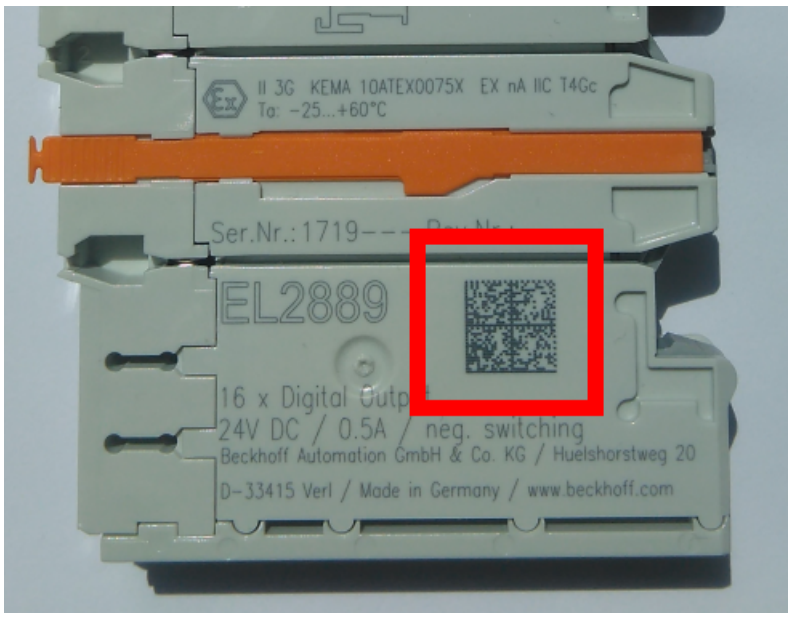


Fig. 1: BIC as data matrix code (DMC, code scheme ECC200)

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:

Position	Type of information	Explanation	Data identifier	Number of digits incl. data identifier	Example
1	Beckhoff order number	**Beckhoff order number**	1P	8	**1P**072222
2	Beckhoff Traceability Number (BTN)	**Unique serial number, see note below**	SBTN	12	**SBTN**k4p562d7
3	Article description	**Beckhoff article description, e.g. EL1008**	1K	32	**1K**EL1809
4	Quantity	**Quantity in packaging unit, e.g. 1, 10, etc.**	Q	6	**Q**1
5	Batch number	Optional: Year and week of production	2P	14	**2P**401503180016
6	ID/serial number	Optional: Present-day serial number system, e.g. with safety products	51S	12	**51S**678294
7	Variant number	Optional: Product variant number on the basis of standard products	30P	12	**30P**F971, 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**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

Accordingly as DMC:

Fig. 2: Example DMC **1P**072222**SBTN**k4p562d7**1K**EL1809 **Q**1 **51S**678294

BTN

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

NOTICE

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

2.5.3 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.

The interface that the product can be electronically addressed by is crucial for the electronic readout.

K-bus devices (IP20, IP67)

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

EtherCAT devices (IP20, IP67)

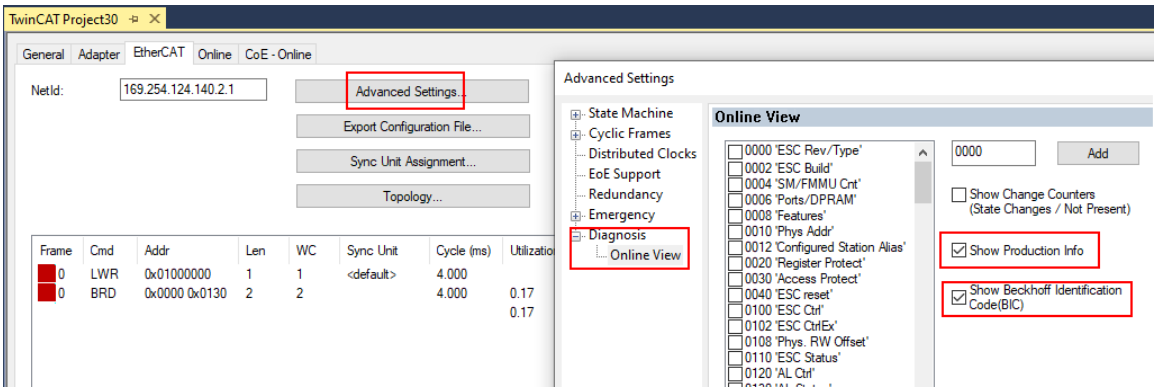
All Beckhoff EtherCAT devices have an ESI-EEPROM which contains the EtherCAT identity with the revision number. The EtherCAT slave information, also colloquially known as the ESI/XML configuration file for the EtherCAT master, is stored in it. See the corresponding chapter in the EtherCAT system manual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

Beckhoff also stores the eBIC in the ESI-EEPROM. The eBIC was introduced into Beckhoff IO production (terminals, box modules) in 2020; as of 2023, implementation is largely complete.

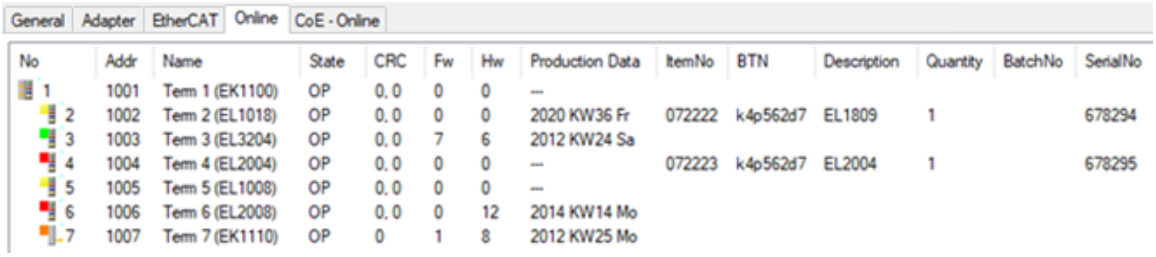
The user can electronically access the eBIC (if present) as follows:

- With all EtherCAT devices, the EtherCAT master (TwinCAT) can read the eBIC from the ESI-EEPROM
  - From TwinCAT 3.1 build 4024.11, the eBIC can be displayed in the online view.

- To do this, check the "Show Beckhoff Identification Code (BIC)" checkbox under EtherCAT → Advanced Settings → Diagnostics:

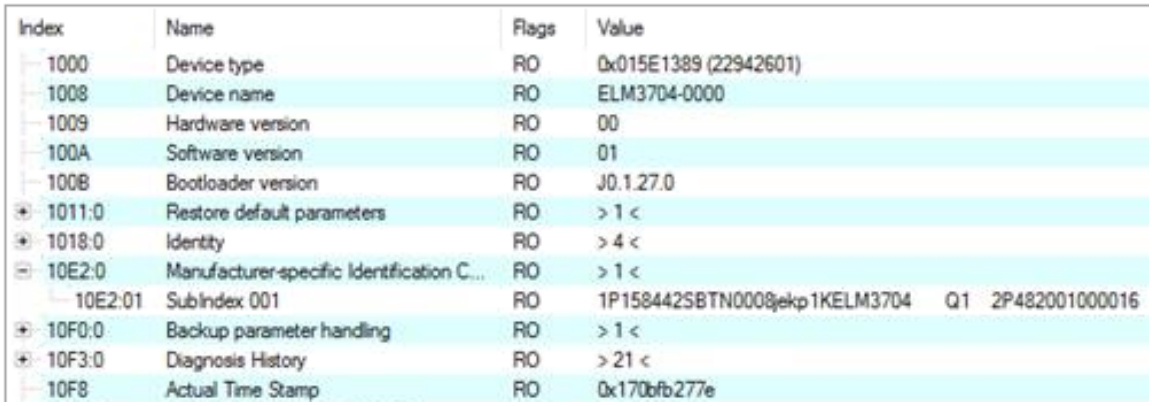


- The BTN and its contents are then displayed:



- Note: As shown in the figure, the production data HW version, FW version, and production date, which have been programmed since 2012, can also be displayed with "Show production info".
- Access from the PLC: From TwinCAT 3.1. build 4024.24, the functions *FB_EcReadBIC* and *FB_EcReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0.

- EtherCAT devices with a CoE directory may also have the object 0x10E2:01 to display their own eBIC, which can also be easily accessed by the PLC:
  - The device must be in PREOP/SAFEOP/OP for access:



  - The object 0x10E2 will be preferentially introduced into stock products in the course of necessary firmware revision.
  - From TwinCAT 3.1. build 4024.24, the functions *FB_EcCoEReadBIC* and *FB_EcCoEReadBTN* for reading into the PLC are available in the Tc2_EtherCAT library from v3.3.19.0

- The following auxiliary functions are available for processing the BIC/BTN data in the PLC in *Tc2_Utilities* as of TwinCAT 3.1 build 4024.24
  - F_SplitBIC: The function splits the Beckhoff Identification Code (BIC) sBICValue into its components using known identifiers and returns the recognized substrings in the ST_SplittedBIC structure as a return value
  - BIC_TO_BTN: The function extracts the BTN from the BIC and returns it as a return value
- Note: If there is further electronic 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 written as an additional category in the ESI-EEPROM during device production. The structure of the ESI content is largely dictated by the ETG specifications, therefore the additional vendor-specific content is stored using a category in accordance with the ETG.2010. ID 03 tells all EtherCAT masters that they may not overwrite these data in the event of an update or restore the data after an ESI update.
  The structure follows the content of the BIC, see here. The EEPROM therefore requires approx. 50..200 bytes of memory.
- 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 which each have 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, and DeviceNet® devices

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

3 Product group: EtherCAT P Box modules

EtherCAT P

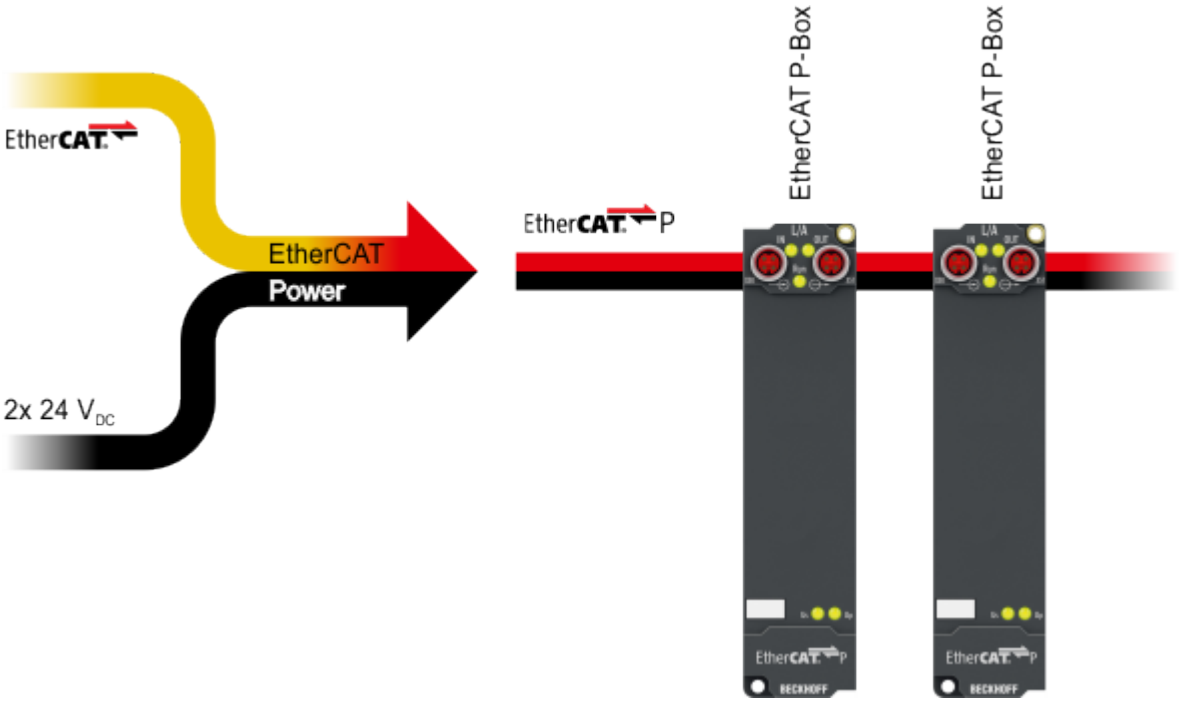
EtherCAT P supplements the EtherCAT technology with a process in which communication and supply voltages are transmitted on a common line. All EtherCAT properties are retained with this process.

Two supply voltages are transmitted per EtherCAT P line. The supply voltages are electrically isolated from each other and can therefore be switched individually. The nominal supply voltage for both is 24 V_{DC}.

EtherCAT P uses the same cable structure as EtherCAT: a 4-core Ethernet cable with M8 connectors. The connectors are mechanically coded so that EtherCAT connectors and EtherCAT P connectors cannot be interchanged.

EtherCAT P Box modules

EtherCAT P Box modules are EtherCAT P slaves with degree of protection IP67. They are designed for operation in wet, dirty or dusty industrial environments.



EtherCAT basics

A detailed description of the EtherCAT system can be found in the EtherCAT system documentation.

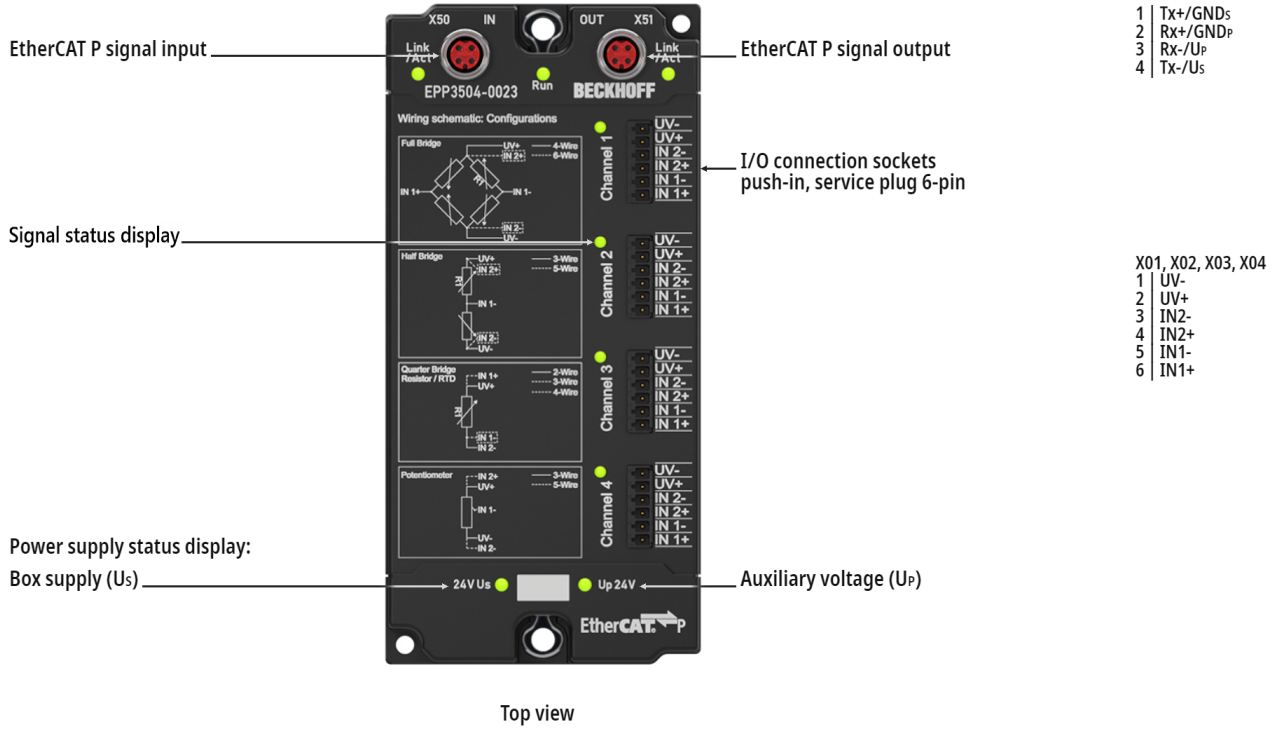
4 Product overview

NOTICE

Documentation status of functions

The EPP3504/ ERP3504 devices are in continuous development, both in terms of new devices and new software functions. As a result, settings that can be found in delivered devices, e.g. in the CoE/PDO/DC dialogs, may not yet be documented. The use of such non-documented properties is not permitted unless it is done with the express supervision of Beckhoff Support.

4.1 EPP3504-0023/ ERP3504-0022 - Introduction



Top view

Fig. 3: EPP3504-0023

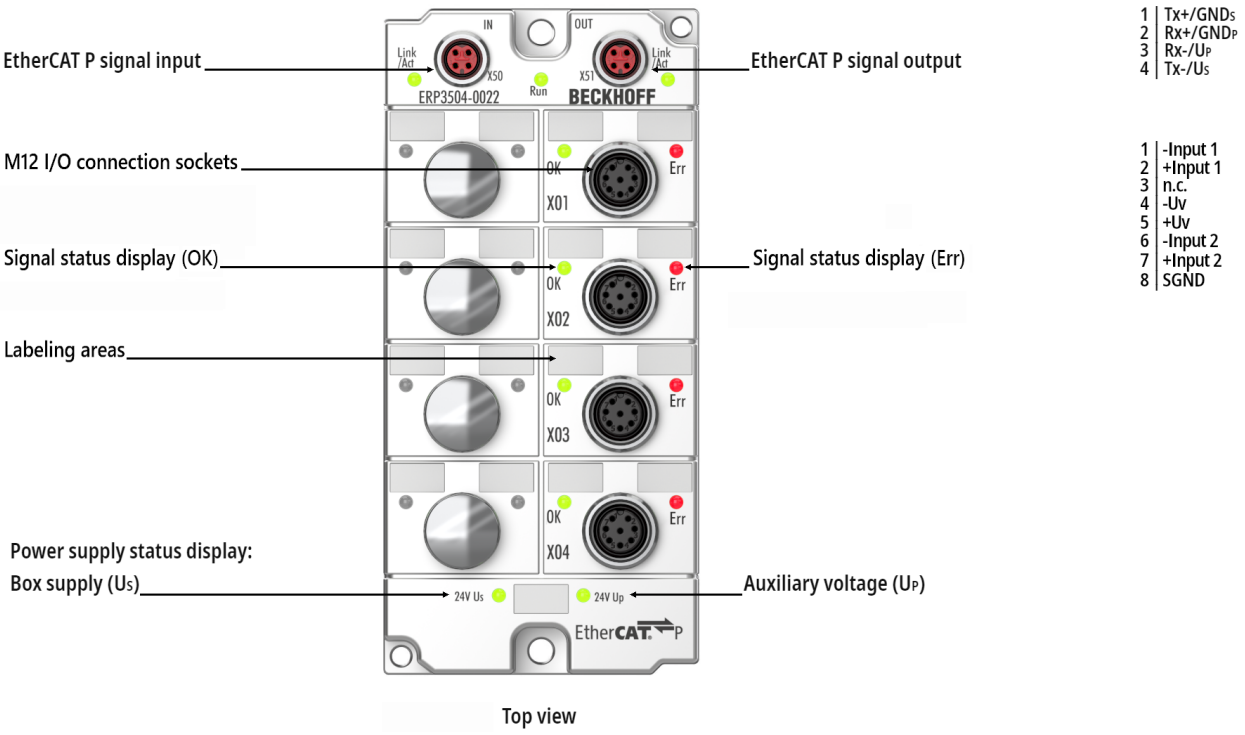


Fig. 4: ERP3504-0022

4-channel measuring bridge,(SG) full/half/quarter bridge, 24 bit, 10 ksps

The EPP3504-0023 EtherCAT P Box is designed for the evaluation of measuring bridges in full-bridge, half-bridge and quarter-bridge configuration. In addition, it supports the measuring ranges potentiometer, PT1000 (RTD) and ±10 V. It is derived from the ELM3504 EtherCAT Terminal and also features a sampling rate of 10,000 sps, internal switchable supplementary resistors and an integrated, parameterizable bridge supply.

It has the same technological features as the ELM3x0x Terminals, i.e. all parameters can be set via EtherCAT using the CoE directory.

The EPP3504-0023 and ERP3504-0022 are designed to be used close to the measurement site.

The EPP3504-0023 is intended for use in a protected environment and is therefore equipped with IP20 bridge connections.

The ERP3504-0022 is an IP67 product that has M12 plugs instead of push-in spring-loaded terminals and is housed in a zinc die-cast housing and is therefore protected against interference. Otherwise, this box is identical to the EPP3504-0023 in terms of its content and functions.

Quick-Links

- IO configuration in TwinCAT
- Mounting and wiring [▶ 209]
- Process data overview [▶ 63]
- Connection view [▶ 63]
- Object description and parameterization [▶ 64]

4.2 EPP3504-0023/ ERP3504-0022 - Technical data

Technical data	EPP3504-0023	ERP3504-0022
Analog inputs	4 channel (differential)	
Time relation between channels to each other	Simultaneous conversion of all channels in the box, synchronous conversion between devices, if DistributedClocks will be used	
ADC conversion method	ΔΣ (Delta-Sigma) with internal sample rate	
	8 msps	
Cutoff frequency input filtering hardware (see explanations within chapter Features/ Firmware filter concept [▶ 118])	Before AD converter: Hardware low pass -3 dB @ 30 kHz type butterworth 3th order Within ADC after conversion:	
	Low pass -3 dB @ 2.6 kHz, ramp-up time 300 µs	
	type sinc3/average filter *The ramp-up time/ settling time/ delay caused by the filtering will be considered within the DistributedClocks-Timestamp.*	
Resolution	24 bit (including sign)	
Connection technology	2/3/4/5/6-wire	
Sampling rate (per channel, simultaneous)	100 µs/10 ksps	
	Free down sampling by Firmware via decimation factor	
Oversampling	1…100 selectable	
Supported EtherCAT cycle time (depending on the operation mode)	DistributedClocks: min. 100 µs, max. 10 ms FrameTriggered/Synchron: min. 200 µs, max. 100 ms FreeRun: not yet supported	
Connection diagnosis	Wire break/short cut	
Internal analog ground AGND	Existing by external connection to -Uv	
Overvoltage protection of the inputs related to -Uv (internal ground)	Value to follow	
Internal power supply	Via EtherCAT P	
Nominal voltage U_S	24 V DC (-15 %/+20 %)	
Current consumption from U_S	typ. 150 mA	
Nominal voltage U_P	24 V DC (-15 %/+20 %)	
Current consumption from U_P	No (diagnosis LED only, sensors will be supplied by U_S)	
Thermal power dissipation	typ. 3 W	
Dielectric strength - destruction limit	max. permitted short-term/permanent voltage between the contact points ±I1, ±I2, +Uv and –Uv: non-supplied ±40 V, supplied ±36 V Note: -Uv corresponds to internal AGND	
Recommended operation voltage range to compliance with specification	Max. permitted voltage during specified normal operation between ±I1 And ±I2: typ. ±10 V against –Uv Note: -Uv corresponds to internal AGND	

Common data	EPP3504-0023	ERP3504-0022
Distributed Clocks	Yes, with oversampling n = 1…100, accuracy << 1 µs	
Special features	Extended range 107 %, freely configurable numeric filters, TrueRMS, integrator/differentiator, non-linear scaling, PeakHold	
Functional diagnosis [1]	Yes	
Electrical isolation channel/channel [2]	-	
Electrical isolation channel/EtherCAT [2]	Functional insulation, 707 V DC (type test)	

Common data	EPP3504-0023	ERP3504-0022
Electrical isolation channel/SGND [2)]	Functional insulation, 707 V DC (type test)	
Configuration	Via the EtherCAT Master, e.g. TwinCAT	
Note to cable length	Signal cable lengths to the sensor / encoder over 3 m must be shielded, the shield design must be in line with the state of the art and be effective. For larger cable lengths >30 m a suitable surge protection should be provided if appropriate interference could affect the signal cable.	

[1)] see chapters Self-test and self-test report and Connection test/switchable connection diagnosis

[2)] see notes to potential groups within chapter Power supply, potential groups

Basic mechanical properties	EPP3504-0023	ERP3504-0022
Connection type	Push-In spring-loaded terminal PC FMC 1.5 6-pin, RM 2.54 mm Pluggable, 25 mating cycles	M12, 8 pol.
Dimensions (W x H x D)	See chapter “Mounting and wiring”/ “Mounting”/ “Dimensions”	
Mounting	Directly on metallic, earthed structure screwed	
Note mounting	Plug partly not within scope of delivery, see chapter Notes on connection technology [▶ 216]	
Weight	approx. 250 g	
Permissible ambient temperature range during operation	-25…+60 °C	
Permissible ambient temperature range during storage	-25…+85 °C	

Environmental data	EPP3504-0023	ERP3504-0022
Permissible operating altitude range	0 up to 2000 m (derating at higher altitudes on request)	
Relative humidity	max. 95%, no condensation	
Protection class	IP 20 (plug connector)	IP67 (M12 connector)

Normative data	EPP3504-0023	ERP3504-0022
Vibration-/shock resistance	Conforms to EN 60068-2-6 / EN 60068-2-27	
EMC-resistance / emission	Conforms to EN 61000-6-2 / EN 61000-6-4	
Approvals/ markings *)	CE, UKCA, EAC	
EMC notes	In case of push-in connectors, ESD air discharges conforming to EN 61000-6-4 into the connectors or to the lines connected there can lead to measurement deviations up to ±FSV within the respective channel or to other channels by crosstalk. Peak voltages (surge) conforming to EN 61000-6-2 that are applied to the cable shield can lead to measurement deviations up to ±FSV.	

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

4.2.1 EPP3504-0023/ ERP3504-0022 overview measurement ranges

Measurement	Connection technology	FSV	Mode	Maximum value/ value range
Voltage	2 wire	±10 V	Extended	±10.737.. V
			Legacy	±10 V
		±80 mV	Extended	±85.9.. mV
			Legacy	±80 mV
PT1000	2/3/4 wire	2000 Ω	Legacy	266 °C
Potentiometer	3/5 wire	±1 V/V	Extended	±1 V/V
			Legacy	
Full bridge	4/6-wire	±32 mV/V	Extended	±34.359.. mV/V
			Legacy	±32 mV/V
		±8 mV/V	Extended	±8.5899.. mV/V
			Legacy	±8 mV/V
		±4 mV/V	Extended	±4.2949.. mV/V
			Legacy	±4 mV/V
		±2 mV/V	Extended	±2.1474.. mV/V
			Legacy	±2 mV/V
		±4 mV/V comp.	Extended	±4.2949.. mV/V
			Legacy	±4 mV/V
		±2 mV/V comp.	Extended	±2.1474.. mV/V
			Legacy	±2 mV/V
Half bridge	3/5-wire	±16 mV/V	Extended	±17.179.. mV/V
			Legacy	±16 mV/V
		±8 mV/V	Extended	±8.5899.. mV/V
			Legacy	±8 mV/V
		±4 mV/V	Extended	±4.2949.. mV/V
			Legacy	±4 mV/V
		±2 mV/V	Extended	±2.1474.. mV/V
			Legacy	±2 mV/V
		±4 mV/V comp.	Extended	±4.2949.. mV/V
			Legacy	±4 mV/V
		±2 mV/V comp.	Extended	±2.1474.. mV/V
			Legacy	±2 mV/V
Quarter bridge 120/350/1000 Ω	2/3 wire	±32 mV/V	Extended	±34.359.. mV/V
			Legacy	±32 mV/V
		±8 mV/V	Extended	±8.5899.. mV/V
			Legacy	±8 mV/V
		±4 mV/V	Extended	±4.2949.. mV/V
			Legacy	±4 mV/V
		±2 mV/V	Extended	±2.1474.. mV/V
			Legacy	±2 mV/V

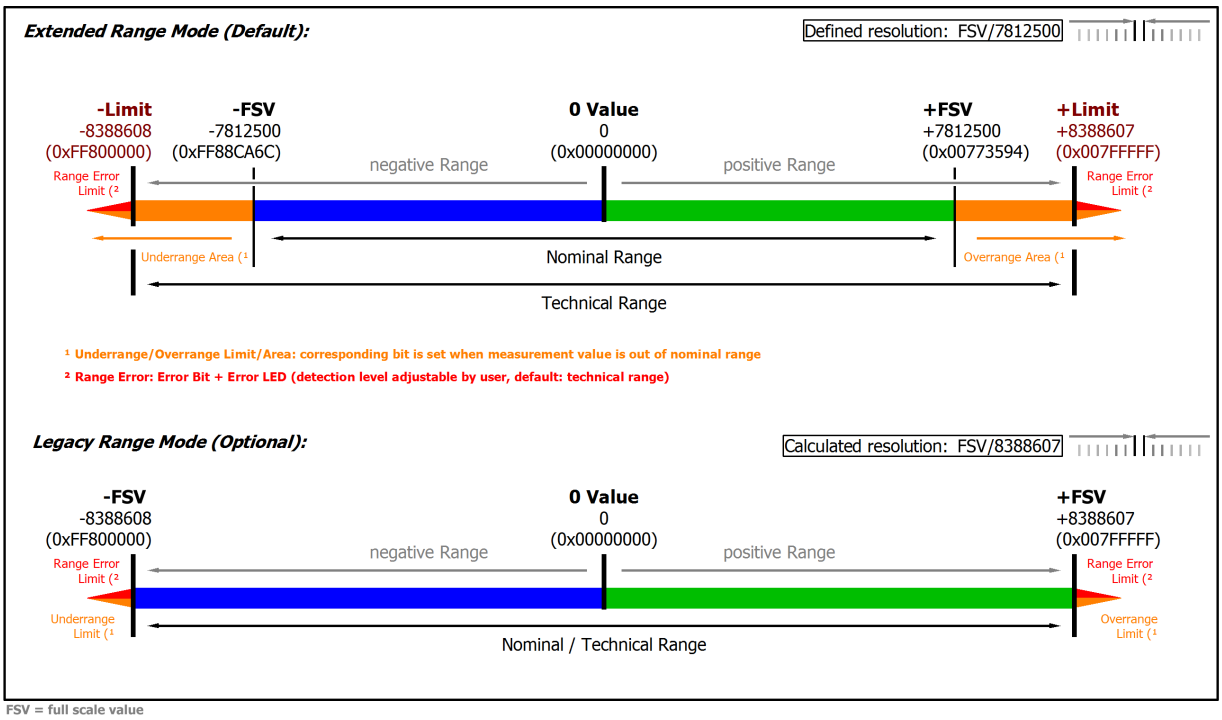


Fig. 5: Overview measurement ranges, Bipolar

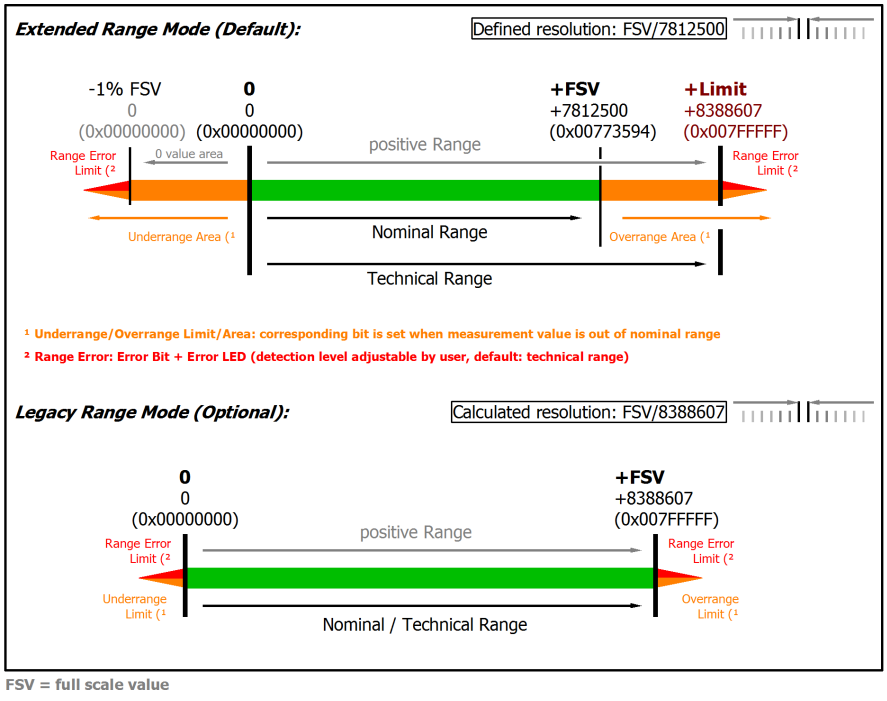


Fig. 6: Overview measurement ranges, Unipolar

Note: In Extended Range Mode the Underrange/Overrange display in the PDO status has the character of an information/warning when the nominal measuring range is exceeded, i.e. no *Error* is displayed in the PDO status and LED. If the technical measuring range is also exceeded, *Error* = *TRUE* is also displayed. The detection limit for Underrange/Overrange *Error* can be set in the CoE.

In Legacy Range mode, an Underrange/Overrange event also leads to an *Error* in the PDO status.

4.2.2 Measurement ±10 V

Measurement mode		±10 V	
Measuring range, nominal		-10…+10 V	
Measuring range, end value (FSV)		10 V	
Measuring range, technically usable		-10.737…+10.737 V	
PDO resolution (including sign)		24 bit	16 bit [2]
PDO LSB (Extended Range)		1.28 µV	327.68 µV
PDO LSB (Legacy Range)		1.192.. µV	305.18.. µV
Basic accuracy: Measuring deviation at 23°C, with averaging, typ. [6]		< ±0.015 %, < ±150 ppm_{FSV} < ±1.5 mV	
Extended basic accuracy: Measuring deviation at 0…55°C, with averaging, typ. [6]		< ±0.023 %, < ±230 ppm_{FSV} < ±2.3 mV	
Offset/Zero point deviation (at 23°C)	E_{Offset}	< 30 ppm_{FSV}	
Gain/scale/amplification deviation (at 23°C)	E_{Gain}	< 140 ppm	
Non-linearity over the whole measuring range	E_{Lin}	< 40 ppm_{FSV}	
Repeatability, over 24 h, with averaging	E_{Rep}	< 20 ppm_{FSV}	
Temperature coefficient, typ.	Tc_{Gain}	< 5 ppm/K	
	Tc_{Offset}	< 2 ppm_{FSV}/K < 20 µV/K	
Input impedance ±Input 1 (internal resistance)		Differential typ. approx. 4.12 MΩ || 11 nF CommonMode typ. approx. 40 nF against SGND	

Preliminary data:

Measurement mode		±10 V		
Largest short-term deviation during a specified electrical interference test		*±0.06* % = *600* ppm_{FSV} typ.		
Common-mode rejection ratio (without filter)		DC: >*115* dB typ.	50 Hz: >*105* dB typ.	1 kHz: >*80* dB typ.
Common-mode rejection ratio (with 50 Hz FIR filter) [5]		DC: >*115* dB typ.	50 Hz: >*115* dB typ.	1 kHz: >*115* dB typ.
Noise (without filtering)	$E_{Noise, PtP}$	< *60.0* ppm_{FSV}	< *469* digits	< *0.6* mV
	$E_{Noise, RMS}$	< *10.0* ppm_{FSV}	< *78* digits	< *100.0* µV
	Max. SNR	> *100* dB		
	Noisedensity@1kHz	< *1.41* $\frac{\mu V/V}{\sqrt{Hz}}$		
Noise (with 50 Hz FIR filter) [5]	$E_{Noise, PtP}$	< *9.0* ppm_{FSV}	< *70* digits	< *90.0* µV
	$E_{Noise, RMS}$	< *1.5* ppm_{FSV}	< *12* digits	< *15.0* µV
	Max. SNR	> *116.5* dB		

[2]) The analog measurement is always done with 24 bits, in 16-bit mode the eight least significant bits are cut off. For further information see chapter "Commissioning"/ "Process data overview" [▶ 63]

[5]) the 50/60 Hz filter setting must be selected via the settings in the CoE depending on the local mains frequency (50 or 60 Hz); the respective nominal mains frequency then applies as the measuring point for common mode rejection or crosstalk.

[6]) Calculated value according to the equation for estimating usability at the given ambient temperature point or estimating usability over the specified ambient temperature range in operation ($T_{ambient}$) as stated in chapter "General information on measuring accuracy/measurement uncertainty" [▶ 57]. In real use, for example at a relatively constant ambient temperature $T_{ambient}$, a lower (better) achievable uncertainty is attained. A specific calculation according to chapter "General information on measuring accuracy/ measurement uncertainty" is recommended, especially if the instrument allows a wider ambient temperature range in operation than 0...55 °C.

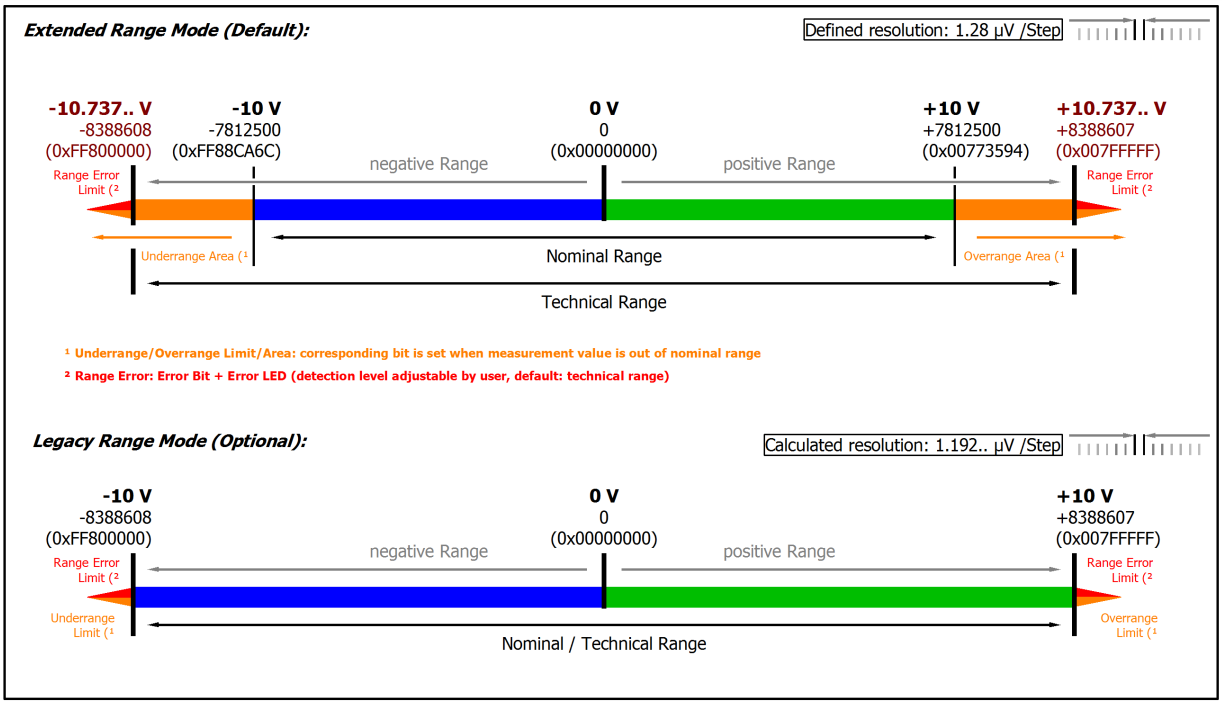


Fig. 7: Representation ±10 V measurement range

Note: In Extended Range Mode the Underrange/Overrange display in the PDO status has the character of an information/warning when the nominal measuring range is exceeded, i.e. no *Error* is displayed in the PDO status and LED. If the technical measuring range is also exceeded, *Error* = *TRUE* is also displayed. The detection limit for Underrange/Overrange *Error* can be set in the CoE.

In Legacy Range mode, an Underrange/Overrange event also leads to an *Error* in the PDO status.

4.2.3 Measurement ±80 mV

Measurement mode		±80 mV	
Measuring range, nominal		-80…+80 mV	
Measuring range, end value (FSV)		80 mV	
Measuring range, technically usable		-85.9…+85.9 mV	
PDO resolution (including sign)		24 bit	16 bit [2]
PDO LSB (Extended Range)		10.24 nV	2.621.. µV
PDO LSB (Legacy Range)		9.536.. nV	2.441.. µV
Basic accuracy: Measuring deviation at 23°C, with averaging, typ. [6]		< ±0.02 %, < ±200 ppm_{FSV} < ±16.0 µV	
Extended basic accuracy: Measuring deviation at 0…55°C, with averaging, typ. [6]		< ±0.0305 %, < ±305 ppm_{FSV} < ±24.4 µV	
Offset/Zero point deviation (at 23°C)	E_{Offset}	< 95 ppm_{FSV}	
Gain/scale/amplification deviation (at 23°C)	E_{Gain}	< 165 ppm	
Non-linearity over the whole measuring range	E_{Lin}	< 60 ppm_{FSV}	
Repeatability, over 24 h, with averaging	E_{Rep}	< 20 ppm_{FSV}	
Temperature coefficient, typ.	Tc_{Gain}	< 5 ppm/K	
	Tc_{Offset}	< 5 ppm_{FSV}/K < 0.40 µV/K	
Input impedance ±Input 1 (internal resistance)		Differential typ. approx. 4.12 MΩ || 11 nF CommonMode typ. approx. 40 nF against SGND	

Preliminary data:

Measurement mode		±80 mV		
Largest short-term deviation during a specified electrical interference test		*±0.08* % = *800* ppm_{FSV} typ.		
Common-mode rejection ratio (without filter)		DC: >*115* dB typ.	50 Hz: >*105* dB typ.	1 kHz: >*80* dB typ.
Common-mode rejection ratio (with 50 Hz FIR filter) [5]		DC: >*115* dB typ.	50 Hz: >*115* dB typ.	1 kHz: >*115* dB typ.
Noise (without filtering)	$E_{Noise, PtP}$	< *150.0* ppm_{FSV}	< *1172* digits	< *0.01* mV
	$E_{Noise, RMS}$	< *25.0* ppm_{FSV}	< *195* digits	< *2.0* µV
	Max. SNR	> *92* dB		
	Noisedensity@1kHz	< *0.03* $\frac{\mu V/V}{\sqrt{Hz}}$		
Noise (with 50 Hz FIR filter) [5]	$E_{Noise, PtP}$	< *18.0* ppm_{FSV}	< *141* digits	< *1.44* µV
	$E_{Noise, RMS}$	< *3.0* ppm_{FSV}	< *23* digits	< *0.24* µV
	Max. SNR	> *110.5* dB		

[2]) The analog measurement is always done with 24 bits, in 16-bit mode the eight least significant bits are cut off. For further information see chapter "Commissioning"/ "Process data overview" [▶ 63]

[5]) the 50/60 Hz filter setting must be selected via the settings in the CoE depending on the local mains frequency (50 or 60 Hz); the respective nominal mains frequency then applies as the measuring point for common mode rejection or crosstalk.

[6]) Calculated value according to the equation for estimating usability at the given ambient temperature point or estimating usability over the specified ambient temperature range in operation ($T_{ambient}$) as stated in chapter "General information on measuring accuracy/measurement uncertainty" [▶ 57]. In real use, for example at a relatively constant ambient temperature $T_{ambient}$, a lower (better) achievable uncertainty is attained. A specific calculation according to chapter "General information on measuring accuracy/ measurement uncertainty" is recommended, especially if the instrument allows a wider ambient temperature range in operation than 0...55 °C.

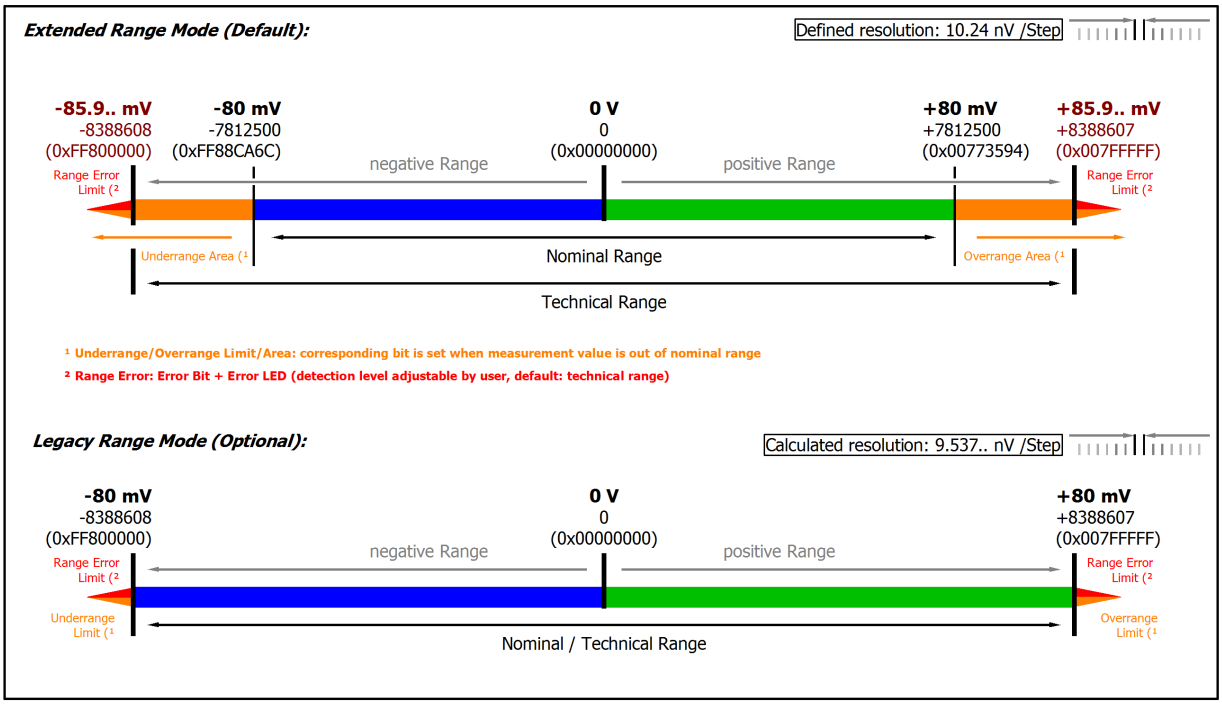


Fig. 8: Representation ±80 mV measurement range

Note: In Extended Range Mode the Underrange/Overrange display in the PDO status has the character of an information/warning when the nominal measuring range is exceeded, i.e. no *Error* is displayed in the PDO status and LED. If the technical measuring range is also exceeded, *Error* = *TRUE* is also displayed. The detection limit for Underrange/Overrange *Error* can be set in the CoE.

In Legacy Range mode, an Underrange/Overrange event also leads to an *Error* in the PDO status.

4.2.4 RTD/Pt1000 measurement

RTD specification and conversion

Temperature measurement with a resistance-dependent RTD sensor generally consists of two steps:

- Electrical measurement of the resistance, if necessary in several ohmic measuring ranges
- Conversion (transformation) of the resistance into a temperature value by software means according to the set RTD type (Pt100, Pt1000…).

Both steps can take place locally in the Beckhoff measurement device. The transformation in the device can also be deactivated if it is to be calculated on a higher level in the control. Depending on the device type, several RTD conversions can be implemented which only differs in software. This means for Beckhoff RTD measurement devices that

- a specification table of the electrical resistance measurement is given
- and based on this, the effect for the temperature measurement is given below depending on the supported RTD type. Note that RTD characteristic curves are always realized as higher-order equations or by a sampling points table in the software, therefore a linear R→T transfer only makes sense in a narrow range.

Application on the EPP3504/ ERP3504

The EPP3504/ ERP3504 supports the measurement of resistances up to 2 kΩ in 2/3/4-wire measurement and the conversion of Pt1000 RTD sensors up to 2000 Ω / 266 °C.

Although the EPP3504/ ERP3504 does not support a sole resistance measurement (without conversion to temperature), a resistance specification is given here because the temperature measurement is based on it.

Note on 2-/3-/4-wire connection in R/RTD mode

With **2-wire measurement**, the line resistance of the sensor supply lines influences the measured value. If a reduction of this systematic error component is desirable for 2-wire measurements, the resistance of the supply line to the measuring resistance should be taken into account, in which case the resistance of the supply line has to be determined first.

Taking into account the uncertainty associated with this supply line resistance, it can then be included statically in the calculation, in the EPP3504-0023 via CoE object 0x80n0:13 [▶ 71].

Any change in resistance of the supply line due to ageing, for example, is not taken into account automatically. Just the temperature dependency of copper lines with approx. 4000 ppm/K (corresponds to 0.4%/K!) is not insignificant during 24/7 operation.

A **3-wire measurement** enables the systematic component to be eliminated, assuming that the two supply lines are identical. With this type of measurement, the lead resistance of a supply line is measured continuously. The value determined in this way is then deducted twice from the measurement result, thereby eliminating the line resistance. Technically, this leads to a significantly more reliable measurement. However, taking into account the measurement uncertainty, the gain from the 3-wire connection is less significant, since this assumption is subject to high uncertainty, in view of the fact that the individual line that was not measured may be damaged, or a varying resistance may have gone unnoticed.

Therefore, although technically the 3-wire connection is a tried and tested approach, for measurements that are methodological assessed based on measurement uncertainty, we strongly recommend fully-compensated **4-wire connection**.

With both 2-wire and 3-wire connection, the contact resistances of the terminal contacts influence the measuring process. The measuring accuracy can be further increased by a user-side adjustment with the signal connection plugged in.

NOTICE

Measurement of small resistances

Especially for measurements in the range < 10 Ω, the 4-wire connection is absolutely necessary due to the relatively high supply and contact resistances. It should also be considered that with such low resistances the relative measurement error in relation to the full scale value (FSV) can become high - for such measurements resistance measurement terminals with small measuring ranges such as EL3692 in 4-wire measurement should be used if necessary.

Corresponding considerations also lead to the common connection methods in bridge operation:

- Full bridge: 4-wire connection without line compensation, 6-wire connection with full line compensation
- Half bridge: 3-wire connection without line compensation, 5-wire connection with full line compensation
- Quarter bridge: 2-wire connection without line compensation, 3-wire connection with theoretical line compensation and 4-wire connection with full line compensation

Resistance measurement 2 kΩ		2/3-wire [1]	4-wire
Operation mode		3 V feed voltage fixed setting on +Uv Internal 1 kΩ reference resistance at –I2 Supply current is given by: 3 V / (1 kΩ + $R_{measurement}$) → max. 3 mA	
Measuring range, nominal		2 kΩ (corresponds to PT1000 +266°C)	
Measuring range, end value (FSV)		2 kΩ	
Measuring range, technically usable		0 … 2 kΩ	
PDO LSB (Extended Range)		Extended range is not supported for resistance measurement	
PDO LSB (Legacy Range only)		Resistance measurement not available as separate measuring range on EPP3504-0023.	
Basic accuracy: Measuring deviation at 23°C, with averaging, incl. Offset, typ. [6]		< ±0.012 $\%_{FSV}$ < ±120 ppm_{FSV} < ±240 mΩ	< ±0.011 $\%_{FSV}$ < ±110 ppm_{FSV} < ±220 mΩ
Extended basic accuracy: Measuring deviation at 0…55°C, with averaging, incl. Offset, typ. [6]		< ±0.0365 $\%_{FSV}$ < ±365 ppm_{FSV} < ±0.73 Ω	< ±0.0345 $\%_{FSV}$ < ±345 ppm_{FSV} < ±0.69 Ω
Offset/Zero point deviation (at 23°C)	E_{Offset}	< 40 ppm_{FSV}	< 30 ppm_{FSV}
Gain/scale/amplification deviation (at 23°C)	E_{Gain}	< 90 ppm	< 80 ppm
Non-linearity over the whole measuring range	E_{Lin}	< 65 ppm_{FSV}	< 65 ppm_{FSV}
Repeatability (at 23°C)	E_{Rep}	< 10 ppm_{FSV}	< 10 ppm_{FSV}
Temperature coefficient, typ.	Tc_{Gain}	< 10 ppm/K	< 10 ppm/K
	Tc_{Offset}	< 4 ppm_{FSV}/K < 8 mΩ/K	< 1.5 ppm_{FSV}/K < 3 mΩ/K
Common-mode rejection ratio (without filter) [3]		tbd.	tbd.
Common-mode rejection ratio (with 50/60 Hz FIR filter) [3] [5]		tbd.	tbd.
Largest short-term deviation during a specified electrical interference test, typ.		tbd. $\%_{FSV}$	tbd. $\%_{FSV}$
Input impedance (internal resistance)		tbd.	tbd.

Preliminary data:

Resistance measurement 2kΩ	2/3-wire [1]	4-wire
Largest short-term deviation during a specified electrical interference test, typ.	*0.048* $\%_{FSV}$ = *480* ppm_{FSV} typ.	*0.044* $\%_{FSV}$ = *440* ppm_{FSV} typ.

[1]) The offset specification does not apply to 2-wire operation, since the offset is increased on the device side. Therefore, a system-side offset adjustment is recommended, see "Note on 2-/3-/4-wire connection in R/RTD mode". The final targeting basic acuuracy within the 2-wire operation is mainingly dependent by the quality of this system-side offset adjustment.

[3]) Values related to a common mode interference between SGND and internal ground.

RTD measuring range

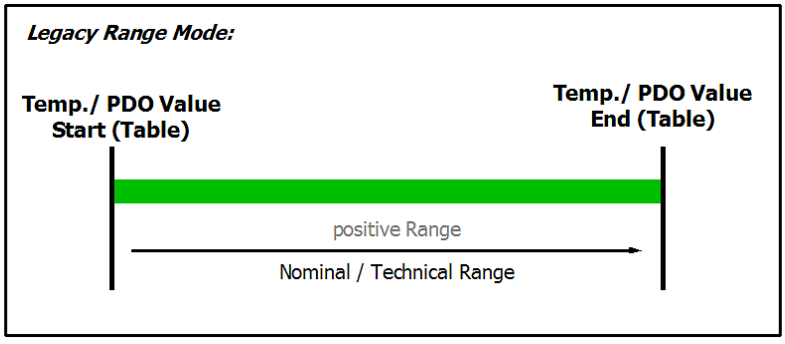


Fig. 9: Chart: RTD measuring range

In temperature mode, only the legacy range is available, the extended range is not available.

The temperature display in [°C/digit] (e.g. 0.1°/digit or 0.01°/digit) is independent from the electrical measurement. It is “just” a display setting and results from the PDO setting, see chapter "Comissioning".

Data for the sensor types in the following table

The values for the sensor types listed in the following table are shown here merely for informative purposes as an orientation aid. All data are given without guarantee and must be cross-checked against the data sheet for the respective sensor employed.

The RTD measurement consists of a chain of measuring and computing elements that affect the attainable measurement deviation:

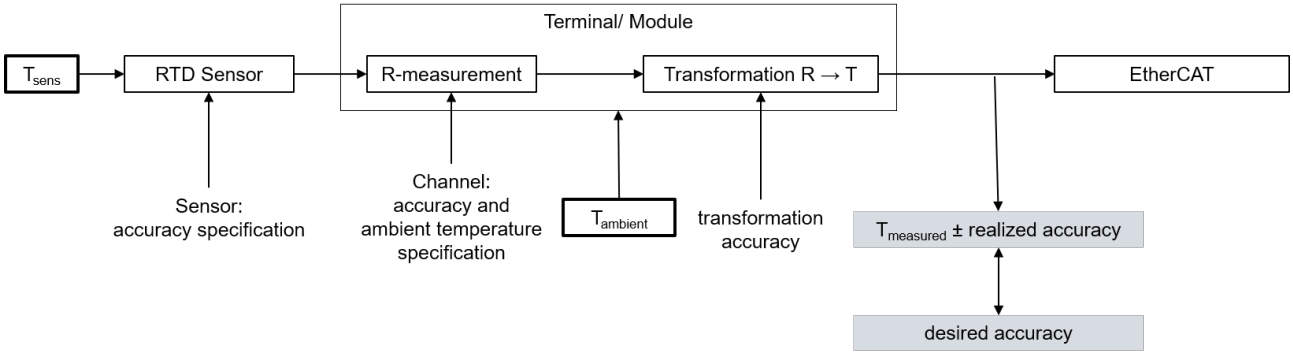


Fig. 10: Concatenation of the uncertainties in RTD measurement

The given resistance specification is decisive for the attainable temperature measurement accuracy. It is applied to the possible RTD types in the following.

On account of

- the non-linearity existing in the RTD and thus the high dependency of the specification data on the sensor temperature T_{sens} and
- the influence of the ambient temperature on the analog input device employed (leads to a change in $T_{measured}$ on account of $\Delta T_{ambient}$ although T_{sens} = constant)

no detailed temperature specification table is given in the following, but

- a short table specifying the electrical measuring range and orientation value for the basic accuracy
- a graph of the basic accuracy over T_{sens} (this at two example ambient temperatures so that the attainable basic accuracy is implied on account of the actual existing ambient temperature)
- equations for calculating further parameters (offset/gain/non-linearity/repeatability/noise) if necessary from the resistance specification at the desired operating point

RTD types supported by the EPP3504-0023:

- Pt1000 according to DIN EN 60751/IEC751 with α= 0.0039083 [1/C°]

RTD temperature measurement	Pt1000 2-wire	Pt1000 3-wire	Pt1000 4-wire
Electrical measuring range used	2 kΩ		
Starting value	-200°C ≈ 185.2 Ω		
End value	266°C ≈ 2000 Ω		
PDO LSB (legacy range only)	0.1/0.01/0.001°C/digit, depending on PDO setting		
Basic accuracy: measurement deviation at 23 °C ambient temperature, with averaging, typ.	The achievable measurement uncertainty, as a system side offset, is essentially dependent on the line resistances and can at best reach the value of the 3-wire measurement.	< ±*132* mK (*preliminary value*)	< ±*120* mK (*preliminary value*)
Temperature coefficient [2], typ.		< *6.6* mK/K (*preliminary value*)	< *5.4* mK/K (*preliminary value*)

[2]) The temperature coefficient, i.e. the change in the measured temperature value in relation to the change in the ambient temperature of the box module, is not constant, as can be seen in the following plot. The value at a sensor temperature of 0 °C is given here as an orientation value. Further values can be taken from the plot.

Basic accuracy for Pt1000, 3-wire connection:

- in preparation -

Fig. 11: Diagram for basic accuracy for Pt1000, 3-wire connection

Basic accuracy for Pt1000, 4-wire connection:

- in preparation -

Fig. 12: Diagram for basic accuracy for Pt1000, 4-wire connection

If further specification data are of interest, they can or must be calculated from the values given in the resistance specification.

The sequence:

- General: The conversion is explained here only for one measuring point (a certain input signal); the steps simply must be repeated in case of several measuring points (up to the entire measuring range).
- If the measured resistance at the measured temperature measuring point is unknown, the measured value (MW) in [Ω] must be determined:
  MW = $R_{Measuring\ point}$ ($T_{Measuring\ point}$) with the help of an R→T table
- The deviation at this resistance value is calculated
  - Via the total equation

$$E_{Total} = \sqrt{(E_{Gain} \cdot \frac{MV}{FSV})^2 + (TC_{Gain} \cdot \Delta T \cdot \frac{MV}{FSV})^2 + E_{Offset}^2 + E_{Lin}^2 + E_{Rep}^2 + (\frac{1}{2} \cdot E_{Noise,PtP})^2 + (TC_{Offset} \cdot \Delta T)^2 + (E_{Age} \cdot N_{Years})^2}$$

  - or a single value, e.g. E_{Single} = 15 ppm_{FSV}
  - the measurement uncertainty in [Ω] must be calculated:
    $E_{Resistance}(R_{Measuring\ point}) = E_{Total}(R_{Measuring\ point}) \cdot FSV$
    or: $E_{Resistance}(R_{Measuring\ point}) = E_{Single}(R_{Measuring\ point}) \cdot FSV$
    or (if already known) e.g.: $E_{Resistance}(R_{Measuring\ point})$ = 0.03 Ω
- The slope at the point used must then be determined:
  $\Delta R_{proK}(T_{Measuring\ point}) = [\ R(T_{Measuring\ point} + 1\ °C) - R(T_{Measuring\ point})]\ /\ 1\ °C$
  with the help of an R→T table
- The temperature measurement uncertainty can be calculated from the resistance measurement uncertainty and the slope
  $E_{Temp}(R_{Measuring\ point}) = (E_{Resistance}(T_{Measuring\ point})) / (\Delta R_{proK}(T_{Measuring\ point}))$
- To determine the error of the entire system consisting of RTD and the measuring device in [°C], the two errors must be added together quadratically:

$$E_{System} = \sqrt{(E_{Temp})^2 + (E_{RTD})^2}$$

The numerical values used in the following three examples are for illustration purposes. The specification values given in the technical data remain authoritative.

Example 1:

Basic accuracy of an ELM3504 at 35 °C ambient temperature, measurement of -100 °C in the PT1000 interface (4-wire), without the influence of noise and aging:

$T_{Measuring\ point}$ = -100 °C

MW = $R_{PT1000,\ -100\ °C}$ = 602.56 Ω

$$E_{Total} = \sqrt{((80\ ppm \cdot (602.56\ \Omega)/(2000\ \Omega))^2 + (10\ ppm/K \cdot 12\ K \cdot (602.56\ \Omega)/(2000\ \Omega))^2 + (30\ ppm_{FSV})^2 \ldots}$$
$$\ldots\overline{\ldots + (65\ ppm_{FSV})^2 + (10\ ppm_{FSV})^2 + (1.5\ (ppm_{FSV})/K \cdot 12\ K)^2)}$$

= 86.238 ppm_{FSV}

$E_{Resistance}(R_{Measuring\ point}) = 86.238\ ppm_{FSV} \cdot 2000\ \Omega = 0.1725\ \Omega$

$\Delta R_{proK}(T_{Measuring\ point}) = (R(-99\ °C) - R(-100\ °C)) / (1\ °C) = 4.05\ \Omega/°C$

$E_{ELM3504@35°C,\ PT1000,\ -100\ °C} = (0.1725\ \Omega)/(4.05\ \Omega/°C) \approx 0.043\ °C$ (means ±0.043 °C)

Example 2:

Consideration of the repeatability alone under the above conditions:

$T_{Measuring\ point} = -100\ °C$

$MW = R_{Measuring\ point}\ (-100\ °C) = 602.56\ \Omega$

$E_{Single} = 10\ ppm_{FSV}$

$E_{Resistance} = 10\ ppm_{FSV} \cdot 2000\ \Omega = 0.02\ \Omega$

$\Delta R_{proK}(T_{Measuring\ point}) = (R_{-99\ °C} - R_{-100\ °C}) / 1\ °C = 4.05\ \Omega/°C$

$E_{Temp}(R_{Measuring\ point}) = 0.02\ \Omega / 4.05\ \Omega/°C \approx 0.005\ °C$ (means ±0.005 °C)

Example 3:

Consideration of the RMS noise alone without filter under the above conditions:

$T_{Measuring\ point} = -100\ °C$

$MW = R_{Measuring\ point}\ (-100\ °C) = 602.56\ \Omega$

$E_{Single} = 37\ ppm_{FSV}$

$E_{Resistance} = 37\ ppm_{FSV} \cdot 2000\ \Omega = 0.074\ \Omega$

$\Delta R_{proK}(T_{Measuring\ point}) = (R_{-99\ °C} - R_{-100\ °C}) / 1\ °C = 4.05\ \Omega/°C$

$E_{Temp}(R_{Measuring\ point}) = 0.074\ \Omega / 4.05\ \Omega/°C \approx 0.018\ °C$ (means ± 0.018 °C)

Example 4:

If the noise $E_{Noise,\ PtP}$ of the above example terminal is considered not for one sensor point -100 °C but in general, the following plot results:

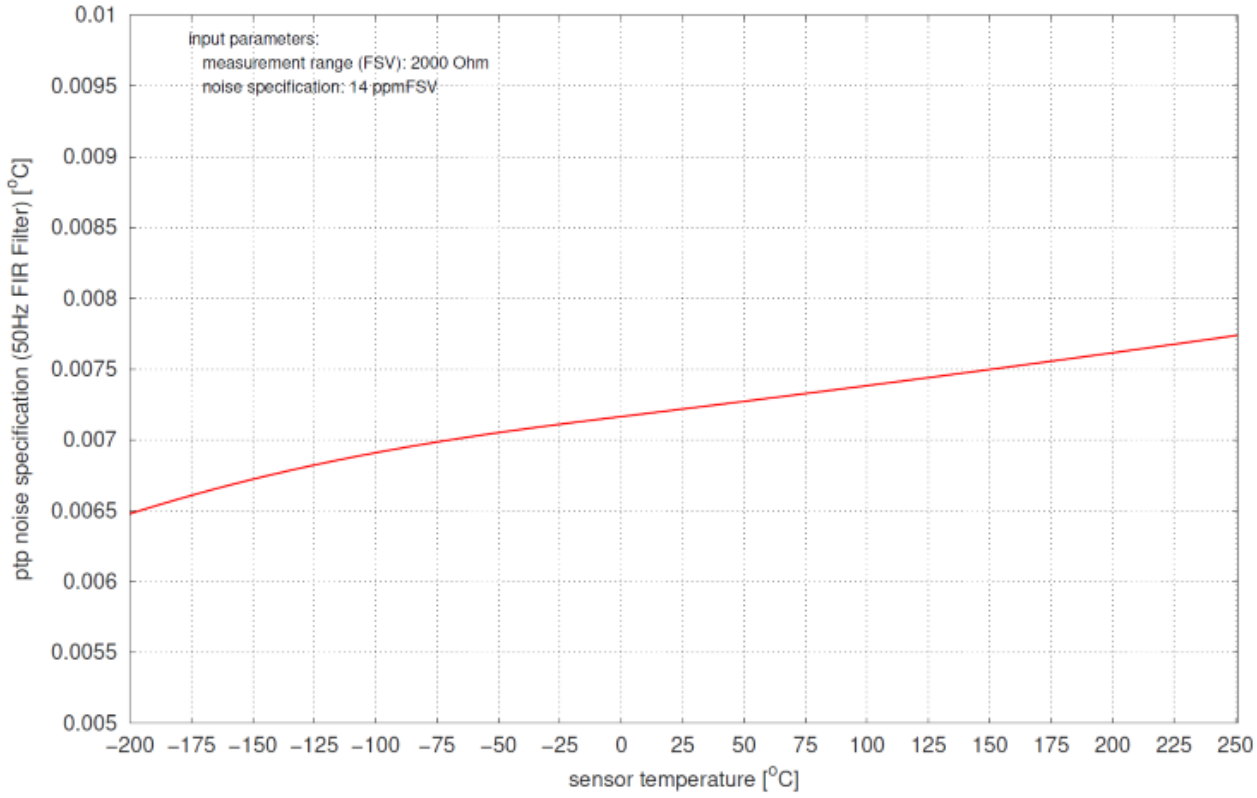


Fig. 13: Diagram noise $E_{Noise, PtP}$ in dependence on sensor temperature

4.2.5 Potentiometer measurement

The potentiometer should be supplied via the integrated power supply unit (max. 5 V, configurable). The slider voltage is then measured in relation to the supply voltage and given as a percentage. Technically, the measurement process is similar to a half-bridge strain gauge.

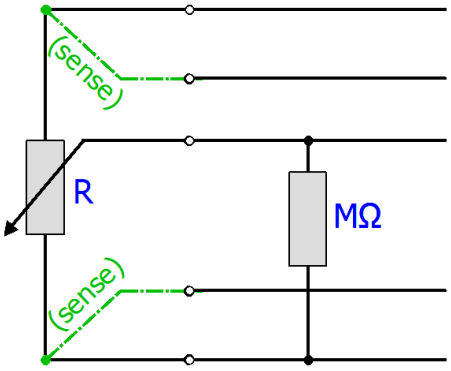
In the following, the specification is given for a 5-wire configuration; external line resistances are compensated for by the 5-wire configuration and the potentiometer is recorded directly in the measuring channel. In the 3-wire configuration, the measuring channel essentially meets the same specification, as it continues to measure internally in 5-wire configuration. However, the lead and contact resistance levels in cables and plugs, which are usually unclear and temperature-dependent, must be taken into account by the connected potentiometer, as they affect measurement. The entire "potentiometer + supply lines + measuring channel" system in the 3-wire configuration will therefore not be able to achieve the specification values listed below.

Potential diagnoses at full-scale deflection or 0 display:

- slider break
- interruption in supply

Potentiometers from 500 Ω can be used.

The specification and the manufacturer's calibration were determined with a 5 kΩ potentiometer. If a different potentiometer resistance is used, the measurements will be incorrect due to the changed parallel circuit ratios, see "Supply, potential groups" chapter.



The distortion can also be recorded mathematically:

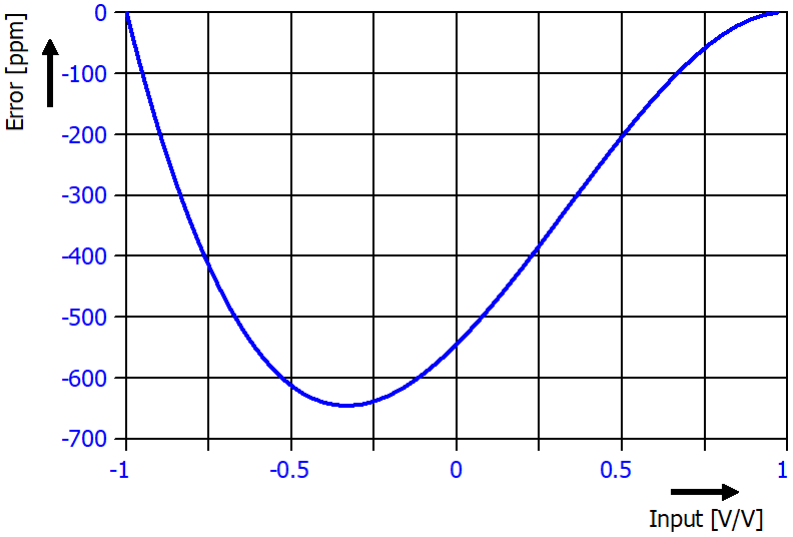


Fig. 14: difference between 500 Ω and 5 kΩ potentiometer

Using the sample programs

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

The following example code can be used to theoretically correct a different potentiometer:

```
VAR
    R_Poti_calibration : LREAL := 5000.0; // Ohm (vendor calibration case)
    R_Poti_actual : LREAL := 500.0; // Ohm (user application case)
    Rin : LREAL := 2.06E6; // Ohm
    PDO : LREAL; // RAW Value
    PDO_Ratio : LREAL;
    R2_act : LREAL;
    R1_act : LREAL;
    R2_act_Rin_parallel : LREAL;
    R2_calibration : LREAL;
    R1_calibration : LREAL;
    R2_calibration_Rin_parallel : LREAL;
    PDO_corrected : LREAL;
END_VAR
```

```
PDO_Ratio := PDO / 7812500.0;
R2_act := R_Poti_actual / 2.0 - PDO_Ratio * R_Poti_actual / 2.0;
R1_act := R_Poti_actual / 2.0 + PDO_Ratio * R_Poti_actual / 2.0;
IF (R2_act + Rin) <> 0 THEN
            R2_act_Rin_parallel := (R2_act * Rin) / (R2_act + Rin);
ELSE
            R2_act_Rin_parallel := 0.0;
END_IF
R2_calibration := R_Poti_calibration / 2.0 - PDO_Ratio * R_Poti_calibration / 2.0;
R1_calibration := R_Poti_calibration / 2.0 + PDO_Ratio * R_Poti_calibration / 2.0;
IF (R2_calibration + Rin) <> 0 THEN
            R2_calibration_Rin_parallel := (R2_calibration * Rin) / (R2_calibration + Rin);
ELSE
            R2_calibration_Rin_parallel := 0.0;
END_IF
IF (R1_act + R2_act_Rin_parallel) <> 0 AND R_Poti_actual <> 0 AND (R1_calibration +
R2_calibration_Rin_parallel) <> 0 AND R_Poti_calibration <> 0 THEN
            PDO_corrected := (PDO_Ratio-
                        (((2.0*(0.5-R2_act_Rin_parallel/(R1_act+R2_act_Rin_parallel)))-
                        (2.0*(0.5-R2_act/R_Poti_actual)))-
                        ((2.0*(0.5-R2_calibration_Rin_parallel/
(R1_calibration+R2_calibration_Rin_parallel)))-
                        (2.0*(0.5-R2_calibration/R_Poti_calibration)))))*7812500.0;
ELSE
            PDO_corrected := 0.0;
END_IF
```

Measurement mode		Potentiometer (3/5-wire)
Operation mode		The supply voltage is configurable via CoE, 1...5 V
Measuring range, nominal		-1 … 1 V/V
Measuring range, end value (FSV)		1 V/V
Measuring range, technically usable		-1 …1 V/V
PDO resolution		24 bit (including sign)
PDO LSB (Extended Range)		0.128 ppm
PDO LSB (Legacy Range)		0.119… ppm
Basic accuracy: Measuring deviation at 23°C, with averaging, typ. [2) 6)]	without Offset	< ±0.0025 $\%_{FSV}$ < ±25 ppm_{FSV} < ±25 µV/V
	incl. Offset	< ±0.0075 $\%_{FSV}$ < ±75 ppm_{FSV} < ±75 µV/V
Extended basic accuracy: Measuring deviation at 0…55°C, with averaging, typ. [2) 6)]	without Offset	< ±0.0075 $\%_{FSV}$ < ±75 ppm_{FSV} < ±75 µV/V
	incl. Offset	< ±0.0105 $\%_{FSV}$ < ±105 ppm_{FSV} < ±105 µV/V
Offset/Zero point deviation (at 23°C)	E_{Offset}	< 70 ppm_{FSV}
Gain/scale/amplification deviation (at 23°C)	E_{Gain}	< 20 ppm
Non-linearity over the whole measuring range	E_{Lin}	< 15 ppm_{FSV}
Repeatability, over 24 h, with averaging	E_{Rep}	< 1 ppm_{FSV}
Temperature coefficient, typ.	Tc_{Gain}	< 2 ppm/K
	Tc_{Offset}	< 1 ppm_{FSV}/K < 1 µV/V/K
Input impedance (internal resistance)		typ. approx. 2.06 MΩ || 100 pF against AGND typ. approx. 20 nF against SGND

Preliminary data:

Measurement mode		Potentiometer (3/5-wire)		
Noise (without filtering, at 23°C)	$E_{Noise, PtP}$	< *105.0* ppm_{FSV} < *820* digits		
	$E_{Noise, RMS}$	< *18.0* ppm_{FSV} < *137* digits		
	Max. SNR	> *95.1* dB		
	Noisedensity@1kHz	< *0.18* $\frac{ppm}{\sqrt{Hz}}$		
Noise (with 50 Hz FIR filter, at 23°C) [5)]	$E_{Noise, PtP}$	< *9.0* ppm_{FSV} < *70* digits		
	$E_{Noise, RMS}$	< *1.5* ppm_{FSV} < *12* digits		
	Max. SNR	> *116.5* dB		
Common-mode rejection ratio (without filter) [3)]		DC: tbd. $\frac{mV/V}{V}$ typ.	50 Hz: tbd. $\frac{mV/V}{V}$ typ.	1 kHz: tbd. $\frac{mV/V}{V}$ typ.
Common-mode rejection ratio (with 50 Hz FIR filter) [3) 5)]		DC: tbd. $\frac{mV/V}{V}$ typ.	50 Hz: tbd. $\frac{mV/V}{V}$ typ.	1 kHz: tbd $\frac{µV/V}{V}$ typ.
Largest short-term deviation during a specified electrical interference test		*0.03* $\%_{FSV}$ = *300* ppm_{FSV} typ.		

[2]) A regular offset adjustment with connected potentiometer is recommended. The given offset specification of the box module is therefore practically irrelevant. Therefore, specification values with and without offset are given here. In practice, the offset component can be eliminated by the functions Tare [▶ 000] and also

ZeroOffset [▶ 000] of the box module or in the controller by a higher-level tare function. The offset deviation over time can change, therefore Beckhoff recommends a regular offset adjustment or careful observation of the change.

[3]) Values related to a common mode interference between SGND and internal ground.

[5]) the 50/60 Hz filter setting must be selected via the settings in the CoE depending on the local mains frequency (50 or 60 Hz); the respective nominal mains frequency then applies as the measuring point for common mode rejection or crosstalk.

[6]) Calculated value according to the equation for estimating usability at the given ambient temperature point or estimating usability over the specified ambient temperature range in operation ($T_{ambient}$) as stated in chapter "General information on measuring accuracy/measurement uncertainty" [▶ 57]. In real use, for example at a relatively constant ambient temperature $T_{ambient}$, a lower (better) achievable uncertainty is attained. A specific calculation according to chapter "General information on measuring accuracy/ measurement uncertainty" is recommended, especially if the instrument allows a wider ambient temperature range in operation than 0...55 °C.

Potentiometer measurement range

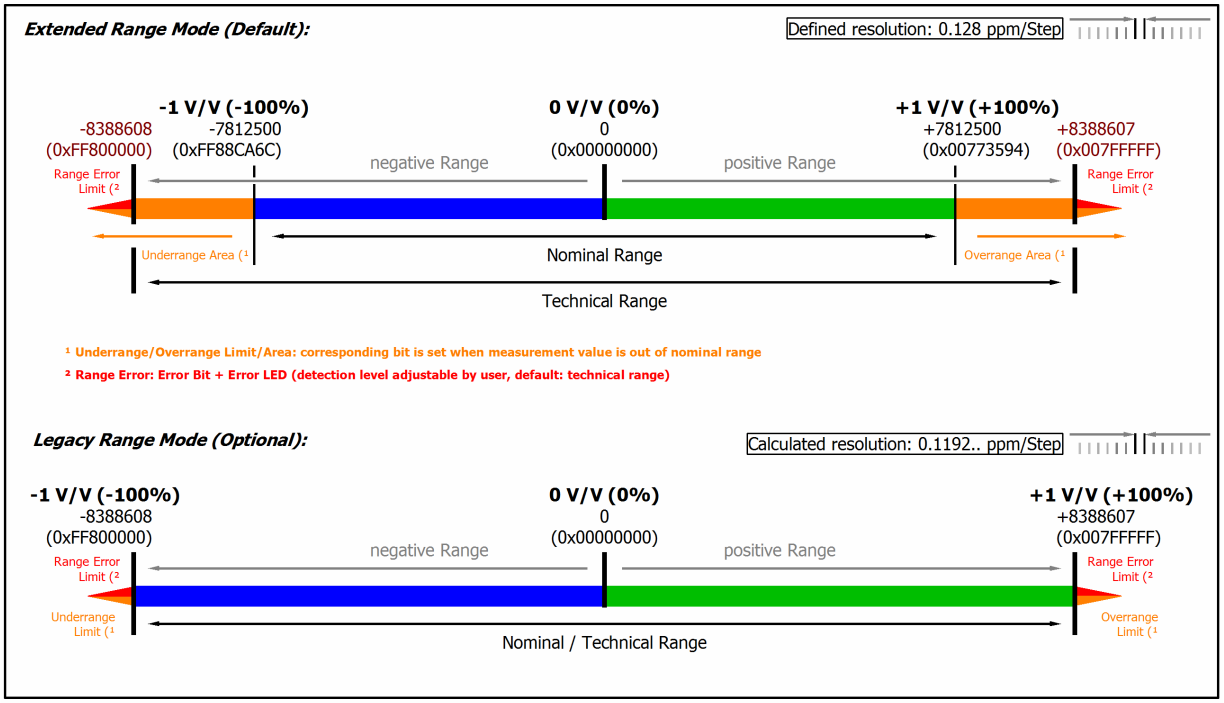


Fig. 15: Representation potentiometer measurement range

Note: In Extended Range Mode the Underrange/Overrange display in the PDO status has the character of an information/warning when the nominal measuring range is exceeded, i.e. no *Error* is displayed in the PDO status and LED. If the technical measuring range is also exceeded, *Error* = *TRUE* is also displayed. The detection limit for Underrange/Overrange *Error* can be set in the CoE.

In Legacy Range mode, an Underrange/Overrange event also leads to an *Error* in the PDO status.

4.2.6 Measurement SG 1/1 bridge (full bridge) 4/6-wire connection

Some notes to EPP3504-0023 full bridge measurement:

The nominal/technical measuring range is specified in "mV/V"; the maximum permitted supply voltage is 5 V. The maximum nominal measuring range that can be used for the bridge voltage is therefore ±32 mV/ V · 5 V = ±160 mV; the internal circuits are configured accordingly.

The internal measurement is ratiometric, i.e. the feed voltage and the bridge voltage are not measured absolutely, but as a ratio.

The integrated supply can be used as power supply. An external supply is permitted, as long as 5 V is not exceeded.

The box module has "real" and "compensated" measuring ranges:

- A "real" measuring range measures electrically as specified e.g., in the range 4 mV/V.
- A "compensated" measuring range helps in applications with a small signal (amplitude) and at the same time a high offset component. It measures in a fixed electrical range (i.e., subject to its electrical specification) and performs a "digital zoom", i.e., increases the resolution. The offset must be eliminated by the integrated ZeroOffset function of the box module.

The following is the specification given for the 6 wire connection. External line resistances are compensated by the 6 wire connection and the full bridge is detected directly from the measuring channel.
In the 4 wire connection, the box module generally has the same specification, but its view of the connected full bridge is clouded by the unclear and temperature-dependent lead resistances within cables and connectors. In this respect, the overall system "full bridge + leads + measurement channel" will practically not achieve specification values given below.

The lead resistances (cables, connectors, ...) have an effect especially on the gain error, also depending on the temperature. The gain error can be estimated by:

$(R_{+uv} (1 + \Delta T \cdot Tc_{Cu}) + R_{-uv}(1 + \Delta T \cdot Tc_{Cu}))/R_{nom}$ with Tc_{Cu}~3930 ppm/K, R_{nom}

e.g. 350 Ω and R_{+uv} or R_{-uv} lead resistances respectively.

The use of the measurement channel in the 6 wire connection is recommended, especially when significant resistors such as a lightning arrester are put into the line.

By a user-side adjustment with a connected bridge sensor, the measurement uncertainity related to gain and offset error can be significant reduced.

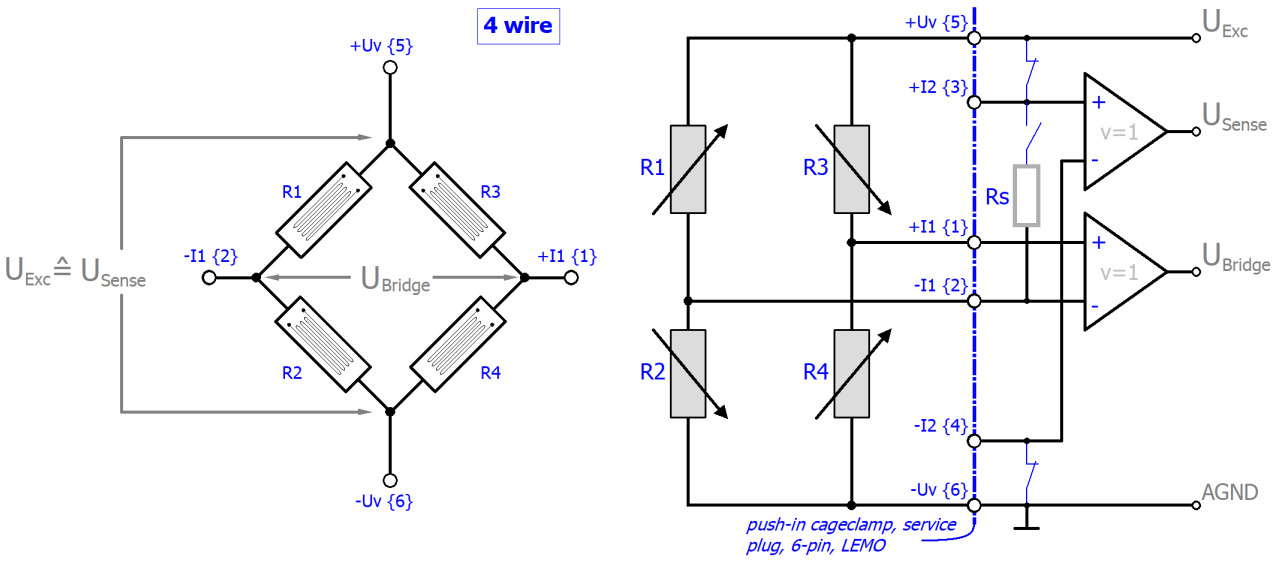
The integrated switcheable shunt resistor can be used to generate a predictable detuning or, in case of deviation, a correction factor.

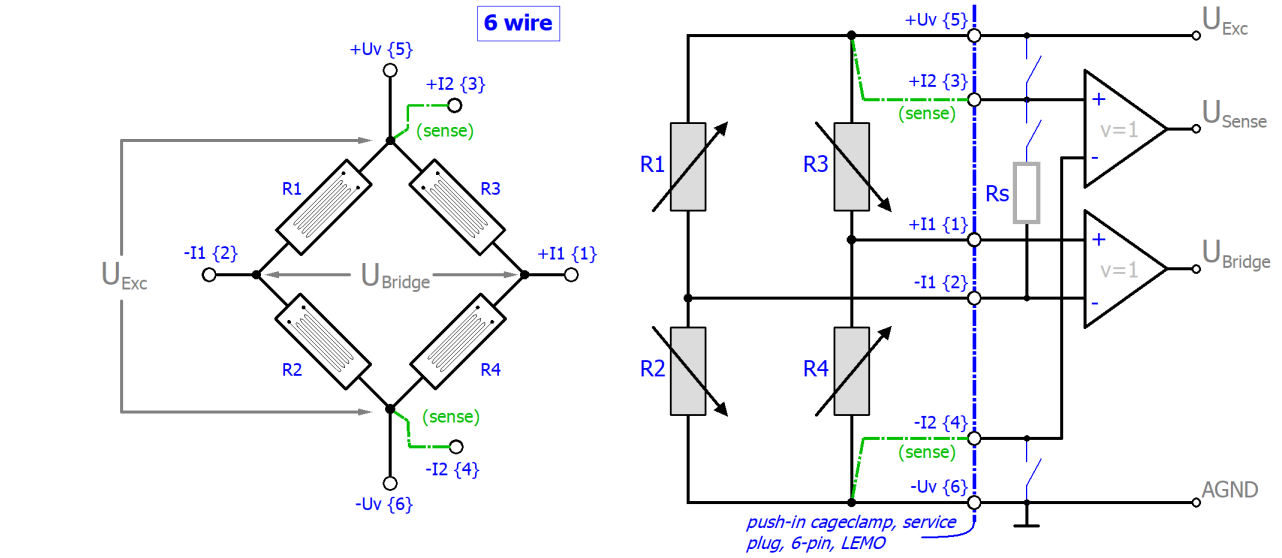
Note: specifications apply for 5 V SG excitation and symmetric 350R SG.

Note: data are valid from production week 01/ 2019 and

- for EPP3504-0023: HW04/ ERP3504-0022: HW06

Full bridge calculation:





The strain relationship (µStrain, µε) is as follows:

$$\frac{U_{Bridge}}{U_{Exc}} = \frac{Nk\varepsilon}{4}$$

$$N = 1, 2, 4, 1-\vartheta, 1+\vartheta, 2(1-\vartheta), 2(1+\vartheta)$$

Measurement mode		Measuring bridge/StrainGauge SG 1/1-Bridge 4/6-wire					
		32 mV/V	**8 mV/V**	**4 mV/V**	**2 mV/V**	**4 mV/V (comp.)** [5]	**2 mV/V (comp.)** [5]
Integrated power supply		1…5V adjustable, max. supply/Excitation 21 mA (internal electronic overload protection) therefore 120R SG: up to 2.5 V; 350R SG: up to 5.0 V					
Measuring range, nominal		-32 … +32 mV/V	-8 … +8 mV/V	-4 … +4 mV/V	-2 … +2 mV/V	-4 … +4 mV/V	-2 … +2 mV/V
Measuring range, end value (FSV)		32 mV/V	8 mV/V	4 mV/V	2 mV/V	4 mV/V	2 mV/V
Measuring range, technically usable		-34.359 … +34.359 mV/V	-8.590 … +8.590 mV/V	-4.295 … +4.295 mV/V	-2.147 … +2.147 mV/V	-4.295 … +4.295 mV/V	-2.147 … +2.147 mV/V
PDO resolution		24 bit (including sign)					
PDO LSB (Extended Range)		0.128 ppm					
PDO LSB (Legacy Range)		0.119… ppm					
Basic accuracy: Measuring deviation at 23°C, with averaging, typ. [2] [6]	without Offset	< ±0.003 $\%_{FSV}$ < ±30 ppm_{FSV} < ±0.96 µV/V	< ±0.006 $\%_{FSV}$ < ±60 ppm_{FSV} < ±0.48 µV/V	< ±0.0085 $\%_{FSV}$ < ±85 ppm_{FSV} < ±0.34 µV/V	< ±0.013 $\%_{FSV}$ < ±130 ppm_{FSV} < ±0.26 µV/V	< ±0.012 $\%_{FSV}$ < ±120 ppm_{FSV} < ±0.48 µV/V	< ±0.024 $\%_{FSV}$ < ±240 ppm_{FSV} < ±0.48 µV/V
	incl. Offset	< ±0.0075 $\%_{FSV}$ < ±75 ppm_{FSV} < ±2.4 µV/V	< ±0.02 $\%_{FSV}$ < ±200 ppm_{FSV} < ±1.6 µV/V	< ±0.039 $\%_{FSV}$ < ±390 ppm_{FSV} < ±1.56 µV/V	< ±0.0775 $\%_{FSV}$ < ±775 ppm_{FSV} < ±1.55 µV/V	< ±0.04 $\%_{FSV}$ < ±400 ppm_{FSV} < ±1.6 µV/V	< ±0.08 $\%_{FSV}$ < ±800 ppm_{FSV} < ±1.6 µV/V
Extended basic accuracy: Measuring deviation At 0…55°C, with averaging, typ. [2] [6]	without Offset	< ±0.0105 $\%_{FSV}$ < ±105 ppm_{FSV} < ±3.36 µV/V	< ±0.0265 $\%_{FSV}$ < ±265 ppm_{FSV} < ±2.12 µV/V	< ±0.0485 $\%_{FSV}$ < ±485 ppm_{FSV} < ±1.94 µV/V	< ±0.093 $\%_{FSV}$ < ±930 ppm_{FSV} < ±1.86 µV/V	< ±0.0525 $\%_{FSV}$ < ±525 ppm_{FSV} < ±2.1 µV/V	< ±0.105 $\%_{FSV}$ < ±1050 ppm_{FSV} < ±2.1 µV/V
	incl. Offset	< ±0.013 $\%_{FSV}$ < ±130 ppm_{FSV} < ±4.16 µV/V	< ±0.034 $\%_{FSV}$ < ±340 ppm_{FSV} < ±2.72 µV/V	< ±0.064 $\%_{FSV}$ < ±640 ppm_{FSV} < ±2.56 µV/V	< ±0.125 $\%_{FSV}$ < ±1250 ppm_{FSV} < ±2.5 µV/V	< ±0.068 $\%_{FSV}$ < ±680 ppm_{FSV} < ±2.72 µV/V	< ±0.1355 $\%_{FSV}$ < ±1355 ppm_{FSV} < ±2.71 µV/V
Offset/Zero point deviation (at 23°C)	E_{Offset}	< 70 ppm_{FSV}	< 190 ppm_{FSV}	< 380 ppm_{FSV}	< 760 ppm_{FSV}	< 380 ppm_{FSV}	< 760 ppm_{FSV}
Gain/scale/ amplification deviation (at 23°C)	E_{Gain}	< 24 ppm	< 50 ppm	< 70 ppm	< 110 ppm	< 100 ppm	< 200 ppm
Non-linearity over the whole measuring range	E_{Lin}	< 18 ppm_{FSV}	< 30 ppm_{FSV}	< 45 ppm_{FSV}	< 65 ppm_{FSV}	< 60 ppm_{FSV}	< 120 ppm_{FSV}
Repeatability, over 24 h, with averaging	E_{Rep}	< 5 ppm_{FSV}	< 10 ppm_{FSV}	< 15 ppm_{FSV}	< 25 ppm_{FSV}	< 20 ppm_{FSV}	< 40 ppm_{FSV}
Common-mode rejection ratio (without filter) [3]	DC	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
	50/60 Hz	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
	1 kHz	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
Common-mode rejection ratio (with 50/60 Hz FIR filter) [1] [3]	DC	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
	50/60 Hz	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
	1 kHz	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
Temperature coefficient, typ.	Tc_{Gain}	< 2.5 ppm/K	< 4 ppm/K	< 5 ppm/K	< 6 ppm/K	< 8 ppm/K	< 16 ppm/K
	Tc_{Offset}	< 2 ppm_{FSV}/K < 0.06 µV/V/K	< 7.5 ppm_{FSV}/K < 0.06 µV/V/K	< 15 ppm_{FSV}/K < 0.06 µV/V/K	< 30 ppm_{FSV}/K < 0.06 µV/V/K	< 15 ppm_{FSV}/K < 0.06 µV/V/K	< 30 ppm_{FSV}/K < 0.06 µV/V/K
Largest short-term deviation during a specified electrical interference test		tbd.	tbd.	tbd.	tbd.	tbd.	tbd.
Input impedance ±Input 1	Differential	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.

Measurement mode		Measuring bridge/StrainGauge SG 1/1-Bridge 4/6-wire					
		32 mV/V	8 mV/V	4 mV/V	2 mV/V	4 mV/V (comp.) 5)	2 mV/V (comp.) 5)
Input impedance ±Input 2	4-wire	No usage of this input in this mode					
	Differential	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.

[1]) the 50/60 Hz filter setting must be selected via the settings in the CoE depending on the local mains frequency (50 or 60 Hz); the respective nominal mains frequency then applies as the measuring point for common mode rejection or crosstalk.

[2]) In real bridge measurement, an offset adjustment is usually carried out after installation. The given offset specification of the box module is therefore practically irrelevant. Therefore, specification values with and without offset are given here. In practice, the offset component can be eliminated by the functions Features [▶ 000] and also Features [▶ 000] of the box module or in the controller by a higher-level tare function. The offset deviation of a bridge measurement over time can change, therefore Beckhoff recommends a regular offset adjustment or careful observation of the change.

[3]) Values related to a common mode interference between SGND and internal ground.

[5]) The channel measures electrically to 8 mV/V, but displays its measured value scaled to 2 or 4 mV/V. The Compensated function facilitates measurement of low levels even with high offset.

[6]) Calculated value according to the equation for estimating usability at the given ambient temperature point or estimating usability over the specified ambient temperature range in operation ($T_{ambient}$) as stated in chapter "General information on measuring accuracy/measurement uncertainty" [▶ 57]. In real use, for example at a relatively constant ambient temperature $T_{ambient}$, a lower (better) achievable uncertainty is attained. A specific calculation according to chapter "General information on measuring accuracy/ measurement uncertainty" is recommended, especially if the instrument allows a wider ambient temperature range in operation than 0...55 °C.

Noise (*preliminary data*):

Measurement mode		Measuring bridge/StrainGauge SG 1/1-Bridge 4/6-wire			
		32 mV/V	8 mV/V	4 mV/V	2 mV/V
Noise (without filtering, at 23°C)	$E_{Noise, PtP}$	< *85.0* ppm_{FSV} < *664* digits < *2.72* µV/V	< *300.0* ppm_{FSV} < *2344* digits < *2.4* µV/V	< *600.0* ppm_{FSV} < *4688* digits < *2.4* µV/V	< *1200.0* ppm_{FSV} < *9375* digits < *2.4* µV/V
	$E_{Noise, RMS}$	< *15.0* ppm_{FSV} < *117* digits < *0.48* µV/V	< *50.0* ppm_{FSV} < *391* digits < *0.4* µV/V	< *100.0* ppm_{FSV} < *781* digits < *0.4* µV/V	< *200.0* ppm_{FSV} < *1563* digits < *0.4* µV/V
	Max. SNR	> *96.5* dB	> *86.0* dB	> *80.0* dB	> *74.0* dB
	Noisedensity @1kHz	$< 6.79 \frac{nV/V}{\sqrt{Hz}}$	$< 5.66 \frac{nV/V}{\sqrt{Hz}}$	$< 5.66 \frac{nV/V}{\sqrt{Hz}}$	$< 5.66 \frac{nV/V}{\sqrt{Hz}}$
Noise (with 50 Hz FIR filter, at 23°C) 1)	$E_{Noise, PtP}$	< *12.0* ppm_{FSV} < *94* digits < *0.38* µV/V	< *30.0* ppm_{FSV} < *234* digits < *0.24* µV/V	< *60.0* ppm_{FSV} < *469* digits < *0.24* µV/V	< *120.0* ppm_{FSV} < *938* digits < *0.24* µV/V
	$E_{Noise, RMS}$	< *2.0* ppm_{FSV} < *16* digits < *0.06* µV/V	< *5.0* ppm_{FSV} < *39* digits < *0.04* µV/V	< *10.0* ppm_{FSV} < *78* digits < *0.04* µV/V	< *20.0* ppm_{FSV} < *156* digits < *0.04* µV/V
	Max. SNR	> *114.0* dB	> *106.0* dB	> *100.0* dB	> *94.0* dB

4.2.7 Measurement SG 1/2 bridge (half bridge) 3/5-wire connection

Some notes to EPP3504-0023 half bridge measurement:

The nominal/technical measuring range is specified in "mV/V"; the maximum permitted supply voltage is 5 V. The maximum nominal measuring range that can be used for the bridge voltage is therefore ±16 mV/V · 5 V = ±80 mV; the internal circuits are designed for the 160 mV of the full bridge measurement.

The internal measurement is ratiometric, i.e. the feed voltage and the bridge voltage are not measured absolutely, but as a ratio.

The integrated supply can be used as power supply. An external supply is permitted, as long as 5 V is not exceeded.

The box module has "real" and "compensated" measuring ranges:

- A "real" measuring range measures electrically as specified e.g., in the range 4 mV/V.
- A "compensated" measuring range helps in applications with a small signal (amplitude) and at the same time a high offset component. It measures in a fixed electrical range (i.e., subject to its electrical specification) and performs a "digital zoom", i.e., increases the resolution. The offset must be eliminated by the integrated ZeroOffset function of the box module.

The following is the specification given for the 5 wire connection. External line resistances are compensated by the 5 wire connection and the half bridge is detected directly from the measuring channel.
In the 3 wire connection, the box module generally has the same specification, but its view of the connected half bridge is clouded by the unclear and temperature-dependent lead resistances within cables and connectors. In this respect, the overall system "half bridge + leads + measurement channel" will practically not achieve specification values given below.

The lead resistances (cables, connectors, ...) have an effect especially on the gain error, also depending on the temperature. The gain error can be estimated by:

$(R_{+uv} (1 + \Delta T \cdot Tc_{Cu}) + R_{-uv}(1 + \Delta T \cdot Tc_{Cu}))/R_{nom}$ with Tc_{Cu}~3930 ppm/K, R_{nom}

e.g. 350 Ω and R_{+uv} or R_{-uv} lead resistances respectively.

The use of the measurement channel in the 5 wire connection is recommended.

By a user-side adjustment with a connected bridge sensor, the measurement uncertainity related to gain and offset error can be significant reduced.

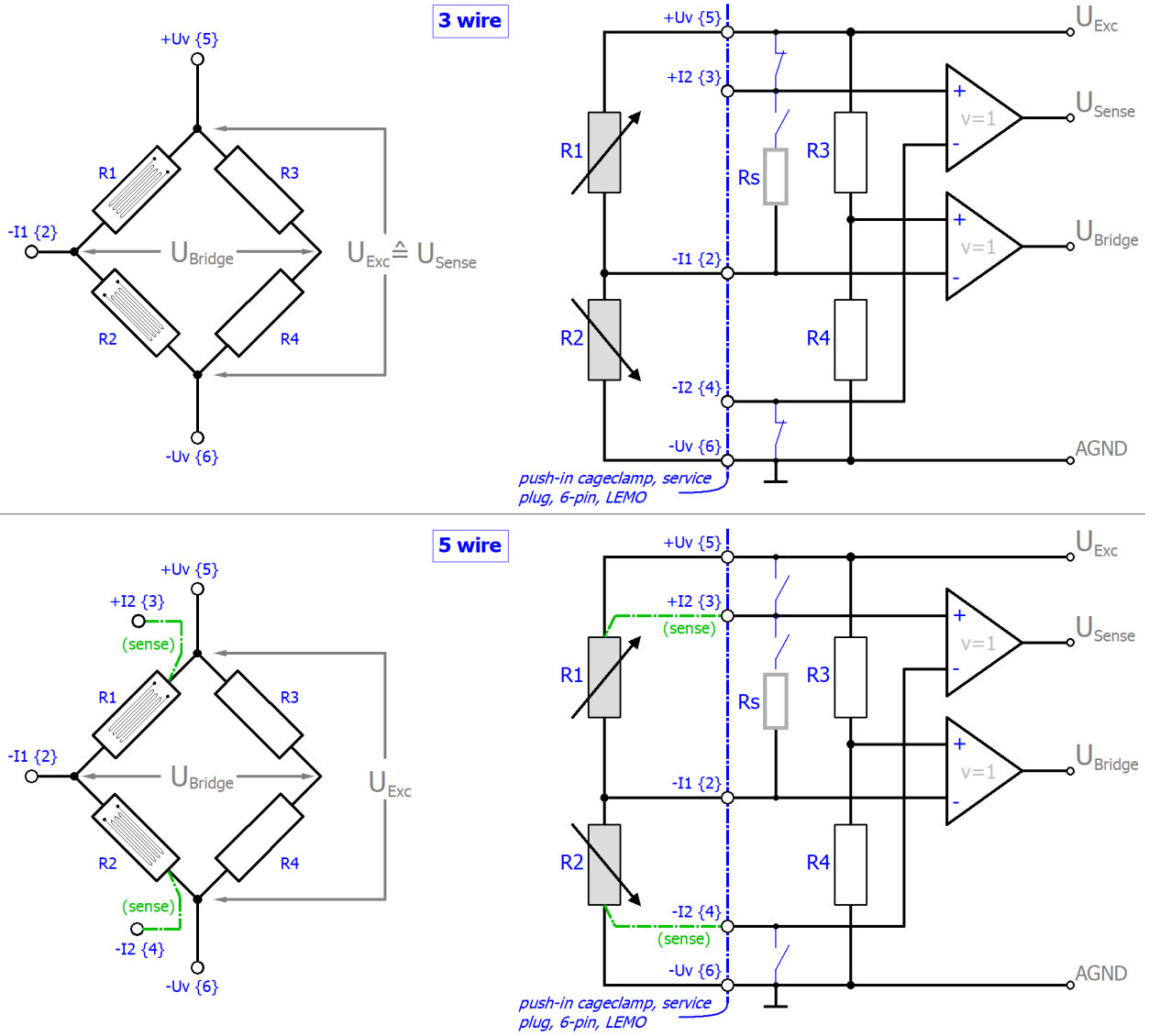
The integrated switcheable shunt resistor can be used to generate a predictable detuning or, in case of deviation, a correction factor.

Note: specifications apply for 3.5 V SG excitation and symmetric 350R SG.

Note: adjustment of the half-bridge measurement and thus validity of the data from production week 2018/50 and

- for EPP3504-0023: HW04/ ERP3504-0022: HW06

To calculate the $R_{1/2}$ half bridge:



$R_{3/4}$ are the internal switchable supplementary resistors of the box module. They have a high resistance of a few kΩ compared to $R_{1/2}$ and thus do not significantly load the internal supply.

Other half-bridge configurations (e.g. $R_{1/4}$ or $R_{1/3}$ variable) cannot be connected.

The strain relationship (µStrain, µε) is as follows:

$$\frac{U_{Bridge}}{U_{Exc}} = \frac{Nk\varepsilon}{4}$$

$$N = 1, 2, 4, 1 - \vartheta, 1 + \vartheta$$

N should be chosen based on the mechanical configuration of the variable resistors (Poisson, 2 active uniaxial, …). The channel value (PDO) is interpreted directly [mV/V].

Measurement mode		Measuring bridge/StrainGauge SG 1/2-Bridge 5/3-wire					
		16 mV/V	**8 mV/V** [1]	**4 mV/V** [1]	**2 mV/V** [1]	**4 mV/V (comp.)** [1] [5]	**2 mV/V (comp.)** [1] [5]
Integrated power supply		1…5V adjustable, max. power supply/Excitation 21 mA (internal electronic overload protection) therefore 120R SG: up to 2.5 V; 350R SG: up to 5.0 V					
Measuring range, nominal		-16 … 16 mV/V	-8 … 8 mV/V	-4 … 4 mV/V	-2 … 2 mV/V	-4 … 4 mV/V	-2 … 2 mV/V
Measuring range, end value (FSV)		16 mV/V	8 mV/V	4 mV/V	2 mV/V	4 mV/V	2 mV/V
Measuring range, technically usable		-17.179 … 17.179 mV/V	-8.589 … 8.589 mV/V	-4.294 … 4.294 mV/V	-2.147 … 2.147 mV/V	-4.294 … 4.294 mV/V	-2.147 … 2.147 mV/V
PDO resolution		24 bit (including sign)					
PDO LSB (Extended Range)		0.128 ppm					
PDO LSB (Legacy Range)		0.119… ppm					
Basic accuracy: Measuring deviation at 23°C, with averaging, typ. [2] [6]	without Offset	< ±0.011 $\%_{FSV}$ < ±110 ppm_{FSV} < ±1.76 µV/V	< ±0.022 $\%_{FSV}$ < ±220 ppm_{FSV} < ±1.76 µV/V	< ±0.044 $\%_{FSV}$ < ±440 ppm_{FSV} < ±1.76 µV/V	< ±0.0925 $\%_{FSV}$ < ±925 ppm_{FSV} < ±1.85 µV/V	< ±0.044 $\%_{FSV}$ < ±440 ppm_{FSV} < ±1.76 µV/V	< ±0.088 $\%_{FSV}$ < ±880 ppm_{FSV} < ±1.76 µV/V
	incl. Offset	< ±0.04 $\%_{FSV}$ < ±400 ppm_{FSV} < ±6.4 µV/V	< ±0.075 $\%_{FSV}$ < ±750 ppm_{FSV} < ±6 µV/V	< ±0.14 $\%_{FSV}$ < ±1400 ppm_{FSV} < ±5.6 µV/V	< ±0.27 $\%_{FSV}$ < ±2700 ppm_{FSV} < ±5.4 µV/V	< ±0.15 $\%_{FSV}$ < ±1500 ppm_{FSV} < ±6 µV/V	< ±0.3 $\%_{FSV}$ < ±3000 ppm_{FSV} < ±6 µV/V
Extended basic accuracy: Measuring deviation at 0… 55°C, with averaging, typ. [2] [6]	without Offset	< ±0.049 $\%_{FSV}$ < ±490 ppm_{FSV} < ±7.84 µV/V	< ±0.082 $\%_{FSV}$ < ±820 ppm_{FSV} < ±6.56 µV/V	< ±0.149 $\%_{FSV}$ < ±1490 ppm_{FSV} < ±5.96 µV/V	< ±0.2955 $\%_{FSV}$ < ±2955 ppm_{FSV} < ±5.91 µV/V	< ±0.164 $\%_{FSV}$ < ±1640 ppm_{FSV} < ±6.56 µV/V	< ±0.3275 $\%_{FSV}$ < ±3275 ppm_{FSV} < ±6.55 µV/V
	incl. Offset	< ±0.0645 $\%_{FSV}$ < ±645 ppm_{FSV} < ±10.32 µV/V	< ±0.113 $\%_{FSV}$ < ±1130 ppm_{FSV} < ±9.04 µV/V	< ±0.2065 $\%_{FSV}$ < ±2065 ppm_{FSV} < ±8.26 µV/V	< ±0.403 $\%_{FSV}$ < ±4030 ppm_{FSV} < ±8.06 µV/V	< ±0.2255 $\%_{FSV}$ < ±2255 ppm_{FSV} < ±9.02 µV/V	< ±0.4505 $\%_{FSV}$ < ±4505 ppm_{FSV} < ±9.01 µV/V
Offset/Zero point deviation (at 23°C)	E_{Offset}	< 385 ppm_{FSV}	< 715 ppm_{FSV}	< 1325 ppm_{FSV}	< 2530 ppm_{FSV}	< 1430 ppm_{FSV}	< 2860 ppm_{FSV}
Gain/scale/ amplification deviation (at 23°C)	E_{Gain}	< 70 ppm	< 130 ppm	< 260 ppm	< 510 ppm	< 260 ppm	< 520 ppm
Non-linearity over the whole measuring range	E_{Lin}	< 85 ppm_{FSV}	< 175 ppm_{FSV}	< 350 ppm_{FSV}	< 760 ppm_{FSV}	< 350 ppm_{FSV}	< 700 ppm_{FSV}
Repeatability, over 24 h, with averaging	E_{Rep}	< 12 ppm_{FSV}	< 25 ppm_{FSV}	< 50 ppm_{FSV}	< 120 ppm_{FSV}	< 50 ppm_{FSV}	< 100 ppm_{FSV}
Temperature coefficient, typ.	Tc_{Gain}	< 5 ppm/K	< 8 ppm/K	< 15 ppm/K	< 25 ppm/K	< 16 ppm/K	< 32 ppm/K
	Tc_{Offset}	< 15 ppm_{FSV}/K < 0.24 µV/V/K	< 25 ppm_{FSV}/K < 0.2 µV/V/K	< 45 ppm_{FSV}/K < 0.18 µV/V/K	< 90 ppm_{FSV}/K < 0.18 µV/V/K	< 50 ppm_{FSV}/K < 0.2 µV/V/K	< 100 ppm_{FSV}/K < 0.2 µV/V/K
Common-mode rejection ratio (without filter) [3]	DC	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
	50/60 Hz:	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
	1 kHz	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
Common-mode rejection ratio (with 50/60 Hz: FIR Filter) [3] [4]	DC	DC: tbd. nV/V / V	DC: tbd. nV/V / V	DC: tbd. nV/V / V	DC: tbd. nV/V / V	DC: tbd. nV/V / V	DC: tbd. nV/V / V
	50/60 Hz:	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
	1 kHz	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V	tbd. µV/V / V
Largest short-term deviation during a specified electrical interference test		tbd.	tbd.	tbd.	tbd.	tbd.	tbd.
Input impedance ±Input 1 (internal resistance)	Differential	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.

Measurement mode		Measuring bridge/StrainGauge SG 1/2-Bridge 5/3-wire					
		16 mV/V	8 mV/V [1]	4 mV/V [1]	2 mV/V [1]	4 mV/V (comp.) [1] [5]	2 mV/V (comp.) [1] [5]
Input impedance ±Input 2 (internal resistance)	3-wire	No usage of this input in this mode					
	Differential	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.	tbd.	tbd.

[1]) the 50/60 Hz filter setting must be selected via the settings in the CoE depending on the local mains frequency (50 or 60 Hz); the respective nominal mains frequency then applies as the measuring point for common mode rejection or crosstalk.

[2]) In real bridge measurement, an offset adjustment is usually carried out after installation. The given offset specification of the box module is therefore practically irrelevant. Therefore, specification values with and without offset are given here. In practice, the offset component can be eliminated by the functions Features [▶ 000] and also Features [▶ 000] of the box module or in the controller by a higher-level tare function. The offset deviation of a bridge measurement over time can change, therefore Beckhoff recommends a regular offset adjustment or careful observation of the change.

[3]) Values related to a common mode interference between SGND and internal ground.

[5]) The channel measures electrically to 8 mV/V, but displays its measured value scaled to 2 or 4 mV/V. The Compensated function facilitates measurement of low levels even with high offset.

[6]) Calculated value according to the equation for estimating usability at the given ambient temperature point or estimating usability over the specified ambient temperature range in operation ($T_{ambient}$) as stated in chapter "General information on measuring accuracy/measurement uncertainty" [▶ 57]. In real use, for example at a relatively constant ambient temperature $T_{ambient}$, a lower (better) achievable uncertainty is attained. A specific calculation according to chapter "General information on measuring accuracy/ measurement uncertainty" is recommended, especially if the instrument allows a wider ambient temperature range in operation than 0...55 °C.

Noise (*preliminary data*):

Measurement mode		Measuring bridge/StrainGauge SG 1/2-Bridge 5/3-wire			
		16 mV/V	8 mV/V	4 mV/V	2 mV/V
Noise (without filtering, at 23°C)	$E_{Noise, PtP}$	< *600.0* ppm_{FSV} < *4688* digits < *9.6* µV/V	< *1200.0* ppm_{FSV} < *9375* digits < *9.6* µV/V	< *2400.0* ppm_{FSV} < *18750* digits < *9.6* µV/V	< *4800.0* ppm_{FSV} < *37500* digits < *9.6* µV/V
	$E_{Noise, RMS}$	< *100.0* ppm_{FSV} < *781* digits < *1.6* µV/V	< *200.0* ppm_{FSV} < *1563* digits < *1.6* µV/V	< *400.0* ppm_{FSV} < *3125* digits < *1.6* µV/V	< *800.0* ppm_{FSV} < *6250* digits < *1.6* µV/V
	Max. SNR	> *80.0* dB	> *74.0* dB	> *68.0* dB	> *61.9* dB
	Noisedensity @1kHz	< *22.63* $\frac{nV/V}{\sqrt{Hz}}$	< *22.63* $\frac{nV/V}{\sqrt{Hz}}$	< *22.63* $\frac{nV/V}{\sqrt{Hz}}$	< *22.63* $\frac{nV/V}{\sqrt{Hz}}$
Noise (with 50 Hz FIR filter, at 23°C) [1]	$E_{Noise, PtP}$	< *35.0* ppm_{FSV} < *273* digits < *0.56* µV/V	< *70.0* ppm_{FSV} < *547* digits < *0.56* µV/V	< *140.0* ppm_{FSV} < *1094* digits < *0.56* µV/V	< *280.0* ppm_{FSV} < *2188* digits < *0.56* µV/V
	$E_{Noise, RMS}$	< *6.0* ppm_{FSV} < *47* digits < *0.1* µV/V	< *12.0* ppm_{FSV} < *94* digits < *0.1* µV/V	< *22.0* ppm_{FSV} < *172* digits < *0.09* µV/V	< *45.0* ppm_{FSV} < *352* digits < *0.09* µV/V
	Max. SNR	> *104.4* dB	> *98.4* dB	> *93.2* dB	> *86.9* dB

NOTICE

Transition resistances of the connection contacts

The transition resistance values of the connection contacts affect the measurement. The measuring accuracy can be further increased by a user-side adjustment with the signal connection plugged in.

Validity of property values

The resistor of the bridge is positioned parallel to the internal resistor of the box module and leads to an offset shifting respectively. The Beckhoff factory calibration will be carried out with the half bridge 350 Ω, thus the values specified above are directly valid for the 350 Ω half bridge. By connection of another dimensioned half-bridge is to:

- perform a balancing (offset correction) by the box module itself or the control/PLC on application side
- or the abstract offset error have to be entered into the balancing parameter S0 of the box module. Example: a 350 Ω half bridge correlates by the compensated effect of the input resistor (2 MΩ) during factory calibration 0.26545 $\%_{FSV}$ (16 mV/V), that corresponds to 20738 digits.

4.2.8 Measurement SG 1/4 bridge (quarter-bridge) 2/3-wire connection

Notes

- Quarter-bridge measurement in 2-wire operation is not recommended in practice. The normal copper supply cables with their own resistance (e.g. ~17 mΩ/m with 1 mm² wire) and their very high temperature sensitivity (~4000 ppm/K, ~0.4%/K) have a considerable effect on the calculation and can only be corrected by continuous offset and gain adjustment. Only 3-wire operation should be used.
- Specifications apply to 5 V excitation.
  The specification deteriorates at lower excitation voltage; Beckhoff does not have detailed information on this.
  If a lower excitation voltage is desired for reasons of sensor self-heating, the excitation voltage can be temporarily switched on/off for non-continuous measurements (clocked operation). Switching on/off must be done from the controller via ADS access to the CoE 0x80n0:02.
- Specifications apply to short cables, e.g. 5 m
  In 3-wire mode, the lead resistances increase the resistance seen by the measuring device and thus reduce the measurable signal level; the noise component increases accordingly. The longer the supply cable, the more difficult it is to evaluate the signal; there is no theoretical maximum cable length.
- Specifications only apply when using wire end sleeves and for cross-sections of 0.5 mm² or more. For smaller cross-sections, increased transition resistance is to be expected.
- Avoid repeated insertion/extraction of the push-in connectors in quarter-bridge operation, since this may increase the transition resistance.
- Integrated supply: 2...5V adjustable, max. supply/excitation 21 mA (internal electronic overload protection).
  Note: effectively only half the voltage is present at the quarter-bridge due to the internally switched bridge supplement.
- data are valid from production week 21/ 2019 and for EPP3504-0023: HW04/ ERP3504-0022: HW06

By a user-side adjustment with a connected bridge sensor, the measurement uncertainity related to gain and offset error can be significant reduced.

The integrated switcheable shunt resistor can be used to generate a predictable detuning or, in case of deviation, a correction factor.

To calculate the quarter-bridge:

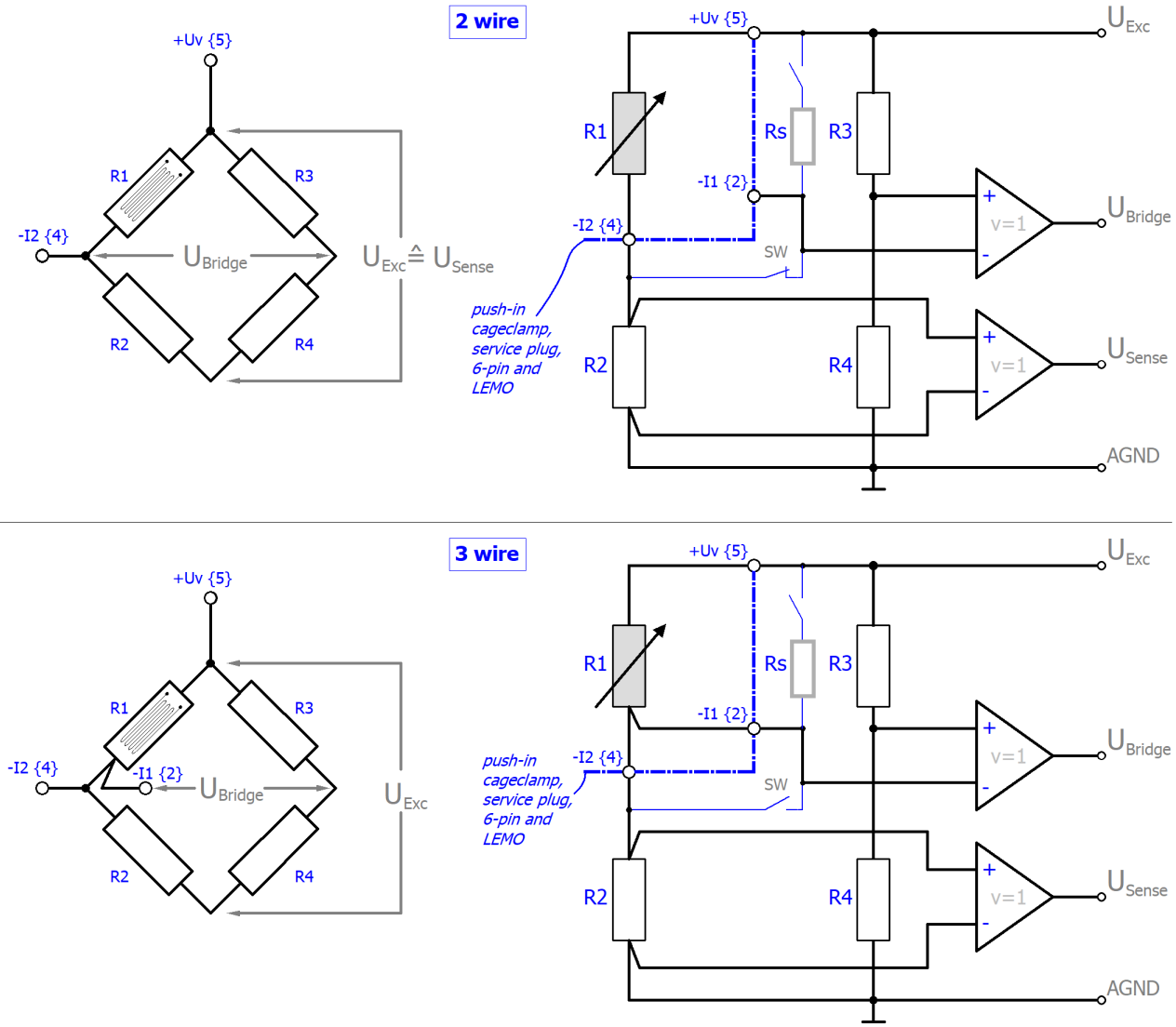


Fig. 16: Connection of the quarter bridge

Explanation:

- R1: external quarter-bridge resistor, nominally 120/350/1000 Ω
- R2: internal supplementary resistor, is set to the same value as R1 after the CoE setting "Interface", and is therefore also 120, 350 or 1000 Ω
- R3, R4: high-resistance internal bridge supplementary resistors, therefore, do not significantly load the internal supply
- Rs: switchable shunt resistor
- SW: internal switch for 2/3-wire operation; open: 3-wire operation

The strain relationship (µStrain, µε) is as follows:

$$\frac{U_{Bridge}}{U_{Exc}} = \frac{N\Delta R_1}{4R_1} = \frac{Nk\varepsilon}{4}$$

$$N = 1$$

For the quarter-bridge, N=1 always applies.

The relationship between U_{Bridge}/U_{Exc} and ΔR_1 is non-linear:

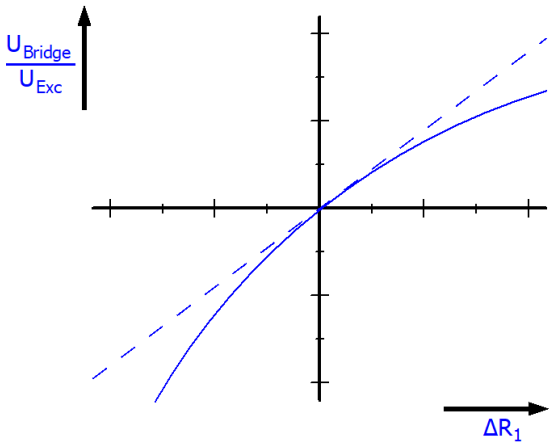


Fig. 17: Relationship between UBridge/UExc and ΔR_1

The EPP3504/ ERP3504 devices apply internal linearization so that the output is already linearized

$$\text{PDO [mV/V]} = \frac{U_{Bridge}}{U_{Exc}'} = \frac{\Delta R_1}{4R_1}$$

since the internal calculation is based on U_{Exc}'.

Measurement mode		Measuring bridge/StrainGauge SG 1/4-bridge 120 Ω 2/3-wire			
		32 mV/V	**8 mV/V**	**4 mV/V (comp.) [5]**	**2 mV/V (comp.) [5]**
Measuring range, nominal		±32 mV/V [corresponds to ±64,000 µε at K=2] 120 ±15.36 Ω	±8 mV/V [corresponds to ±16,000 µε at K=2] 120 ±3.84 Ω	±4 mV/V [corresponds to ±8,000 µε at K=2] 120 ±1.92 Ω	±2 mV/V [corresponds to ±4,000 µε at K=2] 120 ±0.96 Ω
Measuring range, end value (FSV)		32 mV/V	8 mV/V	4 mV/V	2 mV/V
Measuring range, technically usable		±34.359... mV/V	±8.589... mV/V	±4.294... mV/V	±2.147... mV/V
PDO resolution		24 bit (including sign)			
PDO LSB (Extended Range)		0.128 ppm 4.096 nV/V	0.128 ppm 1.024 nV/V	0.128 ppm 0.512 nV/V	0.128 ppm 0.256 nV/V
PDO LSB (Legacy Range)		0.119… ppm 3.814.. nV/V	0.119… ppm 0.9535 nV/V	0.119… ppm 0.47675 nV/V	0.119… ppm 0.238375 nV/V
Basic accuracy: Measuring deviation at 23°C, with averaging, typ. [2] [6]	without Offset	< ±0.026 $\%_{FSV}$ < ±260 ppm_{FSV} < ±8.3 µV/V	< ±0.08 $\%_{FSV}$ < ±800 ppm_{FSV} < ±6.4 µV/V	< ±0.16 $\%_{FSV}$ < ±1600 ppm_{FSV} < ±6.4 µV/V	< ±0.32 $\%_{FSV}$ < ±3200 ppm_{FSV} < ±6.4 µV/V
	incl. Offset	< ±0.1 $\%_{FSV}$ < ±1000 ppm_{FSV} < ±32 µV/V	< ±0.4 $\%_{FSV}$ < ±4000 ppm_{FSV} < ±32 µV/V	< ±0.8 $\%_{FSV}$ < ±8000 ppm_{FSV} < ±32 µV/V	< ±1.6 $\%_{FSV}$ < ±16000 ppm_{FSV} < ±32 µV/V
Extended basic accuracy: Measuring deviation at 0… 55°C, with averaging, typ. [2] [6]	without Offset	< ±0.164 $\%_{FSV}$ < ±1640 ppm_{FSV} < ±52.5 µV/V	< ±0.565 $\%_{FSV}$ < ±5650 ppm_{FSV} < ±45.2 µV/V	< ±1.1295 $\%_{FSV}$ < ±11295 ppm_{FSV} < ±45.2 µV/V	< ±2.2585 $\%_{FSV}$ < ±22585 ppm_{FSV} < ±45.2 µV/V
	incl. Offset	< ±0.1995 $\%_{FSV}$ < ±1995 ppm_{FSV} < ±63.8 µV/V	< ±0.718 $\%_{FSV}$ < ±7180 ppm_{FSV} < ±57.4 µV/V	< ±1.436 $\%_{FSV}$ < ±14360 ppm_{FSV} < ±57.4 µV/V	< ±2.872 $\%_{FSV}$ < ±28720 ppm_{FSV} < ±57.4 µV/V
Offset/Zero point deviation (at 23°C) [4]	E_{Offset}	< 960 ppm_{FSV}	< 3920 ppm_{FSV}	< 7840 ppm_{FSV}	< 15680 ppm_{FSV}
Gain/scale/ amplification deviation (at 23°C)	E_{Gain}	< 160 ppm	< 520 ppm	< 1040 ppm	< 2080 ppm
Non-linearity over the whole measuring range	E_{Lin}	< 200 ppm_{FSV}	< 600 ppm_{FSV}	< 1200 ppm_{FSV}	< 2400 ppm_{FSV}
Repeatability, over 24 h, with averaging	E_{Rep}	< 25 ppm_{FSV}	< 100 ppm_{FSV}	< 200 ppm_{FSV}	< 400 ppm_{FSV}
Common-mode rejection ratio (without filter) [3]		tbd.	tbd.	tbd.	tbd.
Common-mode rejection ratio (with 50/60 Hz FIR filter) [1] [3]		tbd.	tbd.	tbd.	tbd.
Temperature coefficient, typ.	Tc_{Gain}	< 20 ppm/K	< 48 ppm/K	< 96 ppm/K	< 192 ppm/K
	Tc_{Offset}	< 50 ppm_{FSV}/K < 1.6 µV/V/K	< 180 ppm_{FSV}/K < 1.44 µV/V/K	< 360 ppm_{FSV}/K < 1.44 µV/V/K	< 720 ppm_{FSV}/K < 1.44 µV/V/K
Largest short-term deviation during a specified electrical interference test		tbd. $\%_{FSV}$	tbd. $\%_{FSV}$	tbd. $\%_{FSV}$	tbd. $\%_{FSV}$
Input impedance ±Input 1	Differential	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.
Input impedance ±Input 2	3-wire				
	Differential	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.

Noise (*preliminary data*):

Measurement mode		Measuring bridge/StrainGauge SG 1/4-bridge 120 Ω 2/3-wire			
		32 mV/V	**8 mV/V**	**4 mV/V (comp.) 5)**	**2 mV/V (comp.) 5)**
Noise (without filtering, at 23°C)	$E_{Noise, PtP}$	< *285.0* ppm_{FSV} < *2227* digits < *9.12* µV/V	< *1000.0* ppm_{FSV} < *7813* digits < *8.0* µV/V	< *2000.0* ppm_{FSV} < *15625* digits < *8.0* µV/V	< *4000.0* ppm_{FSV} < *31250* digits < *8.0* µV/V
	$E_{Noise, RMS}$	< *50.0* ppm_{FSV} < *391* digits < *1.6* µV/V	< *150.0* ppm_{FSV} < *1172* digits < *1.2* µV/V	< *300.0* ppm_{FSV} < *2344* digits < *1.2* µV/V	< *600.0* ppm_{FSV} < *4688* digits < *1.2* µV/V
	Max. SNR	> *86.0* dB	> *76.5* dB	> *70.5* dB	> *64.4* dB
	Noisedensity @1kHz	< *0.02* $\frac{\mu V/V}{\sqrt{Hz}}$	< *0.02* $\frac{\mu V/V}{\sqrt{Hz}}$	< *0.02* $\frac{\mu V/V}{\sqrt{Hz}}$	< *0.02* $\frac{\mu V/V}{\sqrt{Hz}}$
Noise (with 50 Hz FIR filter, at 23°C) 1)	$E_{Noise, PtP}$	< *20.0* ppm_{FSV} < *156* digits < *0.64* µV/V	< *60.0* ppm_{FSV} < *469* digits < *0.48* µV/V	< *120.0* ppm_{FSV} < *938* digits < *0.48* µV/V	< *240.0* ppm_{FSV} < *1875* digits < *0.48* µV/V
	$E_{Noise, RMS}$	< *4.0* ppm_{FSV} < *31* digits < *0.13* µV/V	< *12.0* ppm_{FSV} < *94* digits < *0.1* µV/V	< *24.0* ppm_{FSV} < *188* digits < *0.1* µV/V	< *48.0* ppm_{FSV} < *375* digits < *0.1* µV/V
	Max. SNR	> *108.0* dB	> *98.4* dB	> *92.4* dB	> *86.4* dB

Measurement mode		Measuring bridge/StrainGauge SG 1/4-bridge 350 Ω 2/3-wire			
		32 mV/V	**8 mV/V**	**4 mV/V (comp.) [5]**	**2 mV/V (comp.) [5]**
Measuring range, nominal		±32 mV/V [corresponds to ±64,000 µε at K=2] 350 ±44.8 Ω	±8 mV/V [corresponds to ±16,000 µε at K=2] 350 ±11.2 Ω	±4 mV/V [corresponds to ±8,000 µε at K=2] 350 ±5.6 Ω	±2 mV/V [corresponds to ±4,000 µε at K=2] 350 ±2.8 Ω
Measuring range, end value (FSV)		32 mV/V	8 mV/V	4 mV/V	2 mV/V
Measuring range, technically usable		±34.359... mV/V	±8.589... mV/V	±4.294... mV/V	±2.147... mV/V
PDO resolution		24 bit (including sign)			
PDO LSB (Extended Range)		0.128 ppm 4.096 nV/V	0.128 ppm 1.024 nV/V	0.128 ppm 0.512 nV/V	0.128 ppm 0.256 nV/V
PDO LSB (Legacy Range)		0.119… ppm 3.814.. nV/V	0.119… ppm 0.9535 nV/V	0.119… ppm 0.47675 nV/V	0.119… ppm 0.238375 nV/V
Basic accuracy: Measuring deviation at 23°C, with averaging, typ. [2) 6)]	without Offset	< ±0.022 $\%_{FSV}$ < ±220 ppm_{FSV} < ±7 µV/V	< ±0.08 $\%_{FSV}$ < ±800 ppm_{FSV} < ±6.4 µV/V	< ±0.16 $\%_{FSV}$ < ±1600 ppm_{FSV} < ±6.4 µV/V	< ±0.32 $\%_{FSV}$ < ±3200 ppm_{FSV} < ±6.4 µV/V
	incl. Offset	< ±0.1 $\%_{FSV}$ < ±1000 ppm_{FSV} < ±32 µV/V	< ±0.4 $\%_{FSV}$ < ±4000 ppm_{FSV} < ±32 µV/V	< ±0.8 $\%_{FSV}$ < ±8000 ppm_{FSV} < ±32 µV/V	< ±1.6 $\%_{FSV}$ < ±16000 ppm_{FSV} < ±32 µV/V
Extended basic accuracy: Measuring deviation at 0…55°C, with averaging, typ. [2) 6)]	without Offset	< ±0.0995 $\%_{FSV}$ < ±995 ppm_{FSV} < ±31.8 µV/V	< ±0.3715 $\%_{FSV}$ < ±3715 ppm_{FSV} < ±29.7 µV/V	< ±0.7425 $\%_{FSV}$ < ±7425 ppm_{FSV} < ±29.7 µV/V	< ±1.485 $\%_{FSV}$ < ±14850 ppm_{FSV} < ±29.7 µV/V
	incl. Offset	< ±0.144 $\%_{FSV}$ < ±1440 ppm_{FSV} < ±46.1 µV/V	< ±0.5565 $\%_{FSV}$ < ±5565 ppm_{FSV} < ±44.5 µV/V	< ±1.113 $\%_{FSV}$ < ±11130 ppm_{FSV} < ±44.5 µV/V	< ±2.2255 $\%_{FSV}$ < ±22255 ppm_{FSV} < ±44.5 µV/V
Offset/Zero point deviation (at 23°C) [4)]	E_{Offset}	< 970 ppm_{FSV}	< 3920 ppm_{FSV}	< 7840 ppm_{FSV}	< 15680 ppm_{FSV}
Gain/scale/ amplification deviation (at 23°C)	E_{Gain}	< 120 ppm	< 440 ppm	< 880 ppm	< 1760 ppm
Non-linearity over the whole measuring range	E_{Lin}	< 180 ppm_{FSV}	< 650 ppm_{FSV}	< 1300 ppm_{FSV}	< 2600 ppm_{FSV}
Repeatability, over 24 h, with averaging	E_{Rep}	< 25 ppm_{FSV}	< 100 ppm_{FSV}	< 200 ppm_{FSV}	< 400 ppm_{FSV}
Common-mode rejection ratio (without filter) [3)]		tbd.	tbd.	tbd.	tbd.
Common-mode rejection ratio (with 50/60 Hz FIR filter) [1) 3)]		tbd.	tbd.	tbd.	tbd.
Temperature coefficient, typ.	Tc_{Gain}	< 12 ppm/K	< 50 ppm/K	< 100 ppm/K	< 200 ppm/K
	Tc_{Offset}	< 30 ppm_{FSV}/K < 0.96 µV/V/K	< 110 ppm_{FSV}/K < 0.88 µV/V/K	< 220 ppm_{FSV}/K < 0.88 µV/V/K	< 440 ppm_{FSV}/K < 0.88 µV/V/K
Largest short-term deviation during a specified electrical interference test		tbd. $\%_{FSV}$	tbd. $\%_{FSV}$	tbd. $\%_{FSV}$	tbd. $\%_{FSV}$
Input impedance ±Input 1	Differential	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.
Input impedance ±Input 2	3-wire				
	Differential	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.

Noise (*preliminary data*):

Measurement mode		Measuring bridge/StrainGauge SG 1/4-bridge 350 Ω 2/3-wire			
		32 mV/V	**8 mV/V**	**4 mV/V (comp.) [5)]**	**2 mV/V (comp.) [5)]**
Noise (without filtering, at 23°C)	$E_{Noise, PtP}$	< *290.0* ppm_{FSV} < *2266* digits < *9.28* µV/V	< *1000.0* ppm_{FSV} < *7813* digits < *8.0* µV/V	< *2000.0* ppm_{FSV} < *15625* digits < *8.0* µV/V	< *4000.0* ppm_{FSV} < *31250* digits < *8.0* µV/V
	$E_{Noise, RMS}$	< *50.0* ppm_{FSV} < *391* digits < *1.6* µV/V	< *160.0* ppm_{FSV} < *1250* digits < *1.28* µV/V	< *320.0* ppm_{FSV} < *2500* digits < *1.28* µV/V	< *640.0* ppm_{FSV} < *5000* digits < *1.28* µV/V
	Max. SNR	> *86.0* dB	> *75.9* dB	> *69.9* dB	> *63.9* dB
	Noisedensity @1kHz	$< 0.02 \frac{\mu V/V}{\sqrt{Hz}}$	$< 0.02 \frac{\mu V/V}{\sqrt{Hz}}$	$< 0.02 \frac{\mu V/V}{\sqrt{Hz}}$	$< 0.02 \frac{\mu V/V}{\sqrt{Hz}}$
Noise (with 50 Hz FIR filter, at 23°C) [1)]	$E_{Noise, PtP}$	< *15.0* ppm_{FSV} < *117* digits < *0.48* µV/V	< *50.0* ppm_{FSV} < *391* digits < *0.4* µV/V	< *100.0* ppm_{FSV} < *781* digits < *0.4* µV/V	< *200.0* ppm_{FSV} < *1563* digits < *0.4* µV/V
	$E_{Noise, RMS}$	< *3.0* ppm_{FSV} < *23* digits < *0.1* µV/V	< *9.0* ppm_{FSV} < *70* digits < *0.07* µV/V	< *18.0* ppm_{FSV} < *141* digits < *0.07* µV/V	< *36.0* ppm_{FSV} < *281* digits < *0.07* µV/V
	Max. SNR	> *110.5* dB	> *100.9* dB	> *94.9* dB	> *88.9* dB

Measurement mode		Measuring bridge/StrainGauge SG 1/4-bridge 1 kΩ 2/3-wire			
		32 mV/V	**8 mV/V**	**4 mV/V (comp.) [5]**	**2 mV/V (comp.) [5]**
Measurement range, nominal		±32 mV/V [corresponds to ±64,000 µε at K=2] 1000 ±128 Ω	±8 mV/V [corresponds to ±16,000 µε at K=2] 1000 ±32 Ω	±4 mV/V [corresponds to ±8,000 µε at K=2] 1000 ±16 Ω	±2 mV/V [corresponds to ±4,000 µε at K=2] 1000 ±8 Ω
Measuring range, end value (FSV)		32 mV/V 128 Ω	8 mV/V 32 Ω	4 mV/V 16 Ω	2 mV/V 8 Ω
Measuring range, technically usable		±34.359... mV/V	±8.589... mV/V	±4.294... mV/V	±2.147... mV/V
PDO resolution		24 bit (including sign)			
PDO LSB (Extended Range)		0.128 ppm 4.096 nV/V	0.128 ppm 1.024 nV/V	0.128 ppm 0.512 nV/V	0.128 ppm 0.256 nV/V
PDO LSB (Legacy Range)		0.119… ppm 3.814.. nV/V	0.119… ppm 0.9535 nV/V	0.119… ppm 0.47675 nV/V	0.119… ppm 0.238375 nV/V
Basic accuracy: Measuring deviation at 23°C, with averaging, typ. [2] [6]	without Offset	< ±0.02 $\%_{FSV}$ < ±200 ppm_{FSV} < ±6.4 µV/V	< ±0.065 $\%_{FSV}$ < ±650 ppm_{FSV} < ±5.2 µV/V	< ±0.13 $\%_{FSV}$ < ±1300 ppm_{FSV} < ±5.2 µV/V	< ±0.26 $\%_{FSV}$ < ±2600 ppm_{FSV} < ±5.2 µV/V
	incl. Offset	< ±0.1 $\%_{FSV}$ < ±1000 ppm_{FSV} < ±32 µV/V	< ±0.4 $\%_{FSV}$ < ±4000 ppm_{FSV} < ±32 µV/V	< ±0.8 $\%_{FSV}$ < ±8000 ppm_{FSV} < ±32 µV/V	< ±1.6 $\%_{FSV}$ < ±16000 ppm_{FSV} < ±32 µV/V
Extended basic accuracy: Measuring deviation at 0… 55°C, with averaging, typ. [2] [6]	without Offset	< ±0.1855 $\%_{FSV}$ < ±1855 ppm_{FSV} < ±59.4 µV/V	< ±0.6975 $\%_{FSV}$ < ±6975 ppm_{FSV} < ±55.8 µV/V	< ±1.3945 $\%_{FSV}$ < ±13945 ppm_{FSV} < ±55.8 µV/V	< ±2.7885 $\%_{FSV}$ < ±27885 ppm_{FSV} < ±55.8 µV/V
	incl. Offset	< ±0.2205 $\%_{FSV}$ < ±2205 ppm_{FSV} < ±70.6 µV/V	< ±0.8415 $\%_{FSV}$ < ±8415 ppm_{FSV} < ±67.3 µV/V	< ±1.683 $\%_{FSV}$ < ±16830 ppm_{FSV} < ±67.3 µV/V	< ±3.366 $\%_{FSV}$ < ±33660 ppm_{FSV} < ±67.3 µV/V
Offset/Zero point deviation (at 23°C) [4]	E_{Offset}	< 980 ppm_{FSV}	< 3940 ppm_{FSV}	< 7880 ppm_{FSV}	< 15760 ppm_{FSV}
Gain/scale/ amplification deviation (at 23°C)	E_{Gain}	< 105 ppm	< 305 ppm	< 610 ppm	< 1220 ppm
Non-linearity over the whole measuring range	E_{Lin}	< 165 ppm_{FSV}	< 560 ppm_{FSV}	< 1120 ppm_{FSV}	< 2240 ppm_{FSV}
Repeatability, over 24 h, with averaging	E_{Rep}	< 25 ppm_{FSV}	< 120 ppm_{FSV}	< 240 ppm_{FSV}	< 480 ppm_{FSV}
Common-mode rejection ratio (without filter) [3]		tbd.	tbd.	tbd.	tbd.
Common-mode rejection ratio (with 50/60 Hz FIR filter) [1] [3]		tbd.	tbd.	tbd.	tbd.
Temperature coefficient, typ.	Tc_{Gain}	< 13 ppm/K	< 25 ppm/K	< 50 ppm/K	< 100 ppm/K
	Tc_{Offset}	< 60 ppm_{FSV}/K < 1.92 µV/V/K	< 230 ppm_{FSV}/K < 1.84 µV/V/K	< 460 ppm_{FSV}/K < 1.84 µV/V/K	< 920 ppm_{FSV}/K < 1.84 µV/V/K
Largest short-term deviation during a specified electrical interference test		tbd. $\%_{FSV}$	tbd. $\%_{FSV}$	tbd. $\%_{FSV}$	tbd. $\%_{FSV}$
Input impedance ±Input 1	Differential	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.
Input impedance ±Input 2	3-wire	No usage of this input in this mode			
	Differential	tbd.	tbd.	tbd.	tbd.
	CommonMode	tbd.	tbd.	tbd.	tbd.

Noise (*preliminary data*):

Measurement mode		Measuring bridge/StrainGauge SG 1/4-bridge 1 kΩ 2/3-wire			
		32 mV/V	**8 mV/V**	**4 mV/V (comp.) [5]**	**2 mV/V (comp.) [5]**
Noise (without filtering, at 23°C)	$E_{Noise, PtP}$	< *350.0* ppm_{FSV} < *2734* digits < *11.2* µV/V	< *820.0* ppm_{FSV} < *6406* digits < *6.56* µV/V	< *1640.0* ppm_{FSV} < *12813* digits < *6.56* µV/V	< *3280.0* ppm_{FSV} < *25625* digits < *6.56* µV/V
	$E_{Noise, RMS}$	< *70.0* ppm_{FSV} < *547* digits < *2.24* µV/V	< *140.0* ppm_{FSV} < *1094* digits < *1.12* µV/V	< *280.0* ppm_{FSV} < *2188* digits < *1.12* µV/V	< *560.0* ppm_{FSV} < *4375* digits < *1.12* µV/V
	Max. SNR	> *83.1* dB	> *77.1* dB	> *71.1* dB	> *65.0* dB
	Noisedensity @1kHz	$< 0.03 \frac{\mu V/V}{\sqrt{Hz}}$	$< 0.02 \frac{\mu V/V}{\sqrt{Hz}}$	$< 0.02 \frac{\mu V/V}{\sqrt{Hz}}$	$< 0.02 \frac{\mu V/V}{\sqrt{Hz}}$
Noise (with 50 Hz FIR filter, at 23°C) [1]	$E_{Noise, PtP}$	< *85.0* ppm_{FSV} < *664* digits < *2.72* µV/V	< *48.0* ppm_{FSV} < *375* digits < *0.38* µV/V	< *96.0* ppm_{FSV} < *750* digits < *0.38* µV/V	< *192.0* ppm_{FSV} < *1500* digits < *0.38* µV/V
	$E_{Noise, RMS}$	< *14.0* ppm_{FSV} < *109* digits < *0.45* µV/V	< *8.0* ppm_{FSV} < *63* digits < *0.06* µV/V	< *16.0* ppm_{FSV} < *125* digits < *0.06* µV/V	< *32.0* ppm_{FSV} < *250* digits < *0.06* µV/V
	Max. SNR	> *97.1* dB	> *101.9* dB	> *95.9* dB	> *89.9* dB

[1]) the 50/60 Hz filter setting must be selected via the settings in the CoE depending on the local mains frequency (50 or 60 Hz); the respective nominal mains frequency then applies as the measuring point for common mode rejection or crosstalk.

[2]) In real bridge measurement, an offset adjustment is usually carried out after installation. The given offset specification of the box module is therefore practically irrelevant. Therefore, specification values with and without offset are given here. In practice, the offset component can be eliminated by the functions Features [▶ 000] and also Features [▶ 000] of the box module or in the controller by a higher-level tare function. The offset deviation of a bridge measurement over time can change, therefore Beckhoff recommends a regular offset adjustment or careful observation of the change.

[3]) Values related to a common mode interference between SGND and internal ground.

[4]) The offset specification does not apply to 2-wire operation, since the offset is increased on the device side. Therefore, a system-side offset adjustment is recommended, see Tare- [▶ 000] or ZeroOffset function [▶ 000]. The final targeting basic acuuracy within the 2-wire operation is mainingly dependent by the quality of this system-side offset adjustment.

[5]) The channel measures electrically to 8 mV/V, but displays its measured value scaled to 2 or 4 mV/V. The Compensated function facilitates measurement of low levels even with high offset.

[6]) Calculated value according to the equation for estimating usability at the given ambient temperature point or estimating usability over the specified ambient temperature range in operation ($T_{ambient}$) as stated in chapter "General information on measuring accuracy/measurement uncertainty" [▶ 57]. In real use, for example at a relatively constant ambient temperature $T_{ambient}$, a lower (better) achievable uncertainty is attained. A specific calculation according to chapter "General information on measuring accuracy/ measurement uncertainty" is recommended, especially if the instrument allows a wider ambient temperature range in operation than 0...55 °C.

NOTICE

Transition resistances of the connection contacts

The transition resistance values of the connection contacts affect the measurement. The measuring accuracy can be further increased by a user-side adjustment with the signal connection plugged in.

The temperature sensitivity of the box module and thus of the measurement setup can be reduced if an external, more temperature-stable supplementary resistor is used for operation of the box module in half-bridge or even full-bridge mode instead of the internal supplementary resistor for quarter-bridge mode.

4.3 Process data interpretation

The entire measurement range is constituted as follows with regard to the output of the cyclic process data:

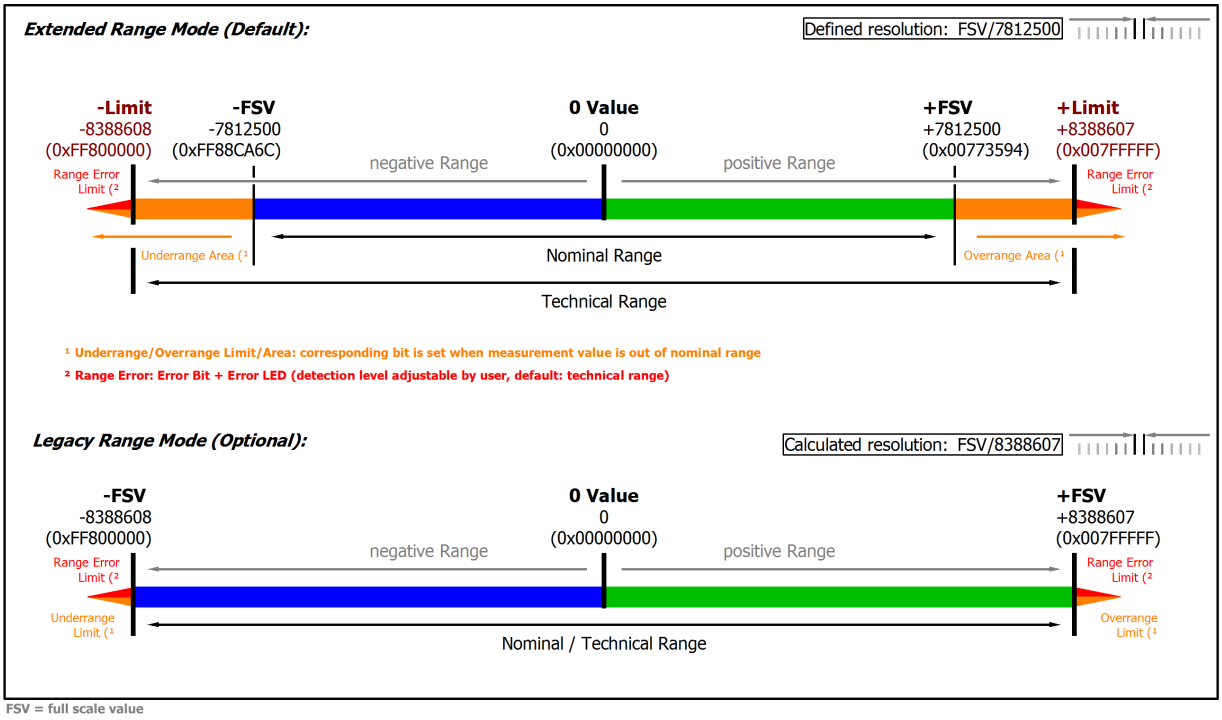


Fig. 18: Basic range of a process data value

The channel for this box module features an option to set the measuring range either to the conventional Beckhoff type, up until now: "nominal full-scale value = PDO end value: LegacyRange" or the new method "technical full-scale value = PDO end value: ExtendedRange".

- For Extended Range mode, the following applies:
  - technical full scale value = PDO end value 0x007FFFFF.
  - for information purposes, the channel can measure up to approx. 107% beyond the nominal range, although accuracy specifications etc. are then no longer valid.
  - outside the nominal measuring range, the Overrange or Underrange bit is set.
  - for further diagnosis, the error bit and the error LED are set if configurable limits are exceeded or not reached. By default, the limits are set to the technical measuring range, although they can be narrowed by the customer.
    Example: In the 4…20 mA measuring range, the lower limit is set to 0 mA, although it can be customized in the CoE by the customer, e.g. set to 3.6 mA, in order to enable earlier detection of potential sensor faults.
  - The Extended Range mode is the default setting for the box module ex works.
  - The mode is defined through the non-periodic rational LSB step size **and** an integer end value. This enables the step size to be used in a PLC program without a rounding error.
- For Legacy Range mode, the following applies:
  - nominal full scale value = PDO end value.
  - compatible with existing interface from EL30xx/EL31xx/EL36xx.
  - Overrange/Underrange, Error bit and Error LED are set simultaneously if the nominal/technical measuring range is exceeded or not met.
  - can be optionally activated in the box module.
  - this mode is defined by an integer end value; assuming that the LSB step is no longer a whole number.

4.4 General Information on Measuring Accuracy/ Measurement Uncertainty

For basic information regarding the explanatory notes below, please refer to chapter “Notes on analog data values” under Further documentation for I/O components with analog in and outputs [▶ 252], particularly for full scale value.

This guidance should be read and followed in order to save extra work, time and, presumably, money.

In-depth familiarization with these instructions will make it easy to master this technology and thus facilitate your work.

Basic information on measurement technology:

Measuring devices are used to try to determine the true value of a measured variable, e.g. ambient temperature, with the amount of effort put into this varying. For various practical reasons this cannot be performed conclusively. Depending on the work involved, the measurement/measured value is subject to a random measuring error that cannot be eliminated. With its practically determined specification data, Beckhoff provides an approach that can be used to calculate the residual measurement uncertainty in the individual case. The following paragraphs elucidate this.

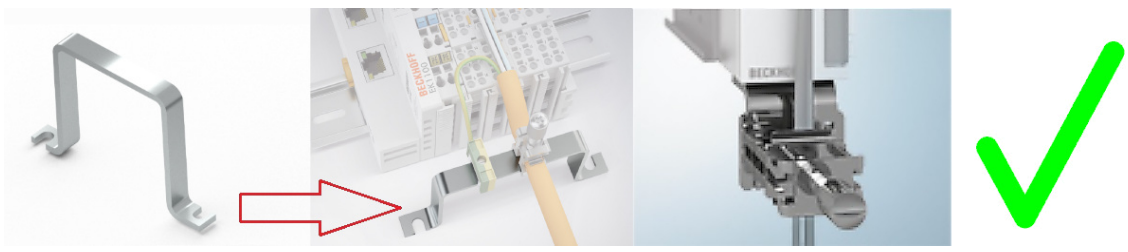
General notes

No special maintenance required, although an annual inspection is recommended for the box module.

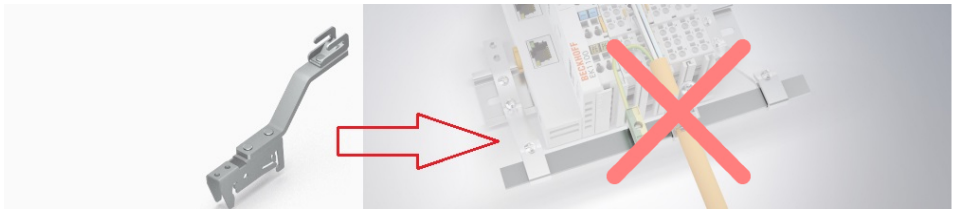
If a factory calibration certificate is available for the device, a one-year recalibration interval is recommended, unless otherwise specified.

Notes regarding the specification data:

- Measurement specifications are usually specified as "% of nominal full scale value" = “% full scale value ($\%_{FSV}$)”, unless otherwise specified.
- With regard to an individual value, "typical" means that on average, this parameter has the specified value. For individual Box modules, the parameter may deviate from the typical value. Current consumption is an example of this.
- In the context of a limit (parameter is typically max./min. X) or with two limits (parameter is typically between X and Y), "typical" means that this parameter predominantly between the limits for the individual Box modules. However, deviations may occur: see confidence level. Noise is an example of this. Usually, no measurements are taken, in order to be able to make statements about standard deviations or result frequencies. A typical value is usually indicated with the abbreviation "typ." after the unit.
- The confidence level is 95%, unless otherwise specified.
- When operating in EMC-disturbed environments, twisted and shielded signal cables which are grounded at one end, at minimum, must be used in order to comply with the specification. The use of Beckhoff shielding accessories ZB8511 or ZS9100-0002 is recommended:



The ZB8520 DIN rail fastening is not recommended with regard to the analog protective effect:



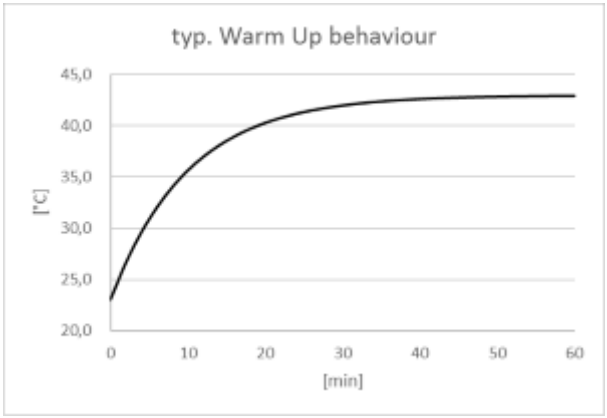
- Unless otherwise specified, measurement errors etc. will be stated in electrical DC operation (no use of AC values). During measurement of an AC value, the frequency slope of the analog input influences the measurement itself.

Note on temperature

The internal/external temperature of the device affects the measurement through the electronics. A measuring setup is generally characterized by a temperature dependence, which is specified in the form of a temperature drift, for example. The specifications apply for a constant ambient temperature. Variable conditions (e.g. heating of the control cabinet, sudden temperature drop due to opening of the control cabinet in cold weather) resulting in a temperature change may alter the measured values through dynamic and heterogeneous temperature distribution. To rectify such effects, the internal temperature of the device can be read online from the CoE and used for calculation. Some devices also electrically indicate that they have thermally stabilized; see diagnostic features.

The specification data apply:

- after a warm-up time under operating voltage and in fieldbus mode of least 60 minutes at constant ambient temperature
  - practical note: after power-on, the device generally heats up exponentially such that the major proportion of the heating has occurred within a short period of approximately 10 to 15 minutes, depending on the device,and the measuring properties shift within the specification limits.
  - for clarification: typical trend of an internal temperature (no significance for a particular device):



  - some devices display that they are internally thermally stabilized and ΔT within the device is very small in the CoE object 0xF900:02 [▶ 85]. This can be evaluated by an application,
- in horizontal installation position, taking the minimum distances into consideration,
- under natural convection (no forced ventilation),
- provided the specifications are adhered to.

Under different conditions, user-specific adjustment is required.

Notes on calculation with the specification data:

The independent specification data can be divided into two groups:

- the data on offset/gain deviation, non-linearity, and repeatability, whose effect on the measurement cannot be influenced by the user. These are summarized by Beckhoff according to the calculation below, at "basic accuracy at 23°C".
- the specification data whose effect on the measurement can be influenced by the user, namely
  - noise: effect can be influenced by sample rate, filtering and
  - the temperature: effect can be influenced by control cabinet air conditioning, shielding, cooling, etc.

The independent individual accuracy data are to be added quadratically according to the formula below in order to determine the total measurement accuracy – if there are no special conditions that contraindicate a uniform distribution and thus the quadratic approach (RSS – root sum squared method).

$$E_{Total} = \sqrt{(E_{Gain} \cdot \frac{MV}{FSV})^2 + (Tc_{Gain} \cdot \Delta T \cdot \frac{MV}{FSV})^2 + E_{Offset}^2 + E_{Lin}^2 + E_{Rep}^2 + (\frac{1}{2} \cdot E_{Noise,PtP})^2 + (Tc_{Offset} \cdot \Delta T)^2 + (E_{Age} \cdot N_{Years})^2}$$

For measurement ranges where the temperature coefficient is only given as $Tc_{Terminal}$:

$$E_{Total} = \sqrt{(E_{Gain} \cdot \frac{MV}{FSV})^2 + E_{Offset}^2 + E_{Lin}^2 + E_{Rep}^2 + (\frac{1}{2} \cdot E_{Noise,PtP})^2 + (Tc_{Terminal} \cdot \Delta T)^2 + (E_{Age} \cdot N_{Years})^2}$$

E_{Offset}	: Offset specification (at 23°C)
E_{Gain}	: Gain/scale specification (at 23°C)
$E_{Noise, PtP}$	: Noise specification as a peak-to-peak value (applies to all temperatures)
MV	: Measured value
FSV	: Full scale value
E_{Lin}	: Non-linearity error over the entire measuring range (applies for all temperatures)
E_{Rep}	: Repeatability (applies to all temperatures)
Tc_{Offset}	: Temperature coefficient offset
Tc_{Gain}	: Temperature coefficient gain
$Tc_{Terminal}$	: Temperature coefficient of the box module
ΔT	: Difference between the ambient temperature ($T_{ambient}$) and the specified basic temperature (23°C unless otherwise specified)
E_{Age}	: Error coefficient for ageing
N_{Years}	: Number of years
E_{Total}	: Theoretically calculated total error

For example, if the following values are obtained at a determined measured value MV of 8.13 V in the 10 V measurement range (FSV = 10 V) (N_{Years} = 0):

- Gain specification: E_{Gain} = 60 ppm
- Offset specification: E_{Offset} = 70 ppm_{FSV}
- Non-linearity: E_{Lin} = 25 ppm_{FSV}
- Repeatability: E_{Rep} = 20 ppm_{FSV}
- Noise (without filtering): $E_{Noise, PtP}$ = 100 $ppm_{peak\text{-}to\text{-}peak}$
- Temperature coefficients:
  - Tc_{Gain} = 8 ppm/K
  - Tc_{Offset} = 5 ppm_{FSV}/K

then the theoretical possible total measurement accuracy at ΔT = 12K for the basic temperature can be calculated as follows:

$$E_{Total} = \sqrt{(60\ ppm \cdot 0.813)^2 + (12K \cdot 8\ ppm/K \cdot 0.813)^2 + (70\ ppm_{FSV})^2 + (25\ ppm_{FSV})^2 + (20\ ppm_{FSV})^2 + (50\ ppm_{FSV})^2 + (12K \cdot 5\ ppm_{FSV}/K)^2}$$

$$= 143.16..\ ppm_{FSV}$$

or = ±0.0143.. $\%_{FSV}$

Remarks: ppm ≙ 10^{-6} % ≙ 10^{-2}

In general, you can calculate this as follows:

- if use at 23°C alone is to be considered:
  Total measurement accuracy = basic accuracy & noise according to above formula
- If use at 23°C is to be considered with slow measurement (=averaging/filtering):
  Total measurement accuracy = basic accuracy
- If general use within a known temperature range and incl. noise is to be considered:
  Total measurement accuracy = basic accuracy & noise & temperature values according to above formula

Beckhoff usually gives the specification data symmetrically in [±%], i.e. ±0.01% or ±100 ppm. Accordingly, therefore, the unsigned total range would be double this given value. A peak-to-peak specification is a total range specification; the symmetrical value is thus half of it. In the quadratic calculation below, the symmetrical value for "one side" is to be inserted without a sign. Noise is usually specified in peak-to-peak form, therefore the equation for the noise value already contains the division factor 2.

Example:

- symmetrical specification: ±0.01% (equivalent to ±100 ppm) e.g. for offset specification
- total range: 0.02% (200 ppm)
- to be used in the equation: 0.01% (100 ppm)

The total measurement accuracy calculated in this way is also to be considered as a symmetrical maximum value and thus to be provided with ± and ≤ for further use.

Example:

- E_{Total} = 100 ppm
- For further use: "≤ ± 100 ppm"

Expressed in words: "The offset of the individual accuracy specifications under the given conditions produced a range of 200 ppm that lies symmetrically around the individual measured value. The measured value specification x thus has an uncertainty of x ±100 ppm; it is thus 95% certain that the true value thus lies in this range".

The noise component can be omitted

The noise component F_{Noise} can be omitted from the above equation (= 0 ppm) if the average value for a set of samples is used instead of a single sample. The averaging can take place in the PLC, or it can be done by a filter in the analog channel. The output value of a moving average of many samples has a noise component that is almost entirely eliminated. The achievable accuracy thus increases when the noise component is decreased.

NOTICE

Error coefficient of ageing

If the specification value for aging from Beckhoff has not (yet) been specified, it must be assumed to be 0 ppm when considering measurement uncertainty, as in the above example, even if in reality it can be assumed that the measurement uncertainty of the device under consideration changes over the operating time, or colloquially stated, the measured value "drifts".

Experience has shown that the basic accuracy of the instrument under consideration, provided it is operated according to specifications, can be taken as the order of magnitude for an annual change (10,000 h). This is an informative statement, does not constitute a specification, and exceptions may occur. In general, the change in ageing will be very application-specific. A general ageing specification from Beckhoff is therefore to be regarded as a guideline rather than a guaranteed upper limit, when published.

If the measurement uncertainty consideration in the application shows that aging over the desired operating time can put the measurement accuracy at risk, Beckhoff recommends a cyclical check (recalibration) of the measurement channel, with regard to sensors, cabling and the Beckhoff measurement modules. In this way, potential long-term changes in the measurement chain can be detected early and, in some cases, the trigger (e.g. overtemperature) can even be eliminated. See also Further documentation for I/O components with analog in and outputs [▶ 252].

Basic accuracy, extended basic accuracy, and averaging

✓ The basic accuracy will be designated separately to simplify usage.

a) The basic accuracy includes the offset/gain deviation, non-linearity and repeatability, but not the temperature coefficient nor the noise and is thereby a subset of the aforementioned complete calculation. It is possible to increase the measurement accuracy beyond the basic accuracy by means of the offset correction.
Note: the ”**extended** basic accuracy” also includes the temperature behavior across the specified operating temperature range, e.g. 0…60°C, via the temperature coefficient.

b) "Averaging" means that the value has been obtained from the arithmetic average of 100,000 (usually) values to eliminate noise. The averaging function integrated into the box module does not necessarily need to be used. If resources are available, averaging can also be executed within the PLC.

Measurement accuracy of the measured value

In several cases, the "Accuracy based on the up-to-date measured value" (percentage of reading), i.e. "Accuracy of value", is sought instead of the "Accuracy related to the full scale value (FSV)" (percentage of range).
This value could easily be calculated from the data given by the specification, as the total accuracy consists of a measured value and full scale value dependent component and an exclusively full scale value dependent component, according to the formula:

$$E_{Total} = \sqrt{\underbrace{\left(E_{Gain} \cdot \frac{MV}{FSV}\right)^2 + \left(Tc_{Gain} \cdot \Delta T \cdot \frac{MV}{FSV}\right)^2}_{\text{Error content, depending on the measurement value}} + \underbrace{E_{Offset}{}^2 + E_{Lin}{}^2 + E_{Rep}{}^2 + \left(\frac{1}{2} \cdot E_{Noise,PtP}\right)^2 + \left(Tc_{Offset} \cdot \Delta T\right)^2 + \left(E_{Age} \cdot N_{Years}\right)^2}_{\text{Error content, exclusive depending on the full scale vlaue}}}$$

4.5 Scope of supply

Make sure that the following components are included in the scope of delivery:

- 1x EtherCAT P Box EPP3504-0023
- 2x protective cap for EtherCAT P socket, M8, red (pre-assembled)
- 4x label, blank

Pre-assembled protective caps do not ensure IP67 protection

Protective caps are pre-assembled at the factory to protect connectors during transport. They may not be tight enough to ensure IP67 protection.

Ensure that the protective caps are correctly seated to ensure IP67 protection.

5 Commissioning

5.1 Notes to short documentation

NOTICE

This short documentation does not contain any further information within this chapter. For the complete documentation please contact the Beckhoff sales department responsible for you.

5.2 CoE overview

5.2.1 General access to online CoE values

Many functionally critical parameters of the EtherCAT Terminals/ Box-Modules are managed *online* in the so-called CoE directory in the device. Under certain circumstances, read or write access is to be made to individual or several entries in the active device during commissioning. That is possible:

- via the OnlineView in TwinCAT 3,
- via a PLC read/write access via ADS and
- via the TwinCAT TF6010 ADS Monitor

OnlineView in TwinCAT 3

The easiest way is the online access in TwinCAT2/3: double-clicking the index/subindex opens an editor window and a value can be changed in decimal/hexadecimal and/or copied and - depending on the type - also written at runtime. Note that the EtherCAT fieldbus is active, the device is addressable and "OnlineData" is visible. Only individual values can be accessed, a so-called CompleteAccess is not possible.

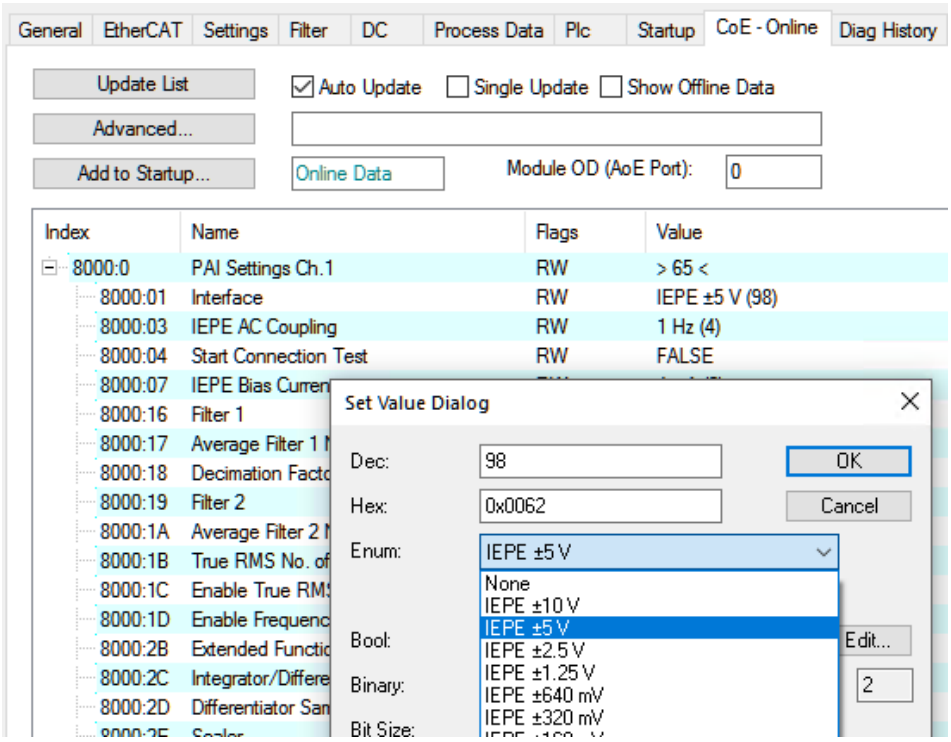


Fig. 19: Example: double click the CoE object *PAI-Settings*, Index *Interface* of channel 1 of an ELM3xxx terminal

PLC access

If values are to be changed or read specifically at application runtime, function blocks (FBs) can be used for CoE access to the TwinCAT TC2_EtherCAT.lib. See also the sample programs in this documentation. Single access and CompleteAccess are possible.

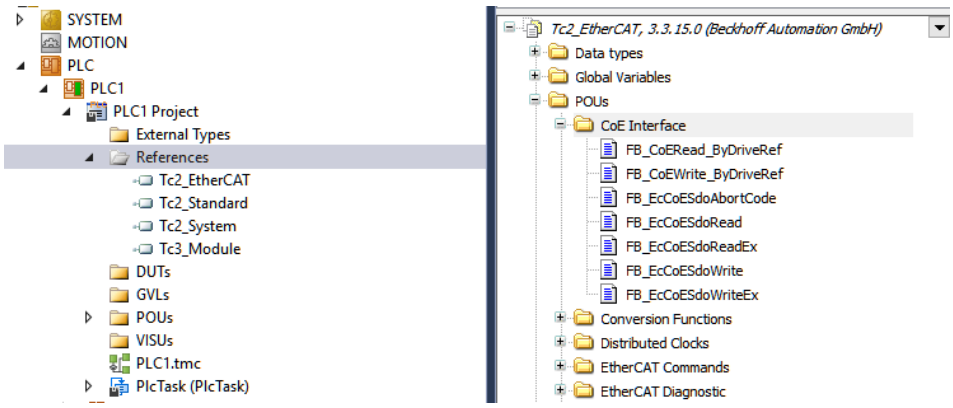


Fig. 20: Function blocks (FBs) for CoE access to the TwinCAT TC2_EtherCAT.lib

TwinCAT TF6010 ADS Monitor

The TF6010 ADS Monitor is a free tool from Beckhoff for monitoring ADS communication. It can be used to read or write CoE values from/to the EtherCAT device (Command Test). Single access and CompleteAccess are possible.

An example: the TwinCAT 3 FilterDesigner TE1310 (or the *FilterControl* on the terminal) generates a set of filter coefficients for digital analog value filtering and sends them to the device.

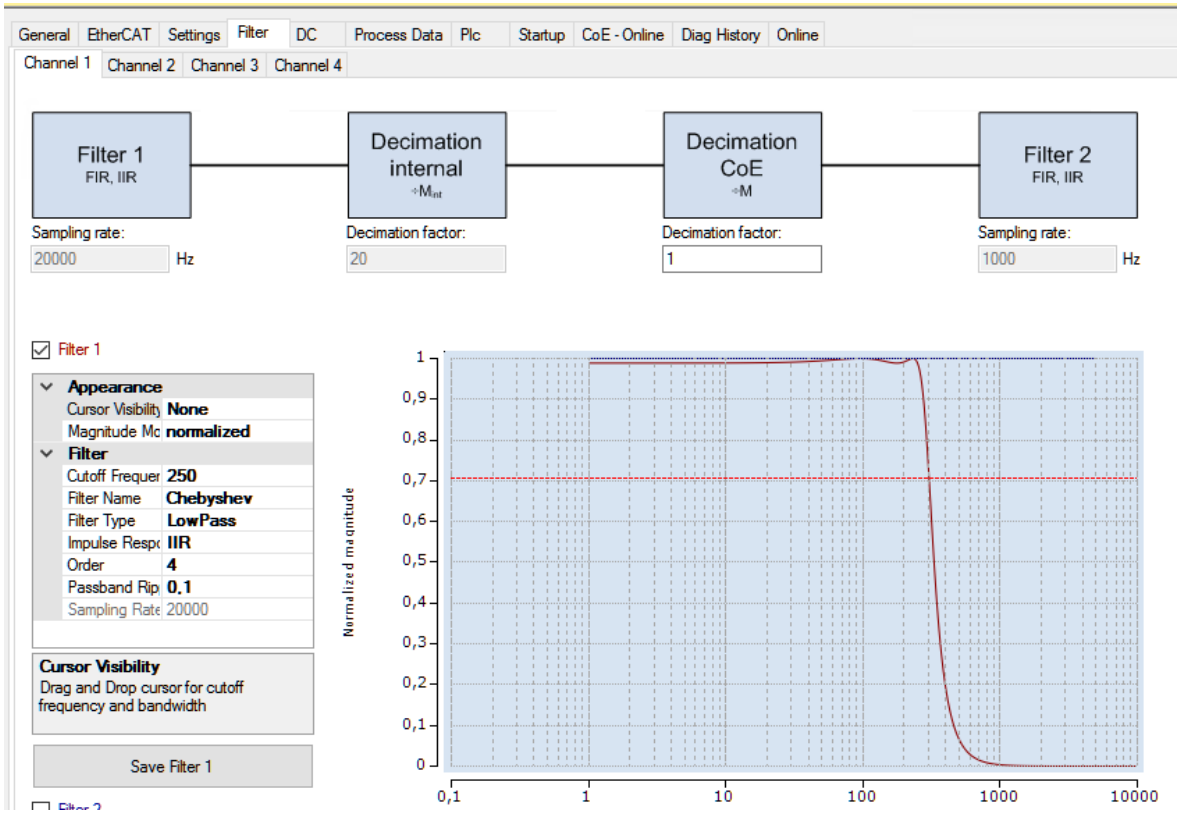


Fig. 21: *TwinCAT FilterControl embedded in the configuration dialog of an* EPP35xx/ ERP35xx box module

The 4-byte coefficients can be read and also copied/edited via the *OnlineView*.

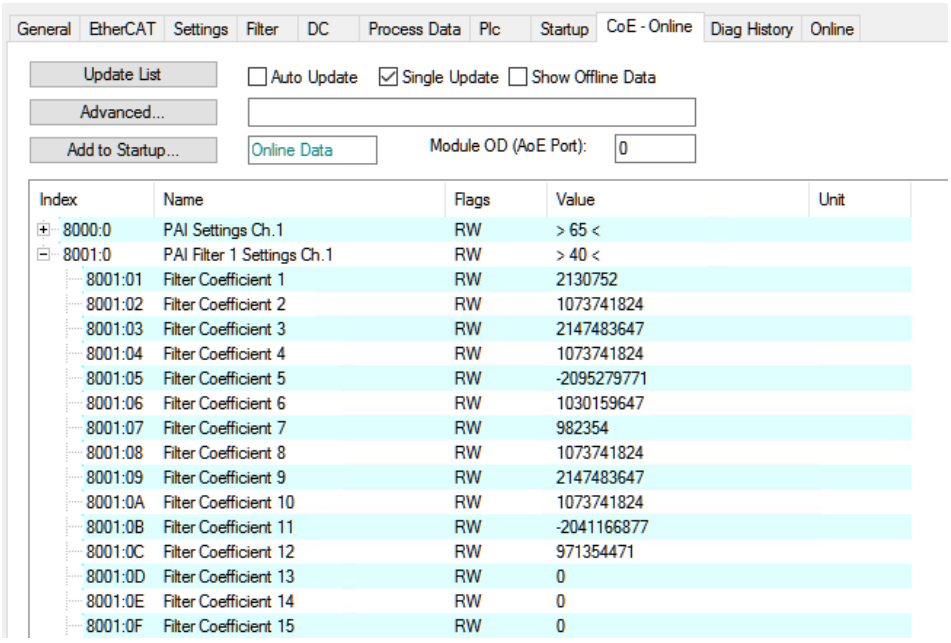


Fig. 22: *Filter coefficients Nos. 1 to 12 of channel 1 in the CoE online of an ELM3602 EtherCAT Terminal*

In case of device replacement

If the coefficients should/have to be reloaded into the new device after a device exchange, they can be stored in a StartUp list:

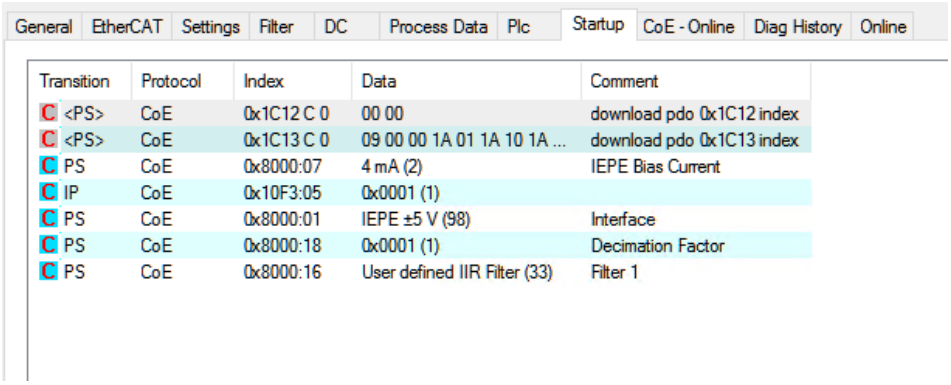


Fig. 23: *StartUp list of an ELM3602 EtherCAT Terminal, already partially modified by direct input*

A total of up to 40 coefficients (i.e. 160 bytes) can be stored there for this device. These are now to be extracted from the box module "in one set", alternatively they could also be copied out manually one after the other as above.

After installing the TF6010 ADS Monitor from the Beckhoff website, it can be started in the development environment menu under [TwinCAT] → [ADS Monitor]:

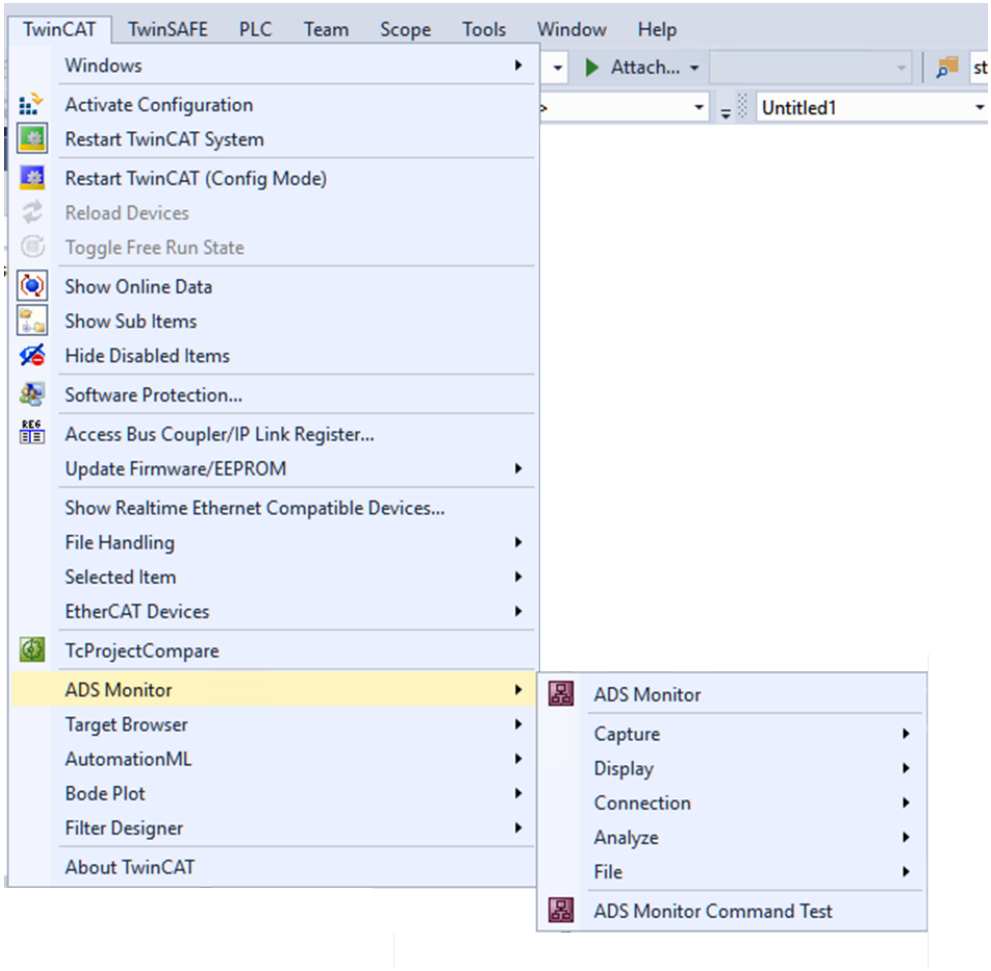


Fig. 24: *Calling the ADS Monitor Command Test*

In order to access the CoE of the EtherCAT box module, "EtherCAT address" must be activated, after which TwinCAT must be activated or restarted.

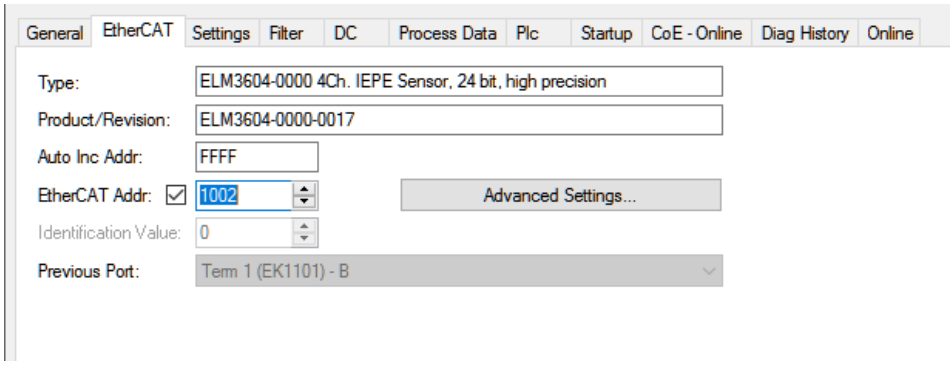
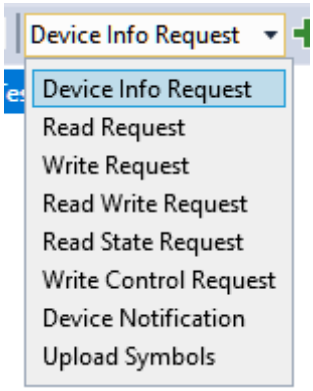


Fig. 25: *Activation of "EtherCAT Addr."*

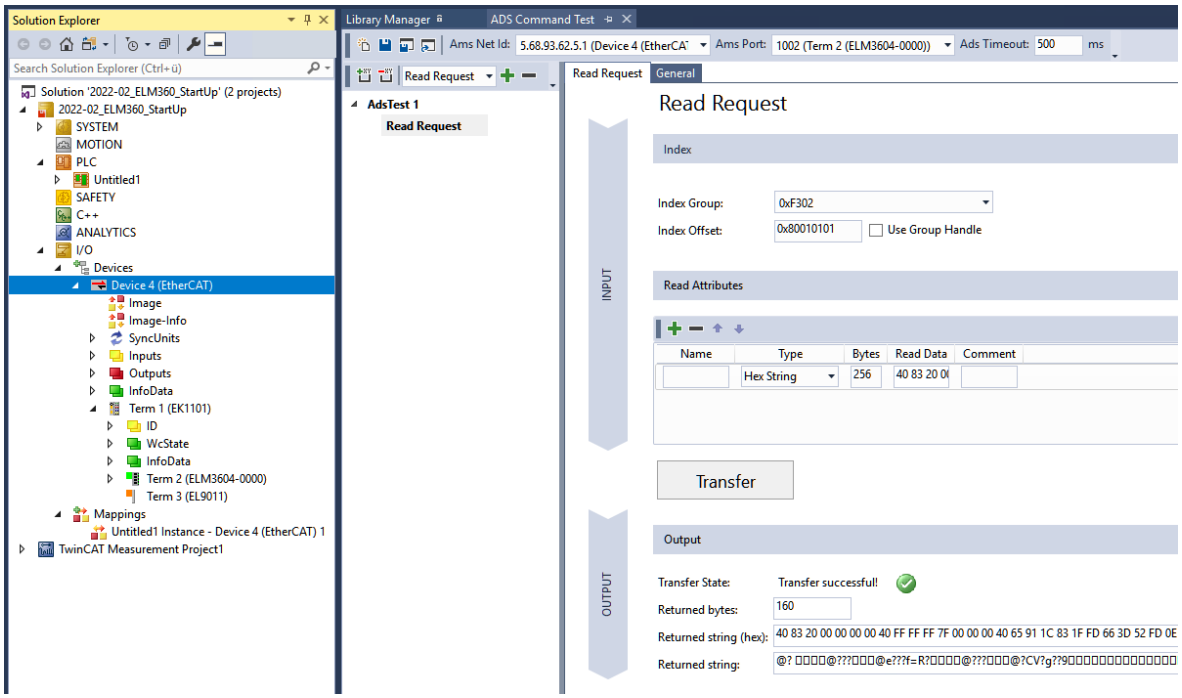
The following is to be entered in the dialog:

- A: Ams Net ID of the EtherCAT master
- B: as port, the EtherCAT address of the 'Slave'

- C: a ReadRequest must be appended for the intended read



- D: as ADS index, the value 0xF302 for the CoE area
- E: composed as ADS offset with eight digits
- first the CoE index, here 0x8001
- then the subindex, here 0x0101 so 257_{dec}
  0...0x00FF would be the regular subindices :01 to :255 in the CoE
  0x0100 is a CompleteAccess including :00, so it returns the size information that is in :00 with (4 bytes)
  0x0101 is a CompleteAccess without the value in :00
- F: the read target must be specified here, e.g. as INT or WORD;
  after successful "transfer" in
- G: are the read data, here the complete 0x8001 content



In this way all 160 bytes (40 coefficients) can be read in one go.
The *byte stream* (BLOP, "binary large object"):
[40 83 20 00 00 00 00 40...] is then to be read inverted:

- Subindex 01: 0x00 20 83 40 = 2130752_{dec}
- Subindex 02: 0x40 00 00 00 = 1073741824_{dec}
- ...

Which corresponds exactly to the values from figure above "Filter coefficients nos. 1 to 12 of channel 1 in CoE-Online of an ELM3602 EtherCAT Terminal". The values can thus be further processed using a spreadsheet program and, if necessary, incorporated into a self-generated Startup.xml.

5.2.2 0x10E2 Manufacturer-specific Identification Code

Index (hex)	Name	Meaning	Data type	Flags	Default
10E2:0	Manufacturer-specific Identification Code	Max. Subindex	UINT8	RO	0x01 (1_{dec})
10E2:01	SubIndex 001	Manufacturer-specific Identification Code that contain the BTN and one or more BIC	STRING(141)	RO	{0}

5.2.3 0x10F3 Diagnosis History

Index (hex)	Name	Meaning	Data type	Flags	Default
10F3:0	Diagnosis History	Max. Subindex	UINT8	RO	0x15 (21_{dec})
10F3:01	Maximum Messages	Maximum Messages	UINT8	RO	0x00 (0_{dec})
10F3:02	Newest Message	Newest Message	UINT8	RO	0x00 (0_{dec})
10F3:03	Newest Acknowledged Message	Subindex of last Acknowledged Message	UINT8	RW	0x00 (0_{dec})
10F3:04	New Messages Available	True: New Messages Available	BOOLEAN	RO	0x00 (FALSE)
10F3:05	Flags	Diagnosis message options (see ETG specification)	UINT16	RW	0x0000 (0_{dec})
10F3:06 .10F3:15	Diagnosis Message 001... Diagnosis Message 016	Diagnosis Message No. 01...16	OCTET-STRING[22]	RO	{0}

5.2.4 0x60n0 PAI Status Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
60n0:0	PAI Status Ch.[n+1]		UINT8	RO	0x0F (15_{dec})
60n0:01	No of Samples	Number of valid samples within the PDO samples	UINT8	RO	0x00 (0_{dec})
60n0:09	Error	TRUE: General error	BOOLEAN	RO	0x00 (FALSE)
60n0:0A	Underrange	TRUE: Measurement event underflow	BOOLEAN	RO	0x00 (FALSE)
60n0:0B	Overrange	TRUE: Measurement event overflow	BOOLEAN	RO	0x00 (FALSE)
60n0:0D	Diag	TRUE: New diagnostic message available	BOOLEAN	RO	0x00 (FALSE)
60n0:0E	TxPDO State	TRUE: data invalid	BOOLEAN	RO	0x00 (FALSE)
60n0:0F	Input cycle counter	Incremented by one when values have changed	BIT2	RO	0x00 (0_{dec})

5.2.5 0x60n1 PAI Samples Ch.[n+1] (24 Bit)

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
60n1:0	PAI Samples Ch.[n+1]		UINT8	RO	0x64 (100_{dec})
60n1:01	Sample	Samples	INT32	RO	0x00000000 (0_{dec})
…	…	…	…	…	…
60n1:64	Sample	Samples	INT32	RO	0x00000000 (0_{dec})

5.2.6 0x60n2 PAI Samples Ch.[n+1] (16 Bit)

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels:

Index (hex)	Name	Meaning	Data type	Flags	Default
60n2:0	PAI Samples Ch.[n+1]		UINT8	RO	0x64 (100_{dec})
60n2:01	Sample	Samples	INT16	RO	0x0000 (0_{dec})
…	…	…	…	…	…
60n2:64	Sample	Samples	INT16	RO	0x0000 (0_{dec})

5.2.7 0x60n3 PAI Samples Ch.[n+1] (REAL32)

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
60n3:0	PAI Samples Ch.[n+1]		UINT8	RO	0x64 (100_{dec})
60n3:01	Sample	Samples	REAL32	RO	0x00000000 (0_{dec})
…	…	…	…	…	…
60n3:64	Sample	Samples	REAL32	RO	0x00000000 (0_{dec})

5.2.8 0x60n5 PAI Timestamp Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
60n5:0	PAI Timestamp Ch.[n+1]		UINT8	RO	0x02 (2_{dec})
60n5:01	Low	Timestamp (low)	UINT32	RO	0x00000000 (0_{dec})
60n5:02	Hi	Timestamp (hi)	UINT32	RO	0x00000000 (0_{dec})

5.2.9 0x60n6 PAI Synchronous Oversampling Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
60n6:0	PAI Synchronous Oversampling Ch.[n+1]		UINT8	RO	0x01 (1_{dec})
60n6:01	Internal Buffer		UINT16	RO	0x0000 (0_{dec})

5.2.10 0x70n0 PAI Control Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
70n0:0	PAI Control Ch.[n+1]		UINT8	RO	0x02 (2_{dec})
70n0:01	Integrator Reset	Restart of the integration with each positive edge	BOOLEAN	RO	0x00 (FALSE)
70n0:02	Peak Hold Reset	Start new peak value detection with each positive edge	BOOLEAN	RO	0x00 (FALSE)

5.2.11 0x80n0 PAI Settings Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:0	PAI Settings Ch.[n+1]		UINT8	RO	0x41 (65_{dec})
80n0:01	Interface	Selection of the measurement configuration: 0 - None 2 - U ±10 V 9 - U ±80 mV 14 - U 0..10 V 42 - PT1000 2Wire 43 - PT1000 3Wire 44 - PT1000 4Wire 65 - Poti 3Wire 66 - Poti 5Wire more.. [▶ 87]	UINT16	RW	0x0000 (0_{dec})
80n0:02	Sensor Supply	Sensor supply: 0 - 0.0 V 2 - 1.0 V 3 - 1.5 V 4 - 2.0 V 5 - 2.5 V 6 - 3.0 V 7 - 3.5 V 8 - 4.0 V 9 - 4.5 V 10 - 5.0 V 65535 - External Supply	UINT16	RW	0x0000 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:04	Start Connection Test	Start connection test with rising edge (see section “Broken wire detection/ optional connection diagnosis”)	BOOLEAN	RW	0x00 (FALSE)
80n0:08	Enable Shunt Calibration	Shunt calibration (Enable/ Disable)	BOOLEAN	RW	0x00 (FALSE)
80n0:13	Wire Resistance Compensation	Wire Resistance Compensation	REAL32	RW	0x00000000 (0.0)
80n0:16	Filter 1	Options for filter 1: 0 – None 1 - FIR Notch 50 Hz 2 - FIR Notch 60 Hz 3 - FIR LP 100 Hz 4 - FIR LP 1000 Hz 5 - FIR HP 150 Hz 16 - IIR Notch 50 Hz 17 - IIR Notch 60 Hz 18 - IIR Butterw. LP 5th Ord. 1 Hz 19 - IIR Butterw. LP 5th Ord. 25 Hz 20 - IIR Butterw. LP 5th Ord. 100 Hz 21 - IIR Butterw. LP 5th Ord. 250 Hz 22 - IIR Butterw. LP 5th Ord. 1000 Hz 32 - User defined FIR Filter 33 - User defined IIR Filter 34 - User defined Average Filter	UINT16	RW	0x0000 (0_{dec})
80n0:17	Average Filter 1 No of Samples	Number of samples for user-defined Average Filter 1	UINT16	RW	0x0001 (1_{dec})
80n0:18	Decimation Factor	Factor of the individual sampling rate (min. 1)	UINT16	RW	0x0001 (1_{dec})
80n0:19	Filter 2	Options for filter 2: 0 - None 1 - IIR 1 2 - IIR 2 3 - IIR 3 4 - IIR 4 5 - IIR 5 6 - IIR 6 7 - IIR 7 8 - IIR 8 16 - User defined FIR Filter 17 - User defined IIR Filter 18 - User defined Average Filter	UINT16	RW	0x0000 (0_{dec})
80n0:1A	Average Filter 2 No of Samples	Number of samples for user-defined Average Filter 2	UINT16	RW	0x0001 (1_{dec})
80n0:1B	True RMS No. of Samples	Number of samples for "True RMS" calculation (min. 1, max. 1000); also see chapter TrueRMS	UINT16	RW	0x0001 (1_{dec})
80n0:1C	Enable True RMS	Activation of "True RMS" calculation	BOOLEAN	RW	0x00 (FALSE)
80n0:1D	Enable Frequency Counter	Enable Frequency Counter	BOOLEAN	RW	0x00 (FALSE)
80n0:1E	Reset Load Cycle Counter	Reset Load Cycle Counter	BOOLEAN	RW	0x00 (FALSE)

Index (hex)	Name	Meaning	Data type	Flags	Default
80n0:2B	Extended Functions	Options for future functions/settings 0 - Disabled 1 - Load Cell Analysis	UINT16	RW	0x0000 (0_{dec})
80n0:2C	Integrator/ Differentiator	Options: 0 - Off 1 - Integrator 1x 2 - Integrator 2x 3 - Differentiator 1x 4 - Differentiator 2x	UINT16	RW	0x0000 (0_{dec})
80n0:2D	Differentiator Samples Delta	Distance of samples for the differentiation; max. value = 1000	UINT16	RW	0x0001 (1_{dec})
80n0:2E	Scaler	Scaling (ENUM): 0 - Extended Range 1 - Linear 2 - Lookup Table 3 - Legacy Range 4 - Lookup Table (additive) 5 - Extended Function 6 - Temperature Celsius 7 - Temperature Kelvin 8 - Temperature Fahrenheit	UINT16	RW	0x0000 (0_{dec})
80n0:2F	Lookup Table Length	Number of taps of the LookUp table	UINT16	RW	0x0064 (100_{dec})
80n0:32	Low Range Error	Lowest limit at which the error bit and the error LED are set	INT32	RW	0xFF800000 (-8388608_{dec})
80n0:33	High Range Error	Highest limit at which the error bit and the error LED are set	INT32	RW	0x007FFFFF (8388607_{dec})
80n0:34	Timestamp Correction	Value for correcting StartNextLatchTime (timestamp of the first sample)	INT32	RW	0xFFFB6C20 (-300000_{dec})
80n0:35	Low Limiter	Smallest PDO output value	REAL32	RW	0xFF7FFFFD (-3.4028231e+38)
80n0:36	High Limiter	Largest PDO output value	REAL32	RW	0x7F7FFFFD (3.4028231e+38)
80n0:37	Bridge Resistance	Bridge resistance	REAL32	RW	0x43AF0000 (350.0)
80n0:38	Wire Resistance neg. Supply	Wire Resistance neg. Supply	REAL32	RW	0x00000000 (0.0)
80n0:39	Wire Resistance pos. Supply	Wire Resistance pos. Supply	REAL32	RW	0x00000000 (0.0)
80n0:3A	Low Load Cycle Limit	Low load cycle limit	REAL32	RW	0xFF7FFFFD (-3.4028231e+38)
80n0:3B	High Load Cycle Limit	High load cycle limit	REAL32	RW	0x7F7FFFFD (3.4028231e+38)
80n0:40	Filter 1 Type Info	Filter 1 type information	STRING	RW	N/A
80n0:41	Filter 2 Type Info	Filter 2 type information	STRING	RW	N/A

5.2.12 0x80n1 PAI Filter 1 Settings Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
80n1:0	PAI Filter 1 Settings Ch.[n+1]		UINT8	RO	0x28 (40_{dec})
80n1:01	Filter Coefficient 1	Coefficients for filter 1	INT32	RW	0x00000000 (0_{dec})
…	…	…	…	…	…
80n1:28	Filter Coefficient 40	Coefficients for filter 1	INT32	RW	0x00000000 (0_{dec})

5.2.13 0x80n3 PAI Filter 2 Settings Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
80n3:0	PAI Filter 2 Settings Ch.[n+1]		UINT8	RO	0x28 (40_{dec})
80n3:01	Filter Coefficient 1	Coefficients for filter 2	INT32	RW	0x00000000 (0_{dec})
…	…	…	…	…	…
80n3:28	Filter Coefficient 40	Coefficients for filter 2	INT32	RW	0x00000000 (0_{dec})

5.2.14 0x80n6 PAI Scaler Settings Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
80n6:0	PAI Scaler Settings Ch.[n+1]	Scaling values offset/gain or LookUp table with 50 x/y value pairs	UINT8	RO	0x64 (100_{dec})
80n6:01	Scaler Offset/ Scaler Value 1	Scaling offset or LookUp x value 1	REAL32	RW	0x00000000 (0_{dec})
80n6:02	Scaler-Gain/ Scaler Value 2	Scaling gain or LookUp y value 1	REAL32	RW	0x00000000 (0_{dec})
80n6:03	Scaler Value 3	LookUp x value 2	REAL32	RW	0x00000000 (0_{dec})
80n6:04	Scaler Value 4	LookUp y value 2	REAL32	RW	0x00000000 (0_{dec})
..	..	..	..	..	..
80n6:63	Scaler Value 99	LookUp x value 50	REAL32	RW	0x00000000 (0_{dec})
80n6:64	Scaler Value 100	LookUp y value 50	REAL32	RW	0x00000000 (0_{dec})

5.2.15 0x80nA PAI Extended Settings Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels
(Special settings for the „Extended Functions“)

Index (hex)	Name	Meaning	Data type	Flags	Default
80nA:0	PAI Extended Settings Ch.[n+1]	Special settings for the „Extended Functions“	UINT8	RO	0x05 (5_{dec})
80nA:01	Sensitivity (Compression)	Sensitivity (mech. compression)	REAL32	RW	0x40000000 (2.0)
80nA:02	Sensitivity (Tension)	Sensitivity (mech. tension)	REAL32	RW	0xC0000000 (-2.0)
80nA:03	Zero Balance	Zero balance	REAL32	RW	0x00000000 (0.0)
80nA:04	Maximum Capacity	Maximum capacity	REAL32	RW	0x40A00000 (5.0)
80nA:05	Gravity of Earth	Gravity of earth	REAL32	RW	0x411CE80A (9.8066502)

5.2.16 0x80nE PAI User Calibration Data Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
80nE:0	PAI User Calibration Data Ch.1		UINT8	RO	0x0C (12_{dec})
80nE:01	Calibration Date	Date of calibration	OCTET-STRING[4]	RW	-
80nE:02	Signature	Signature of the calibration values	OCTET-STRING[256]	RW	-
80nE:03	S0	Offset	REAL32	RW	0x00000000 (0_{dec})
80nE:04	S1	Coefficient for first-order samples (S1 * sample)	REAL32	RW	0x3E800000 (1.0)
80nE:05	S2	Coefficient for second-order samples (S2 * sample²)	REAL32	RW	0x00000000 (0_{dec})
80nE:06	S3	Coefficient for third-order samples (S3 * sample³)	REAL32	RW	0x00000000 (0_{dec})
80nE:07	T1	Temperature coefficient for first-order temperature value (T1 * temp)	REAL32	RW	0x00000000 (0_{dec})
80nE:08	T1S1	Combined coefficient for first-order gain and temperature values (T1S1 * temp * sample)	REAL32	RW	0x00000000 (0_{dec})
80nE:09	T2	Temperature coefficient for second-order temperature value (T2 * temp²)	REAL32	RW	0x00000000 (0_{dec})
80nE:0A	T2S1	Combined coefficient for second-order gain and temperature values (T2S1 * temp² * sample)	REAL32	RW	0x00000000 (0_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
80nE:0B	T3	Temperature coefficient for third-order temperature value (T3 * temp³)	REAL32	RW	0x00000000 (0_{dec})
80nE:0C	T3S1	Combined coefficient for third-order gain and temperature values (T3S1 * temp³ * sample)	REAL32	RW	0x00000000 (0_{dec})

5.2.17 0x80nF PAI Vendor Calibration Data Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
80nF:0	PAI Vendor Calibration Data Ch.1		UINT8	RO	0x0C (12_{dec})
80nF:01	Calibration Date	Date of calibration	OCTET-STRING[4]	RW	-
80nF:02	Signature	Signature of the calibration values	OCTET-STRING[256]	RW	-
80nF:03	S0	Offset	REAL32	RW	0x00000000 (0_{dec})
80nF:04	S1	Coefficient for first-order samples (S1 * sample)	REAL32	RW	-
80nF:05	S2	Coefficient for second-order samples (S2 * sample²)	REAL32	RW	0x00000000 (0_{dec})
80nF:06	S3	Coefficient for third-order samples (S3 * sample³)	REAL32	RW	0x00000000 (0_{dec})
80nF:07	T1	Temperature coefficient for first-order temperature value (T1 * temp)	REAL32	RW	0x00000000 (0_{dec})
80nF:08	T1S1	Combined coefficient for first-order gain and temperature values (T1S1 * temp * sample)	REAL32	RW	0x00000000 (0_{dec})
80nF:09	T2	Temperature coefficient for second-order temperature value (T2 * temp²)	REAL32	RW	0x00000000 (0_{dec})
80nF:0A	T2S1	Combined coefficient for second-order gain and temperature values (T2S1 * temp² * sample)	REAL32	RW	0x00000000 (0_{dec})
80nF:0B	T3	Temperature coefficient for third-order temperature value (T3 * temp³)	REAL32	RW	0x00000000 (0_{dec})
80nF:0C	T3S1	Combined coefficient for third-order gain and temperature values (T3S1 * temp³ * sample)	REAL32	RW	0x00000000 (0_{dec})

5.2.18 0x90n0 PAI Internal Data Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
90n0:0	PAI Internal Data Ch.[n+1]		UINT8	RO	0x22 (34_{dec})
90n0:02	ADC Raw Value	ADC Raw Value	INT32	RO	0x00000000 (0_{dec})
90n0:03	Calibration Value	Value after calibration	INT32	RO	0x00000000 (0_{dec})
90n0:04	Zero Offset Value	Zero offset value	INT32	RO	0x00000000 (0_{dec})
90n0:05	Resistor Value	Resistor Value	INT32	RO	0x00000000 (0_{dec})
90n0:07	Actual Negative Peak Hold	Current absolute minimum value	INT32	RO	0x00000000 (0_{dec})
90n0:08	Actual Positive Peak Hold	Current absolute maximum value	INT32	RO	0x00000000 (0_{dec})
90n0:09	Previous Negative Peak Hold	Absolute minimum value up to last rising edge of "Peak Hold Reset"	INT32	RO	0x00000000 (0_{dec})
90n0:0A	Previous Positive Peak Hold	Absolute maximum value up to last rising edge of "Peak Hold Reset"	INT32	RO	0x00000000 (0_{dec})
90n0:0B	Filter 1 Value	Value after filter 1	INT32	RO	0x00000000 (0_{dec})
90n0:0C	Filter 2 Value	Value after filter 2	INT32	RO	0x00000000 (0_{dec})
90n0:0D	True RMS Value	Value after "True RMS" calculation	INT32	RO	0x00000000 (0_{dec})
90n0:0E	Extended Functions Value	Value after advanced (optional) function	INT32	RO	0x00000000 (0_{dec})
90n0:0F	Integrator/ Differentiator Value	Value after integration or differentiation	INT32	RO	0x00000000 (0_{dec})
90n0:10	Scaler Value	Value after scaling	INT32	RO	0x00000000 (0_{dec})
90n0:11	Limiter Value	Value after limitation	INT32	RO	0x00000000 (0_{dec})
90n0:21	Signal Frequency	Frequency of the input signal	UINT32	RO	0x00000000 (0_{dec})
90n0:22	Signal Duty Cycle	Duty Cycle of the input signal	UINT8	RO	0x00 (0_{dec})

5.2.19 0x90n2 PAI Info Data Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels

Index (hex)	Name	Meaning	Data type	Flags	Default
90n2:0	PAI Info Data Ch.[n+1]		UINT8	RO	0x12 (18_{dec})

Index (hex)	Name	Meaning	Data type	Flags	Default
90n2:01	Effective Sample Rate	Effective Sample Rate	UINT32	RO	0x00000000 (0_{dec})
90n2:02	Channel Temperature	Temperature of the channel	REAL32	RO	0x00000000 (0_{dec})
90n2:03	Min. Channel Temperature	Minimum temperature of the channel	REAL32	RO	0x00000000 (0_{dec})
90n2:04	Max. Channel Temperature	Maximum temperature of the channel	REAL32	RO	0x00000000 (0_{dec})
90n2:05	Overload Time	Absolute time during overload "Overload" means that the channel is electrically overloaded. This is a non-recommendable condition that may eventually lead to atypical aging or even damage. This condition should be avoided. Its accumulated duration is displayed here informatively.	UINT32	RO	0x00000000 (0_{dec})
90n2:06	Saturation Time	Absolute time during saturation "Saturation" means that the measuring range of the ADC of the channel is fully utilized, the ADC thus outputs its maximum value and the measured value can no longer be used. "Saturation" is therefore a pre-deregistration, with further signal increase it comes to "overload". The saturation state is not fundamentally harmful, but it indicates an insufficient dimensioning of the measurement channel. Its accumulated response time is displayed here informatively.	UINT32	RO	0x00000000 (0_{dec})
90n2:07	Overtemperature Time (Channel)	Time of exceeded temperature of the channel	UINT32	RO	0x00000000 (0_{dec})
90n2:10	Load Cycle Counter	Load Cycle Counter	UINT32	RO	0x00000000 (0_{dec})
90n2:11	Vendor Calibration Counter	Counter of the vendor calibration (related to the selected interface) The counter counts +1 when data has changed and the memory code word is written. Depending on the adjustment method, the counter may therefore count several times.	UINT16	RO	0x0000 (0_{dec})
90n2:12	User Calibration Counter	Counter of the user calibration (related to the selected interface) The counter counts +1 when data has changed and the memory code word is written. Depending on the adjustment method, the counter may therefore count several times.	UINT16	RO	0x0000 (0_{dec})

5.2.20 0x90nF PAI Calibration Dates Ch.[n+1]

0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels:

Index (hex)	Name	Meaning	Data type	Flags	Default
90nF:0	PAI Calibration Dates Ch. [n+1]		UINT8	RO	0xC4 (196_{dec})
90nF:01	Vendor U ±10 V		OCTET-STRING[4]	RO	{0}
90nF:02	Vendor U ±80 mV		OCTET-STRING[4]	RO	{0}
90nF:03	Vendor U 0..10 V		OCTET-STRING[4]	RO	{0}
90nF:04	Vendor PT1000 2Wire		OCTET-STRING[4]	RO	{0}
90nF:05	Vendor PT1000 3Wire		OCTET-STRING[4]	RO	{0}
90nF:06	Vendor PT1000 4Wire		OCTET-STRING[4]	RO	{0}
90nF:07	Vendor Poti 3Wire		OCTET-STRING[4]	RO	{0}
90nF:08	Vendor Poti 5Wire		OCTET-STRING[4]	RO	{0}
90nF:09	Vendor SG Full-Bridge 4Wire 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:0A	Vendor SG Full-Bridge 4Wire 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:0B	Vendor SG Full-Bridge 4Wire 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:0C	Vendor SG Full-Bridge 4Wire 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:0D	Vendor SG Full-Bridge 4Wire 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:0E	Vendor SG Full-Bridge 4Wire 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:0F	Vendor SG Full-Bridge 6Wire 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:10	Vendor SG Full-Bridge 6Wire 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:11	Vendor SG Full-Bridge 6Wire 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:12	Vendor SG Full-Bridge 6Wire 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:13	Vendor SG Full-Bridge 6Wire 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:14	Vendor SG Full-Bridge 6Wire 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:15	Vendor SG Half-Bridge 3Wire 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:16	Vendor SG Half-Bridge 3Wire 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:17	Vendor SG Half-Bridge 3Wire 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:18	Vendor SG Half-Bridge 3Wire 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:19	Vendor SG Half-Bridge 3Wire 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:1A	Vendor SG Half-Bridge 3Wire 16 mV/V		OCTET-STRING[4]	RO	{0}
90nF:1B	Vendor SG Half-Bridge 5Wire 2 mV/V		OCTET-STRING[4]	RO	{0}

Index (hex)	Name	Meaning	Data type	Flags	Default
90nF:1C	Vendor SG Half-Bridge 5Wire 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:1D	Vendor SG Half-Bridge 5Wire 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:1E	Vendor SG Half-Bridge 5Wire 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:1F	Vendor SG Half-Bridge 5Wire 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:20	Vendor SG Half-Bridge 5Wire 16 mV/V		OCTET-STRING[4]	RO	{0}
90nF:21	Vendor SG Quarter-Bridge 2Wire 120R 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:22	Vendor SG Quarter-Bridge 2Wire 120R 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:23	Vendor SG Quarter-Bridge 2Wire 120R 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:24	Vendor SG Quarter-Bridge 2Wire 120R 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:25	Vendor SG Quarter-Bridge 2Wire 120R 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:26	Vendor SG Quarter-Bridge 2Wire 120R 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:27	Vendor SG Quarter-Bridge 3Wire 120R 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:28	Vendor SG Quarter-Bridge 3Wire 120R 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:29	Vendor SG Quarter-Bridge 3Wire 120R 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:2A	Vendor SG Quarter-Bridge 3Wire 120R 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:2B	Vendor SG Quarter-Bridge 3Wire 120R 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:2C	Vendor SG Quarter-Bridge 3Wire 120R 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:2D	Vendor SG Quarter-Bridge 2Wire 350R 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:2E	Vendor SG Quarter-Bridge 2Wire 350R 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:2F	Vendor SG Quarter-Bridge 2Wire 350R 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:30	Vendor SG Quarter-Bridge 2Wire 350R 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:31	Vendor SG Quarter-Bridge 2Wire 350R 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:32	Vendor SG Quarter-Bridge 2Wire 350R 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:33	Vendor SG Quarter-Bridge 3Wire 350R 2 mV/V		OCTET-STRING[4]	RO	{0}

Index (hex)	Name	Meaning	Data type	Flags	Default
90nF:34	Vendor SG Quarter-Bridge 3Wire 350R 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:35	Vendor SG Quarter-Bridge 3Wire 350R 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:36	Vendor SG Quarter-Bridge 3Wire 350R 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:37	Vendor SG Quarter-Bridge 3Wire 350R 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:38	Vendor SG Quarter-Bridge 3Wire 350R 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:39	Vendor SG Quarter-Bridge 2Wire 1k 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:3A	Vendor SG Quarter-Bridge 2Wire 1k 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:3B	Vendor SG Quarter-Bridge 2Wire 1k 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:3C	Vendor SG Quarter-Bridge 2Wire 1k 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:3D	Vendor SG Quarter-Bridge 2Wire 1k 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:3E	Vendor SG Quarter-Bridge 2Wire 1k 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:3F	Vendor SG Quarter-Bridge 3Wire 1k 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:40	Vendor SG Quarter-Bridge 3Wire 1k 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:41	Vendor SG Quarter-Bridge 3Wire 1k 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:42	Vendor SG Quarter-Bridge 3Wire 1k 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:43	Vendor SG Quarter-Bridge 3Wire 1k 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:44	Vendor SG Quarter-Bridge 3Wire 1k 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:81	User U ±10 V		OCTET-STRING[4]	RO	{0}
90nF:82	User U ±80 mV		OCTET-STRING[4]	RO	{0}
90nF:83	User U 0..10 V		OCTET-STRING[4]	RO	{0}
90nF:84	User PT1000 2Wire		OCTET-STRING[4]	RO	{0}
90nF:85	User PT1000 3Wire		OCTET-STRING[4]	RO	{0}
90nF:86	User PT1000 4Wire		OCTET-STRING[4]	RO	{0}
90nF:87	User Poti 3Wire		OCTET-STRING[4]	RO	{0}
90nF:88	User Poti 5Wire		OCTET-STRING[4]	RO	{0}
90nF:89	User SG Full-Bridge 4Wire 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:8A	User SG Full-Bridge 4Wire 2 mV/V compensated		OCTET-STRING[4]	RO	{0}

Index (hex)	Name	Meaning	Data type	Flags	Default
90nF:8B	User SG Full-Bridge 4Wire 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:8C	User SG Full-Bridge 4Wire 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:8D	User SG Full-Bridge 4Wire 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:8E	User SG Full-Bridge 4Wire 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:8F	User SG Full-Bridge 6Wire 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:90	User SG Full-Bridge 6Wire 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:91	User SG Full-Bridge 6Wire 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:92	User SG Full-Bridge 6Wire 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:93	User SG Full-Bridge 6Wire 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:94	User SG Full-Bridge 6Wire 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:95	User SG Half-Bridge 3Wire 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:96	User SG Half-Bridge 3Wire 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:97	User SG Half-Bridge 3Wire 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:98	User SG Half-Bridge 3Wire 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:99	User SG Half-Bridge 3Wire 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:9A	User SG Half-Bridge 3Wire 16 mV/V		OCTET-STRING[4]	RO	{0}
90nF:9B	User SG Half-Bridge 5Wire 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:9C	User SG Half-Bridge 5Wire 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:9D	User SG Half-Bridge 5Wire 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:9E	User SG Half-Bridge 5Wire 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:9F	User SG Half-Bridge 5Wire 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:A0	User SG Half-Bridge 5Wire 16 mV/V		OCTET-STRING[4]	RO	{0}
90nF:A1	User SG Quarter-Bridge 2Wire 120R 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:A2	User SG Quarter-Bridge 2Wire 120R 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:A3	User SG Quarter-Bridge 2Wire 120R 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:A4	User SG Quarter-Bridge 2Wire 120R 4 mV/V compensated		OCTET-STRING[4]	RO	{0}

Index (hex)	Name	Meaning	Data type	Flags	Default
90nF:A5	User SG Quarter-Bridge 2Wire 120R 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:A6	User SG Quarter-Bridge 2Wire 120R 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:A7	User SG Quarter-Bridge 3Wire 120R 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:A8	User SG Quarter-Bridge 3Wire 120R 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:A9	User SG Quarter-Bridge 3Wire 120R 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:AA	User SG Quarter-Bridge 3Wire 120R 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:AB	User SG Quarter-Bridge 3Wire 120R 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:AC	User SG Quarter-Bridge 3Wire 120R 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:AD	User SG Quarter-Bridge 2Wire 350R 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:AE	User SG Quarter-Bridge 2Wire 350R 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:AF	User SG Quarter-Bridge 2Wire 350R 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:B0	User SG Quarter-Bridge 2Wire 350R 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:B1	User SG Quarter-Bridge 2Wire 350R 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:B2	User SG Quarter-Bridge 2Wire 350R 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:B3	User SG Quarter-Bridge 3Wire 350R 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:B4	User SG Quarter-Bridge 3Wire 350R 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:B5	User SG Quarter-Bridge 3Wire 350R 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:B6	User SG Quarter-Bridge 3Wire 350R 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:B7	User SG Quarter-Bridge 3Wire 350R 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:B8	User SG Quarter-Bridge 3Wire 350R 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:B9	User SG Quarter-Bridge 2Wire 1k 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:BA	User SG Quarter-Bridge 2Wire 1k 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:BB	User SG Quarter-Bridge 2Wire 1k 4 mV/V		OCTET-STRING[4]	RO	{0}

Index (hex)	Name	Meaning	Data type	Flags	Default
90nF:BC	User SG Quarter-Bridge 2Wire 1k 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:BD	User SG Quarter-Bridge 2Wire 1k 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:BE	User SG Quarter-Bridge 2Wire 1k 32 mV/V		OCTET-STRING[4]	RO	{0}
90nF:BF	User SG Quarter-Bridge 3Wire 1k 2 mV/V		OCTET-STRING[4]	RO	{0}
90nF:C0	User SG Quarter-Bridge 3Wire 1k 2 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:C1	User SG Quarter-Bridge 3Wire 1k 4 mV/V		OCTET-STRING[4]	RO	{0}
90nF:C2	User SG Quarter-Bridge 3Wire 1k 4 mV/V compensated		OCTET-STRING[4]	RO	{0}
90nF:C3	User SG Quarter-Bridge 3Wire 1k 8 mV/V		OCTET-STRING[4]	RO	{0}
90nF:C4	User SG Quarter-Bridge 3Wire 1k 32 mV/V		OCTET-STRING[4]	RO	{0}

5.2.21 0xF000 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	n

n = number of existing channels by the box module

5.2.22 0xF008 Code word

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

5.2.23 0xF009 Password protection

Index (hex)	Name	Meaning	Data type	Flags	Default
F009:0	Password protection		UINT32	RW	0x00000000 (0_{dec})

5.2.24 0xF010 Module list

Index (hex)	Name	Meaning	Data type	Flags	Default
F010:0	Module list		UINT8	RW	n
F010:01	Subindex 001		UINT32	RW	0x0000015E (350_{dec})
...	...	...	...	...	...
F010:n	Subindex n		UINT32	RW	0x0000015E (350_{dec})

n = number of existing channels by the box module

5.2.25 0xF083 BTN

Index (hex)	Name	Meaning	Data type	Flags	Default
F083:0	BTN	Beckhoff Traceability Number	STRING	RO	00000000

5.2.26 0xF900 PAI Info Data

Index (hex)	Name	Meaning	Data type	Flags	Default
F900:0	PAI Info Data		UINT8	RO	0x13 (19_{dec})
F900:01	CPU Usage	CPU load in [%] [1)]	UINT16	RO	0x0000 (0_{dec})
F900:02	Device State	Device State Permitted values: 0 – Ok 1 – Warm Up	UINT16	RO	0x0000 (0_{dec})
F900:03	Operating Time	Operating time in [min]	UINT32	RO	0x00000000 (0_{dec})
F900:04	Overtemperature Time (Device)	Time of overtemperature of the device	UINT32	RO	0x00000000 (0_{dec})
F900:11	Device Temperature	Measured temperature in the terminal	REAL32	RO	0x00000000 (0_{dec})
F900:12	Min. Device Temperature	Lowest measured temperature in the terminal	REAL32	RO	0x00000000 (0_{dec})
F900:13	Max. Device Temperature	Highest measured temperature in the terminal	REAL32	RO	0x00000000 (0_{dec})

[1)] This value depends on additional enabled features (Filters, True RMS, ...); the more functions of the box module are in use, the greater is the value. Notice amongst others the „Input cycle counter“ (PAI Status [▶ 69]). The CPU load is an informative value with particularly regard to the "Device-specific Diag messages".

5.2.27 0xF912 Filter info

Index (hex)	Name	Meaning	Data type	Flags	Default
F912:0	Filter info		UINT8	RO	m
F912:01	Info header	Basic information for the filter designer	OCTET-STRING[8]	RO	{0}
F912:02	Filter 1	Informations for the filter designer	OCTET-STRING[30]	RO	{0}
...	...	...	...	...	...
F912:m	Filter n	Informations for the filter designer	OCTET-STRING[30]	RO	{0}

m = (2 · No. of channels) + 1

5.2.28 0xFB00 PAI Command

Index (hex)	Name	Meaning	Data type	Flags	Default
FB00:0	PAI Command		UINT8	RO	0x03 (3_{dec})
FB00:01	Request	Command request The respective functional chapters explain which value is to be entered here.	OCTET-STRING[2]	RW	{0}
FB00:02	Status	Command status This indicates that the command is still running or has been executed. Functional dependent, see respective sections. Otherwise: 0: Command not existing 1: executed without errors 2,3: executed not successful 100..200: indicates the execution progress (100 = 0% etc.) 255: function is busy, if [100..200] won't be used as progress display	UINT8	RO	0x00 (0_{dec})
FB00:03	Response	Command response If the transferred command returns a response, it will be displayed here. Functional dependent, see resprective sections.	OCTET-STRING[6]	RO	{0}

5.2.29 0x80n0:01 PAI Settings.Interface

0x80n0:01 PAI Settings.Interface (0 ≤ n ≤ m, n+1 = Channel number, m+1 = max. No. of channels) - continued

Index (hex)	Meaning	Data type	Flags	Default
80n0:01	Selection of the measurement configuration (continued): 0x80n0 PAI Settings [▶ 71] ... 259 - SG Full-Bridge 4Wire 2 mV/V 260 - SG Full-Bridge 4Wire 2 mV/V compensated 261 - SG Full-Bridge 4Wire 4 mV/V 262 - SG Full-Bridge 4Wire 4 mV/V compensated 263 - SG Full-Bridge 4Wire 8 mV/V 268 - SG Full-Bridge 4Wire 32 mV/V 291 - SG Full-Bridge 6Wire 2 mV/V 292 - SG Full-Bridge 6Wire 2 mV/V compensated 293 - SG Full-Bridge 6Wire 4 mV/V 294 - SG Full-Bridge 6Wire 4 mV/V compensated 295 - SG Full-Bridge 6Wire 8 mV/V 300 - SG Full-Bridge 6Wire 32 mV/V 323 - SG Half-Bridge 3Wire 2 mV/V 324 - SG Half-Bridge 3Wire 2 mV/V compensated 325 - SG Half-Bridge 3Wire 4 mV/V 326 - SG Half-Bridge 3Wire 4 mV/V compensated 327 - SG Half-Bridge 3Wire 8 mV/V 329 - SG Half-Bridge 3Wire 16 mV/V 355 - SG Half-Bridge 5Wire 2 mV/V 356 - SG Half-Bridge 5Wire 2 mV/V compensated 357 - SG Half-Bridge 5Wire 4 mV/V 358 - SG Half-Bridge 5Wire 4 mV/V compensated 359 - SG Half-Bridge 5Wire 8 mV/V 361 - SG Half-Bridge 5Wire 16 mV/V 388 - SG Quarter-Bridge 2Wire 120R 2 mV/V compensated 390 - SG Quarter-Bridge 2Wire 120R 4 mV/V compensated 391 - SG Quarter-Bridge 2Wire 120R 8 mV/V 396 - SG Quarter-Bridge 2Wire 120R 32 mV/V 420 - SG Quarter-Bridge 3Wire 120R 2 mV/V compensated 422 - SG Quarter-Bridge 3Wire 120R 4 mV/V compensated 423 - SG Quarter-Bridge 3Wire 120R 8 mV/V 428 - SG Quarter-Bridge 3Wire 120R 32 mV/V 452 - SG Quarter-Bridge 2Wire 350R 2 mV/V compensated 454 - SG Quarter-Bridge 2Wire 350R 4 mV/V compensated 455 - SG Quarter-Bridge 2Wire 350R 8 mV/V 460 - SG Quarter-Bridge 2Wire 350R 32 mV/V 484 - SG Quarter-Bridge 3Wire 350R 2 mV/V compensated 486 - SG Quarter-Bridge 3Wire 350R 4 mV/V compensated 487 - SG Quarter-Bridge 3Wire 350R 8 mV/V 492 - SG Quarter-Bridge 3Wire 350R 32 mV/V 516 - SG Quarter-Bridge 2Wire 1k 2 mV/V compensated 518 - SG Quarter-Bridge 2Wire 1k 4 mV/V compensated 519 - SG Quarter-Bridge 2Wire 1k 8 mV/V 524 - SG Quarter-Bridge 2Wire 1k 32 mV/V 548 - SG Quarter-Bridge 3Wire 1k 2 mV/V compensated 550 - SG Quarter-Bridge 3Wire 1k 4 mV/V compensated 551 - SG Quarter-Bridge 3Wire 1k 8 mV/V 556 - SG Quarter-Bridge 3Wire 1k 32 mV/V	UINT16	RW	0x0000 (0_{dec})

5.3 Sample programs

Using the sample programs

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

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

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

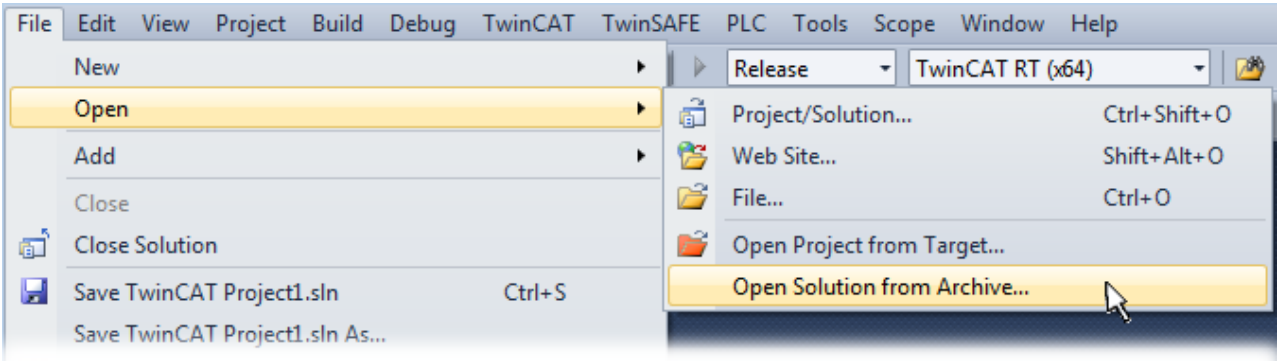


Fig. 26: Opening the *. tnzip archive

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

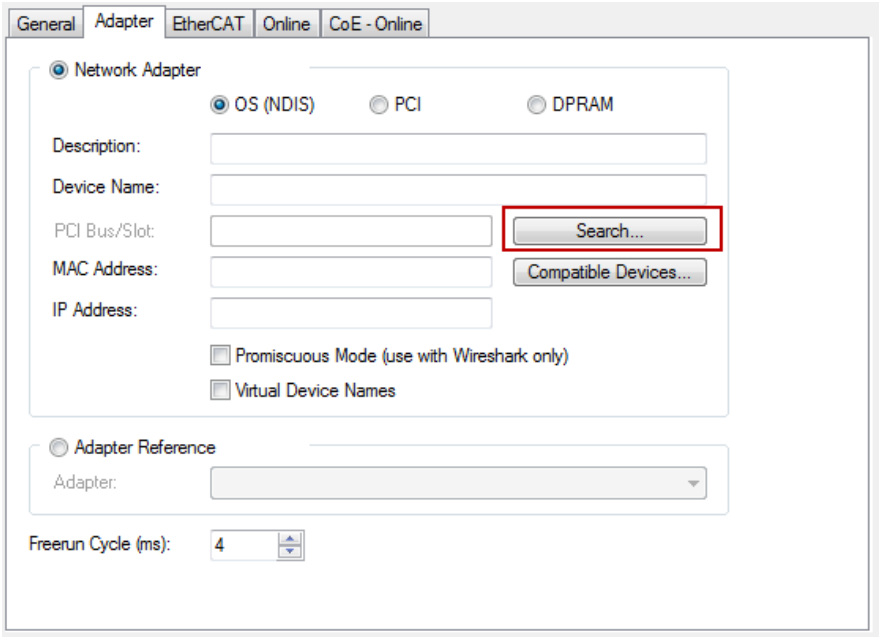
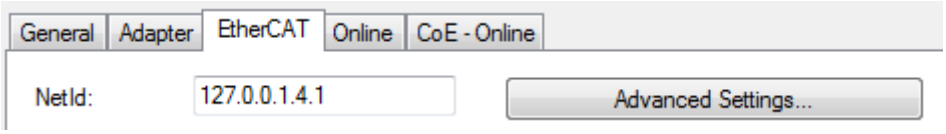


Fig. 27: Search of the existing HW configuration for the EtherCAT configuration of the example

- Checking NetId: the “EtherCAT” tab of the EtherCAT device shows the configured NetId:

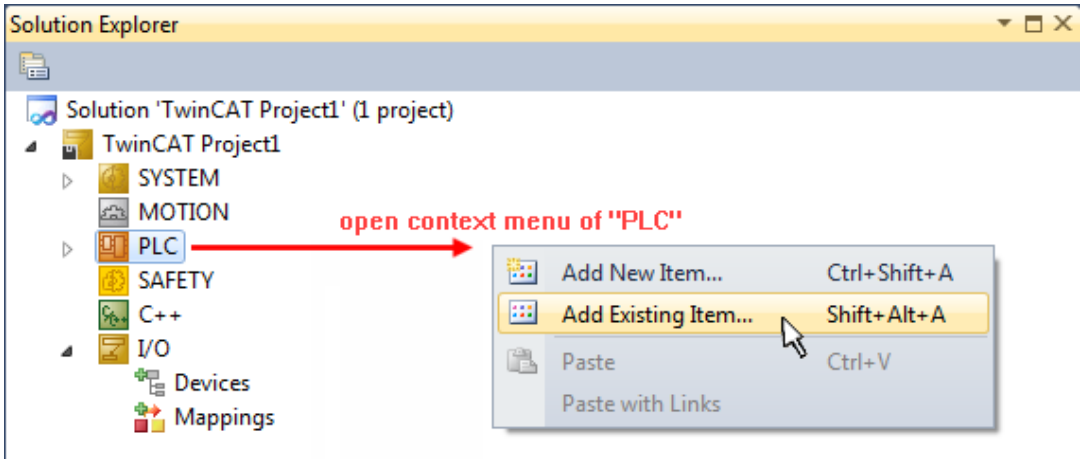


  The first four numbers must be identical with the project NetId of the target system. The project NetId can be viewed within the TwinCAT environment above, where a pull down menu can be opened to choose a target system (by clicking right in the text field). The number blocks are placed in brackets there next to each computer name of a target system.
- Modify the NetId: By right clicking on “EtherCAT device” within the solution explorer a context menu opens where “Change NetId...” have to be selected. The first four numbers of the NetId of the target computer must be entered; both last values are 4.1 usually.
  Example:
  - NetId of project: myComputer (123.45.67.89.1.1)
  - Entry via „Change NetId...“: 123.45.67.89.4.1

Preparation to start the sample program (tpzip file/ TwinCAT 3)

- After clicking the Download button, save the zip file locally on your hard disk, and unzip the *.tpzip -archive file into a temporary working folder.
- Create a new TwinCAT project as described in section:
  TwinCAT Quickstart, TwinCAT 3, Startup [▶ 139]

- Open the context menu of "PLC" within the "Solutionexplorer" and select "Add Existing Item..."



- Select the beforehand unpacked .tpzip file (sample program).

5.3.1 Sample program 1 and 2 (offset/gain)

Download TwinCAT 3 project:

https://infosys.beckhoff.com/content/1033/epp3504/Resources/2152667403.zip

Program description / function:

- Calculation of an *Offset* (correction) value on the basis of the amplitudes of an AC input voltage (DC component ≠ 0) until a deviation of the offset smaller than "wOFFSET_MIN_VAL_REF" (in digits) is achieved.
- Calculation of a *Gain* correction value by presetting via "nPRESET_MAX_VAL" (in digits).

The configuration of the minimum permitted input frequency, the order of the Gain and Offset calculations, and the direct writing to the CoE directory ("PAI Scaler Settings" object) can be done in this sample program (see Variable declaration).

The following procedure is foreseen:

1. Configuration of "*bWriteToCoEEnable*" = TRUE, i.e. on completion of the calculation of the correction values, they are written to the CoE object "PAI Scaler Settings".
2. Set the box module to "Extended Range" (0) via the object "PAI Settings Ch. 1" 0x8000:2E in the CoE directory.
3. Connect a periodic signal (triangle, sine, square, …) to the box module within the selected voltage/ current range via the PAI Settings object 0x8000:01 (Interface).
4. Start the program by setting "*bEnable*" to "TRUE".
5. The end of the execution is recognizable by the variables "bScaleGainDone" and "bScaleOffsetDone", which are then both TRUE.
6. If writing is enabled in the CoE ("*bWriteToCoEEnable*" = TRUE), the values determined should have been written to the object "PAI Scaler Settings" in the CoE directory (see variable "*bError*").
7. If 6th was executed, the box module can be set to "Linear" (1) via the object "PAI Settings Ch. 1" 0x8000:2E in the CoE directory. As a result, the box module also performs the correction calculation internally (see: "*nScaledSampleVal*").

Comments:

Alternatively, the TC3 Analytics Library (TF3510) can be used instead of the function block "FB_GET_MIN_MAX". The function block "FB_ALY_MinMaxAvg_1Ch" can also be used for the determination of the min./max. values. The total calculation can then also be modified in this program by using the mean value provided by this function block.

In the case of the EPP3504/ ERP3504 box module , the "PAI Scaler Settings" object is 0x80n6, in addition to which the *nOffset* and *nGain* variables can also be directly written without the type conversion (REAL to DINT); scaling of the amplitude correction values with 65536 is also no longer necessary.

Example program 1 and 2 variable declaration:

```
PROGRAM MAIN
VAR_INPUT
   bEnable               :BOOL; // Start the code (Offset / Gain adjust)
   nPAI_Sample AT%I*     :DINT; // Input samples of the measurement value
END_VAR
VAR
   // Enter your Net-Id here:
   userNetId             :T_AmsNetId := 'a.b.c.d.x.y';
   // Enter EtherCAT device address here:
   nUserSlaveAddr        :UINT := 1002; // Check, if correct
   // Configurations:
   fMinFrequencyIn       :REAL:=1.5; // Hz
   bScalingOrder         :BOOL:=FALSE; // TRUE: Start scale offset first
   bWriteToCoEEnable     :BOOL:=FALSE; // TRUE: Enable writing to CoE
   // ================================================
   // "Main" State controlling Offset/Gain adjusting:
   nMainCal_State        :BYTE:=0;
   // For CoE Object 0x8005 access:
   fb_coe_write          :FB_EcCoESdoWrite; // FB for writing to CoE
   nSTATE_WRITE_COE      :BYTE := 0;
   nSubIndex             :BYTE;
   nCoEIndexScaler       :WORD := 16#8005; // Use channel 1
   // For EPP3504/ ERP3504 this is 0x8006
   nSubIndScalGain       :BYTE := 16#02;
   nSubIndScalOffs       :BYTE := 16#01;
   nADSErrId             :UDINT; // Copy of ADS-Error ID
   // ================================================
   fb_get_min_max        :FB_GET_MIN_MAX; // Min/Max values needed
   // Note: you may also use "FB_ALY_MinMaxAvg_1Ch" of TwinCAT analytics)
   // instead; there avg (average values can also be determinated
   // Variables used for offset scaling:
   nSTATE_SCALE_OFFSET   :INT := 0;
   bScaleOffsetStart     :BOOL := FALSE;
   bScaleOffsetDone      :BOOL := FALSE;
   fOffsetDeviationVal   :REAL;
   nOFFSET_MIN_VAL_REF   :WORD := 200;  // Max. limit value for offset
   // Variables used for gain scaling:
   nSTATE_SCALE_GAIN     :INT := 0;
   bScaleGainStart       :BOOL := FALSE;
   bScaleGainDone        :BOOL := FALSE;
   nPRESET_MAX_VAL       :REAL := 3000000; // Target amplitude value
```

```
   // ============================================
   // Variables for evaluating of gain and offset:
   nOffset               :REAL := 0; // Offset value
   nGain                 :REAL := 1; // Gain value
   nScaledSampleVal      :REAL;
   nDINT_Value           :DINT;
   fb_trig_bEnable       :R_TRIG; // Trigger FB for Enable
   bError                :BOOL := FALSE; // Evaluate..
END_VAR
```

Execution part:

```
// THIS CODE IS ONLY AN EXAMPLE - YOU HAVE TO CHECK APTITUDE FOR YOUR APPLICATION
// Example program 1 and 2 program code:
// ====================================
// 1. PAI setting of 0x80n0:2E must be "Extended Range" at first
// 2. When writing of scaling values were done, switch to "Linear"

// Calculation of the temporary value (..and use for ScopeView to check)
nScaledSampleVal := nOffset + nGain * DINT_TO_REAL(nPAI_Sample);
// Main-State Procedure:
CASE nMainCal_State OF
   0:
      fb_trig_bEnable(CLK:=(bEnable AND NOT bError));
      IF fb_trig_bEnable.Q THEN // Poll switch or button
         // Initialize temporary offset and gain values:
         nOffset:= 0;
         nGain  := 1;
         bScaleOffsetStart := bScalingOrder;
         bScaleGainStart   := NOT bScalingOrder;

         fb_get_min_max.nMinFreqInput := fMinFrequencyIn;

         nMainCal_State := 10; // Start
      END_IF
   10:
      IF (bScaleGainDone AND NOT bScalingOrder)
        OR (bScaleOffsetDone AND bScalingOrder) THEN
         bScaleOffsetStart := NOT bScalingOrder;
         bScaleGainStart   := bScalingOrder;
         nMainCal_State := nMainCal_State + 10;
      END_IF
   20:
      IF bScaleGainDone AND bScaleOffsetDone THEN
         nMainCal_State :=0; // All done, initalization for next start
      END_IF
END_CASE

// ----- Offset scaling (program 1) -----
IF bScaleOffsetStart THEN
   CASE nSTATE_SCALE_OFFSET OF
   0:
      bScaleOffsetDone := FALSE; // Initialization of confirmation flag
      // Get min/max values within a period of the signal:
      fb_get_min_max(nInputValue:=nScaledSampleVal);
      IF fb_get_min_max.bRESULT THEN // Wait if Limit-Values are valid
         // Min/Max Values valid, continue..
         // calculate current offset deviation:
         fOffsetDeviationVal :=
         (fb_get_min_max.nMaxVal - ABS((fb_get_min_max.nMaxVal-fb_get_min_max.nMinVal)/2));

         // Offset deviation check:
         IF ABS(fOffsetDeviationVal) < nOFFSET_MIN_VAL_REF THEN
            // Deviation in acceptable range - offset scaling done,
            // now write correction value into CoE Object:
            nDINT_Value := REAL_TO_DINT(nOffset);

            // Initiate writing to CoE:
            nSubIndex := nSubIndScalOffs;
            nSTATE_WRITE_COE := 10;
            nSTATE_SCALE_OFFSET := nSTATE_SCALE_OFFSET + 10;
         ELSE
            // Calculate new offset value (new by old with deviation)
            nOffset := nOffset - fOffsetDeviationVal;
         END_IF
      END_IF
   10:
      IF(nSTATE_WRITE_COE = 0) THEN
         // Scaling offset done within CoE for the device
         bScaleOffsetDone := TRUE;
```

```
         bScaleOffsetStart := FALSE;
         nSTATE_SCALE_OFFSET := 0;
      END_IF
   END_CASE
END_IF

// ----- Gain scaling (program 2) -----
IF bScaleGainStart THEN
   CASE nSTATE_SCALE_GAIN OF
   0:
      bScaleGainDone := FALSE; // Initialization of confirmation flag
      // Get min/max values within a period of the signal:
      fb_get_min_max(nInputValue:=DINT_TO_REAL(nPAI_Sample));
      IF fb_get_min_max.bRESULT THEN // Wait if Limit-Values are valid

         // Calculate Gain
         nGain := nPRESET_MAX_VAL/ABS((fb_get_min_max.nMaxVal-fb_get_min_max.nMinVal)/2);
         // ..shift gain value by 16 Bit left and convert to DINT:
         nDINT_Value := REAL_TO_DINT(65536 * nGain);

         //Due to 'output = gain * input + offset', the offset have to be adapted:
         nOffset := nOffset * nGain;

          // Initiate writing to CoE:
         nSubIndex := nSubIndScalGain;
         nSTATE_WRITE_COE := 10;
         nSTATE_SCALE_GAIN := nSTATE_SCALE_GAIN + 10;
      END_IF
   10:
      IF(nSTATE_WRITE_COE = 0) THEN
         IF NOT (nOffset = 0) THEN
            // (bScalingOrder is TRUE)
            nDINT_Value := REAL_TO_DINT(nOffset);
            // Initiate writing to CoE (again):
            nSubIndex := nSubIndScalOffs;
            nSTATE_WRITE_COE := 10;
         END_IF

         nSTATE_SCALE_GAIN := nSTATE_SCALE_GAIN + 10;
      END_IF
   20:
      IF(nSTATE_WRITE_COE = 0) THEN
         // Scaling gain done within CoE for the device
         bScaleGainStart := FALSE;
         bScaleGainDone := TRUE;
         nSTATE_SCALE_GAIN := 0; // Set initial state
      END_IF
   END_CASE
END_IF

IF (nSTATE_WRITE_COE > 0) THEN
   IF bWriteToCoEEnable THEN
      CASE nSTATE_WRITE_COE OF
      10:
         // Prepare CoE write access
         fb_coe_write(
            sNetId:=     userNetId,
            nSlaveAddr:= nUserSlaveAddr,
            nIndex:=     nCoEIndexScaler,
            bExecute:=   FALSE,
            tTimeout:=   T#1S
         );
         nSTATE_WRITE_COE := nSTATE_WRITE_COE + 10;
      20:
         // Write nDINT_Value to CoE Index "Scaler":
         fb_coe_write(
         nSubIndex:= nSubIndex,
         pSrcBuf:= ADR(nDINT_Value),
         cbBufLen:= SIZEOF(nDINT_Value),
         bExecute:= TRUE
         );
         nSTATE_WRITE_COE := nSTATE_WRITE_COE + 10;
      30:
         fb_coe_write();
         IF NOT fb_coe_write.bBusy THEN
            nSTATE_WRITE_COE := 0;
         END_IF
      END_CASE
   ELSE
      nSTATE_WRITE_COE := 0;
```

```
   END_IF
END_IF

IF(fb_coe_write.bError) AND NOT bError THEN
   bError := TRUE;
   nADSErrId := fb_coe_write.nErrId;
   // CoE write acccess error occured: reset all
   nSTATE_WRITE_COE := nMainCal_State := 0;
   bScaleOffsetDone := bScaleOffsetStart := FALSE;
   bScaleGainDone   := bScaleGainStart   := FALSE;
END_IF
```

5.3.1.1 Function block FB_GET_MIN_MAX

Declaration part:

```
FUNCTION_BLOCK FB_GET_MIN_MAX
VAR CONSTANT
   CMAXinit       :REAL := -3.402823E+38;
   CMINinit       :REAL := 3.402823E+38;
END_VAR
VAR_INPUT
   bInit          :BOOL := TRUE;
   nInputValue    :REAL;
   nMinFreqInput :REAL;
END_VAR
VAR_OUTPUT
   bRESULT        :BOOL;
   nMaxVal        :REAL;
   nMinVal        :REAL;
END_VAR
VAR
   CMMcnt         :UINT;
   nMaxValCnt     :UINT;
   nMinValCnt     :UINT;
   bValidMinVal   :BOOL;
   bValidMaxVal   :BOOL;
   fbGetCurTaskIdx : GETCURTASKINDEX;
END_VAR
```

Execution part:

```
IF bInit THEN
   // Counter initialization:
   // [counter value] > [1/(<input frequency> * TaskCycleTime)]
   fbGetCurTaskIdx();
   CMMcnt := REAL_TO_UINT(
      1.1E7/(nMinFreqInput*UDINT_TO_REAL(
      _TaskInfo[fbGetCurTaskIdx.index].CycleTime)));
   // At least an entire period have to be sampled for min/max determination
   // Initialization, go on:
   nMaxValCnt :=CMMcnt;
   nMinValCnt :=CMMcnt;
   nMaxVal :=CMAXinit;
   nMinVal :=CMINinit;
   bInit := FALSE;
END_IF
// Assertions: new min/max values exists:
bValidMaxVal := TRUE;
bValidMinVal := TRUE;
// Filter min/max values
IF (nMaxVal < nInputValue) THEN
   bValidMaxVal := FALSE;
   nMaxVal := nInputValue; // Max value was found
END_IF
IF (nMinVal > nInputValue) THEN
   bValidMinVal := FALSE;
   nMinVal := nInputValue; // Min value was found
END_IF
// Count down, if no new value come in:
IF (bValidMaxVal AND (nMaxValCnt > 0)) THEN
   nMaxValCnt := nMaxValCnt - 1;
END_IF
// Count down, if no new value come in:
IF (bValidMinVal AND (nMinValCnt > 0)) THEN
   nMinValCnt := nMinValCnt - 1;
END_IF
IF ((nMaxValCnt = 0) AND (nMinValCnt = 0)) THEN
```

```
   // Consequence: min/max determined
   bInit := TRUE; // Prepare next call
   bRESULT := NOT (nMaxVal = nMinVal); // Sign valid results
ELSE
   bRESULT := FALSE; // Sign still invalid results
END_IF
```

5.3.2 Sample program 3 (write LookUp table)

Download TwinCAT 3 project: https://infosys.beckhoff.com/content/1033/epp3504/Resources/2152669707.zip

Program description

Transmission of LookUp table interpolation values for mapping of an equation $f(x) = x^3$ via CoE into the box module.

Variable declaration sample program 3

```
PROGRAM MAIN
VAR
   //LookUp-Table (LUT) generated by: MBE * x³
   aLUT:ARRAY[0..99] OF DINT :=
   [
      -7812500,-7812500,-7493593,-6894382,
      -7174765,-6051169,-6855859,-5279674,-6536953,-4576709,
      -6218125,-3939087,-5899218,-3363620,-5580390,-2847120,
      -5261484,-2386402,-4942578,-1978275,-4623750,-1619555,
      -4304843,-1307052,-3985937,-1037580,-3667109,-807951,
      -3348203,-614978,-3029375,-455472,-2710468,-326248,
      -2391562,-224117,-2072734,-145892,-1753828,-88385,
      -1434921,-48409,-1116093,-22776,-797187,-8300,
      -478281,-1792,-159453,-66,159453,66,
      478281,1792,797187,8300,1116093,22776,
      1434921,48409,1753828,88385,2072734,145892,
      2391562,224117,2710468,326248,3029375,455472,
      3348203,614978,3667109,807951,3985937,1037580,
      4304843,1307052,4623750,1619555,4942578,1978275,
      5261484,2386402,5580390,2847120,5899218,3363620,
      6218125,3939087,6536953,4576709,6855859,5279674,
      7174765,6051169,7493593,6894382,7812500,7812500
   ];
   // For CoE 0x8000 and 0x8005 - write values:
   // ================================================
   wCoEIndexScaler :WORD := 16#8005; // CoE Index
   wState          :BYTE := 0; // Write status
   fb_coe_writeEx  :FB_EcCoESdoWriteEx; // Function Block for writing in CoE
   userNetId       :T_AmsNetId := '172.128.1.1.5.1'; // Have to be entered
   userSlaveAddr   :UINT := 1003; // Have to be entered
   bWriteLUT2CoE   :BOOL:=FALSE; // Sign for start writing
   bError          :BOOL:=FALSE; // Sign for any error
END_VAR
```

Remarks:

- The variable "startWrite" (BOOL) is also declared in sample program 4.
- The variable 'userNetId' must include the EtherCAT net ID of the device. It can be viewed in the "EtherCAT" tab under "Device (EtherCAT)".
- The variable "userSlaveAddr" must contain the EtherCAT address of the box module.

Sample program for transferring the LookUp table:

Execution part:

```
// Example program 3:
// ###### Write LookUp table into CoE object 0x8005: #######
IF bWriteLUT2CoE THEN
CASE wState OF
   0:
      fb_coe_writeEx(bExecute := FALSE);// Prepare CoE-Access
      wState := wState + 1;// Next state
   1:
```

```
      // Write 100 X/Y LookUp-Table entries
      fb_coe_writeEx(
         sNetId:= userNetId,
         nSlaveAddr:= userSlaveAddr,
         nSubIndex:= 1,
         nIndex:= wCoEIndexScaler,
         pSrcBuf:= ADR(aLUT),
         cbBufLen:= SIZEOF(aLUT),
         bCompleteAccess:= TRUE,
         bExecute:= TRUE
         );
      wState := wState + 1; // Next state
   2:
      // Proceed with writing to CoE
      fb_coe_writeEx();
      IF NOT fb_coe_writeEx.bBusy THEN
         wState := 0;// Done
         bWriteLUT2CoE := FALSE;
         bError := fb_coe_writeEx.bError; // See nErrId if TRUE
      END_IF
END_CASE
END_IF
```

A simple variable query, e.g., via button linked with bEnable, can be used to initiate the transfer. The variable declaration must contain

```
VAR_INPUT
   bEnable   AT%I*   :BOOL;
END_VAR
```

and the following program lines:

```
IF bEnable AND NOT startWrite THEN
   bWriteLUT2CoE := TRUE;
END_IF
```

5.3.3 Sample program 4 (generate LookUp table)

Download TwinCAT 3 project: https://infosys.beckhoff.com/content/1033/epp3504/Resources/2152669707.zip

Program description / function:

Inclusion of LookUp table interpolation values from a box module input signal to a field variable (and optional subsequent transfer of the LookUp table interpolation values via CoE access to the box module using sample program 3).

It is envisaged to use a ramp generator with a trigger input, whose level, in conjunction with an input of a digital input terminal (e.g., EL1002) sets the variable "*bStartRecord*" to TRUE via a link (e.g., push button connected to +24 V). This allows recording of the values to be synchronized with the ramp input voltage. Alternatively, an output terminal can be used (e.g., EL2002), whose output controls the trigger input and whose output is then set to TRUE via the TwinCAT development environment ("*bStartRecord*" would then have to be declared as AT%Q* and linked to a terminal output).

Variable declaration sample program 4

```
// Variable declaration for example program 4
PROGRAM MAIN
VAR CONSTANT
   nEndX                        : BYTE := 50; // Number of support values
END_VAR
VAR
   nPAISampleIn            AT%I* : DINT; // PDO PAISamples
   bStartRecord            AT%I* : BOOL; // +Electrical junction to trigger ramp
   bGetMinMax                    : BOOL := FALSE;
   bRecordLUT                    : BOOL := FALSE;
   r_trigStartRecord             : R_TRIG;
   nX                            : BYTE := 0;
   aValues                       : ARRAY[0..nEndX-1] OF DINT;
   nYstepValue                   : DINT;
   tp_timer                      : TP;
   ton_timer                     : TON;
   nMinValue                     : DINT := 7812500;
```

```
   nMaxValue                    : DINT := -7812500;
   nYvalue                      : DINT;
   tRepeatTimerValue            : TIME := T#51MS;
   aLUT                         : ARRAY[0..99] OF DINT;
END_VAR
```

Execution part:

```
// Example program 4:
// ################# Recording of 50 sample points: #################
// a) Determination of min./max. values (corresponding to the value range of the sensor)
tp_timer(IN:=bGetMinMax, PT:=T#2.51S); // Periodic duration of ramp (+reserve)
IF tp_timer.Q THEN
   nMinValue := MIN(nPAISampleIn, nMinValue);
   nMaxValue := MAX(nPAISampleIn, nMaxValue);
END_IF
// b) Recording of values: Start
r_trigStartRecord(CLK:=bStartRecord);
IF r_trigStartRecord.Q THEN
   nX := 0;
   memset(ADR(aLUT), 0 , 100);
   bRecordLUT := TRUE;
END_IF
ton_timer();
IF bRecordLUT OR ton_timer.Q THEN
   bRecordLUT := FALSE;
   ton_timer(IN:=FALSE);
   IF(nX < nEndX) THEN
      // b.1) Record of values:
      aValues[nX] := nPAISampleIn;
      nX := nX + 1;
      ton_timer(IN:=TRUE, PT:=tRepeatTimerValue); // T=2.5s/49 = 51ms
   ELSE
      // b.2) Recording end:
      // Create linearized values:
      nYstepValue := (nMaxValue - nMinValue) / nEndX; // Y steps
      nYvalue := aValues[0];  // Common start value of the LUT
      FOR nX:=0 TO nEndX DO
         // Create LUT (X = actual values, Y = target values):
         aLUT[nX*2] := aValues[nX]; // X value
         aLUT[nX*2+1] := nYvalue;   // Y value
         // next Y value of the LUT (make a "straight"):
         nYvalue := nYvalue + nYstepValue; // f(x) = b+x
END_FOR
   END_IF
END_IF
```

5.3.4 Sample program 5 (write filter coefficients)

Download TwinCAT 3 project: https://infosys.beckhoff.com/content/1033/epp3504/Resources/2152672011.zip

Program description

Transmission of exemplary filter coefficients via CoE access into the box module.

General settings

- The function block "FB_EcCoESdoWrite" requires the "Tc2_EtherCAT" library
- <AmsNetID> must show the local device EtherCAT NetID in inverted commas (e.g. '168.57.1.1.5.1')
- <DeviceEtherCATAddress> must show the local device EtherCAT address of the EPP3504-0023/ ERP3504-0022 box module (e.g., 1007)

Variable declaration sample program 5

```
PROGRAM MAIN
// Variable declaration example program 5
VAR CONSTANT
NumOfFilterCoeff                :BYTE:=40;
END_VAR
VAR
// Function block of library "Tc2_EtherCAT" for CoE Object access:
```

```
fb_coe_write                    :FB_EcCoESdoWrite;
userNetId                       :T_AmsNetId := '???';
userSlaveAddr                   :UINT := ???;
// Writing PLC state for coefficients transfer (Set to 0 for start)
wState                          :BYTE:=255;
index                           :BYTE:=1; // Start index for coefficients transfer
wCoEIndexUserFilterCoeffizents  :WORD:=16#8001;
aFilterCoeffs:ARRAY[0..NumOfFilterCoeff] OF LREAL :=
   [
   // Example filter coefficients FIR band pass: 3600..3900 Hz
   // Usage: "User defined FIR Filter" (32)
      0.03663651655662163,
      0.04299467480848277,
      -0.007880289104928245,
      0.0664029021294729,
      -0.0729038234874446,
      -0.00005849791174519834,
      0.05628409460964408,
      -0.0525134329294473,
      0.026329003448584205,
      0.00027114381194760643,
      -0.03677629552114248,
      0.06743018479714939,
      -0.0560894442193289,
      0.0009722394088121363,
      0.05676876756757213,
      -0.07775650809213645,
      0.05330627422911416,
      0.0009941073749156226,
      -0.055674804078696793,
      0.07874009379691002,
      -0.055674804078696793,
      0.0009941073749156226,
      0.05330627422911416,
      -0.07775650809213645,
      0.05676876756757213,
      0.0009722394088121363,
      -0.0560894442193289,
      0.06743018479714939,
      -0.03677629552114248,
      0.00027114381194760643,
      0.026329003448584205,
      -0.0525134329294473,
      0.05628409460964408,
      -0.00005849791174519834,
      -0.0729038234874446,
      0.0664029021294729,
      -0.007880289104928245,
      0.04299467480848277,
      0.03663651655662163,
      0
   ];
nValue :DINT; // Temporary variable
END_VAR
```

Execution part:

```
// Example program 5:
// writes filter coefficients of
// "User defined FIR Filter" (32)
// incl. example coefficients for band pass
// Note: writing possible, if CoE Object
// PAI Settings Ch.1 (0x8000:16) has value 32 or 33 set, only!
// (32 = User defined FIR Filter / 33 = User defined IIR Filter)
// ===============================================================
CASE wState OF
   0:
      fb_coe_write(bExecute := FALSE);// Prepare CoE access
      wState := wState + 1;// Go to next state
   1:
      //nValue := REAL_TO_DINT(DINT_TO_REAL(aFilterCoeffs[index]) *16384);
      nValue := LREAL_TO_DINT(aFilterCoeffs[index] * 1073741824); // Bit-shift factor: 2^30
      // Write filter coefficients (max. 40 entries)
      fb_coe_write(
         sNetId:= userNetId,
         nSlaveAddr:= userSlaveAddr,
         nSubIndex:= index,
         nIndex:= wCoEIndexUserFilterCoeffizents,
         pSrcBuf:= ADR(nValue),
         cbBufLen:= SIZEOF(nValue),
```

```
        bExecute:= TRUE,
        tTimeout:= T#1S
      );
      wState := wState + 1; // Go to next state
   2:
      // Execute writing to CoE
      fb_coe_write();
      IF fb_coe_write.bError THEN
         wState := 100; // Error case
      ELSE
         IF NOT fb_coe_write.bBusy THEN
         index := index + 1;
            IF index <= (NumOfFilterCoeff) THEN
               fb_coe_write(bExecute := FALSE);// Prepare the next CoE access
               wState := 1;// Write next value
            ELSE
               wState := 255;// Done
            END_IF
         END_IF
      END_IF
   100:
      ; // Error handling
   255:
      ; // Go on..
END_CASE
```

5.3.5 Sample program 6 (interlacing of measured values)

Program description / function

Note on this chapter: The use of EL3751/ELM3xxx terminals also applies accordingly to EPP35xx.

In some use cases a particularly fine temporal resolution of the signal is desired, e.g. so that many measuring points are available for an FFT. Two ways to do this are shown below:

- Use of an analog input terminal with the correspondingly high sampling rate, e.g. 20 ksps.
- Use of two analog input terminals with half the sampling rate, i.e. 10 ksps, and so-called *interlacing of measured values*; the result is likewise a 20 ksps sampling of the signal.

The second way is described in this sample: Use of two EL3751 EtherCAT Terminals, each with a maximum sampling rate of 10 kSps (and thus a conversion time of 100 µs in this case, cf. Further documentation for I/O components with analog in and outputs [▶ 252], chapter “Temporal aspects of analog/digital or digital/analog conversion”). Due to their parallel connection, both terminals are fed the same signal simultaneously and are configured by Distributed Clocks in such a way that they sample not at the same time, but offset by half the conversion time (in this case: 50 µs). If the two measured data streams are now combined alternately in the controller, i.e. "interlaced", the result is a net measured data stream of 20 ksps.

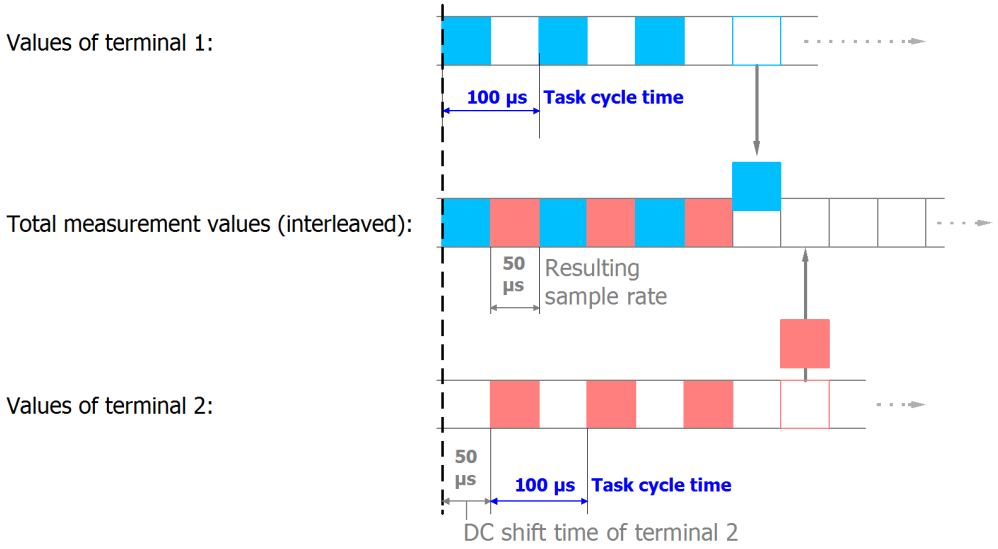


Fig. 28: Process of interlacing the input data

The following configuration is used for this purpose:

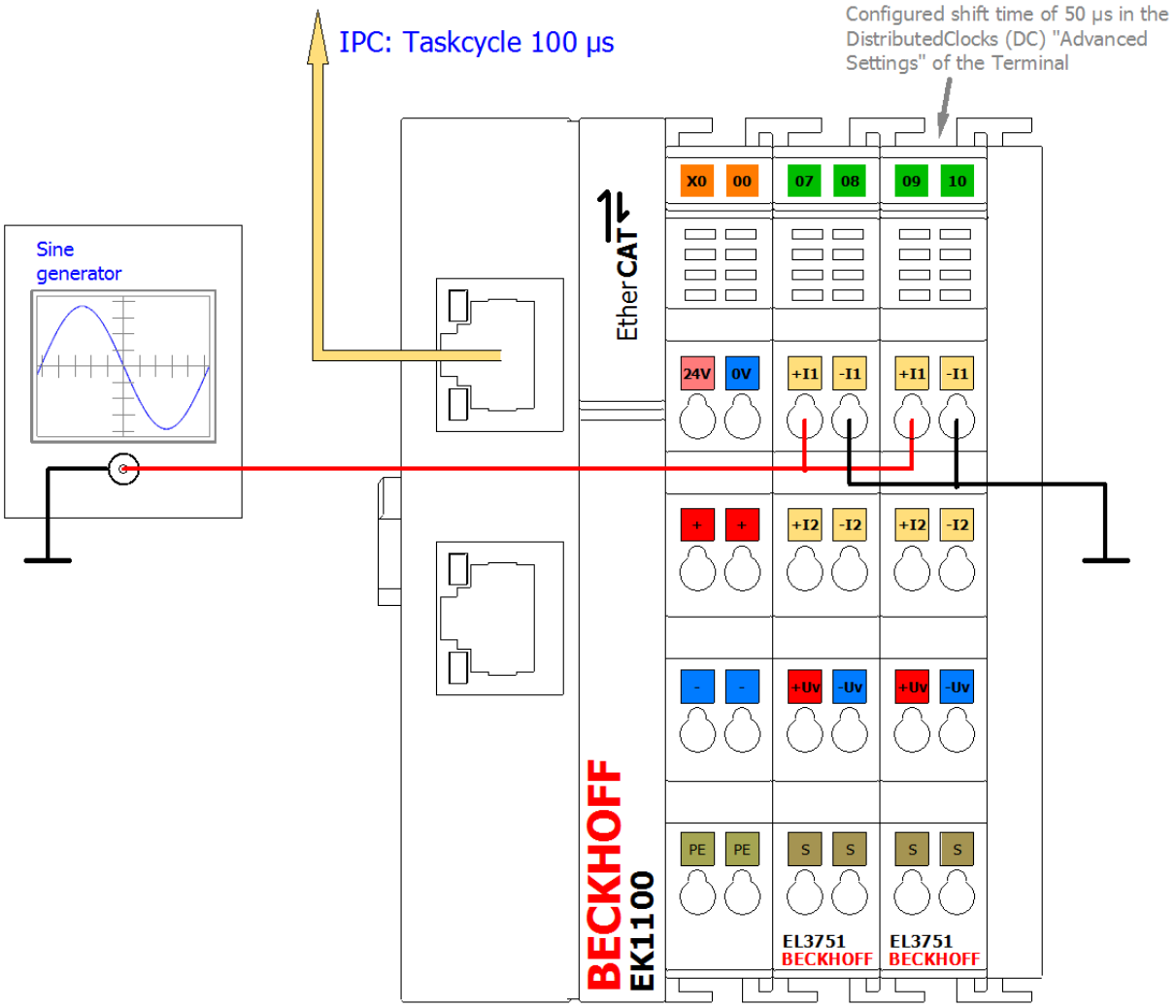


Fig. 29: Configuration and setup for sample program 6: Doubling of the sample rate with 2 x EL3751

The sample is also available with corresponding adaptations for other EL3xxx/ELM3xxx terminals or box modules. There may then be different oversampling factors, shift times, etc. The optionally existing task with 50 µs in sample 6a may then also not be usable.

So that the input values can be successively combined to form a total value, a corresponding shift time is necessary for each channel/terminal; in this sample 50 µs for the second terminal. This is set in the "Advanced settings" for Distributed Clocks ("DC" tab) for the second terminal:

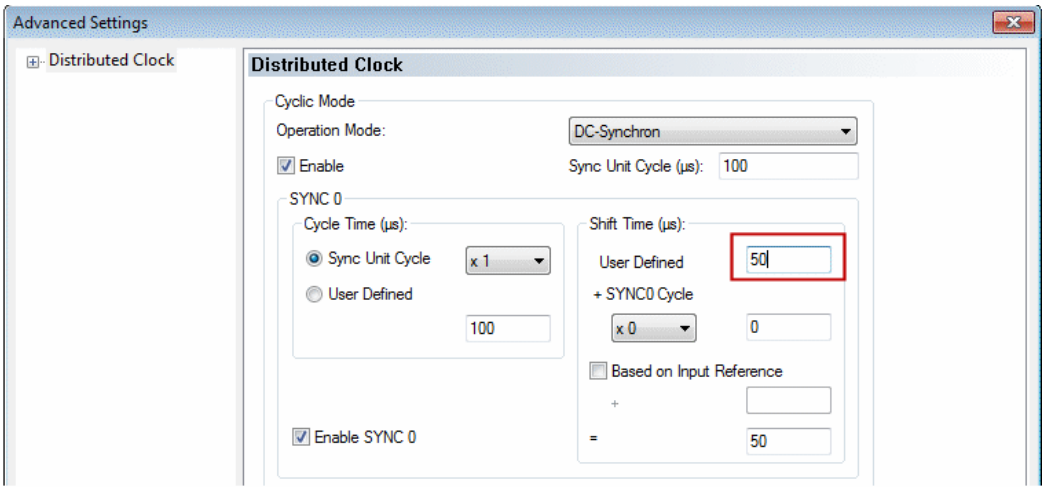


Fig. 30: Setting the DC shift time for terminal 2

Some notes and restrictions

- This principle can be implemented with two (as described above) or more terminals; the limit is the shift time fineness of 1 µs.
- The terminals used must support Distributed Clocks. Oversampling is helpful, but not necessary. The sampling methods simultaneous vs. multiplex must be considered; see corresponding documentation with the question: *"when the channels sample their values in relation to Distributed Clocks"*.
- Although this approach doubles the sampling rate of the signal under observation, the frequency response and attenuation specified in the technical data for the terminal still apply! It is therefore not possible to read signals that are twice as fast with twice the sampling rate. Sample: the EL3751 with a sampling rate of 10 ksps can meaningfully (alias-free) read signals up to half the sampling rate = 5 kHz. This limit remains even with multiple parallel sampling! The attenuation of -3 dB at 3 kHz given as an example also applies to the interlaced sum signal.
- Only one EtherCAT terminal can be functionally time-shifted as a whole by Distributed Clocks shift time, not the individual channel of a terminal. The shift then affects all the channels of a terminal. Therefore, for the given principle, two or more terminals/box modules must always be used; the interlacing of two channels of the same terminal/box is not possible.
- The specified measurement uncertainty must be observed: the unavoidably different real measurement uncertainty and thus the amplitude difference between the two terminals or their channels used on the same signal can become visible as a noise component after interlacing. Therefore, terminals should be used for this principle that exhibit a much smaller measurement uncertainty than is necessary for the application. It is expressly recommended to carry out an explicit user calibration of at least the offset of the two electrically interconnected channels in order to minimize this effect.
- Terminals with the same HW/FW version should be used.

Sample program

This setting, like the base time and the task cycle time, is already configured in the sample program:

Download TwinCAT 3 project / sample program 6a: https://infosys.beckhoff.com/content/1033/epp3504/Resources/4867888523.zip

In the following section, the simplest form of input value interlacing in Structured Text is initially shown with oversampling = 1 for each input value: each of two elements of a field variable receives a value from a terminal. The variable can be used for further processing and is shown here in the TwinCAT ScopeView. In the EL3751 the programming instructions are assigned to a 100 µs task:

Variable declaration sample program 6a

```
PROGRAM MAIN
VAR
   nSamples_1         AT%I*      :DINT; // EL3751 input with no added shift time
   nSamples_2         AT%I*      :DINT; // EL3751 input with -50 µs added shift time
   aCollectedResult              :ARRAY[0..1] OF DINT;
END_VAR
```

Execution part:

```
// Example program 6a:
// 100 µs task
// ============================================================
aCollectedResult[0] := nSamples_1;   // Put 1st Value of sequence into array
// Pattern: 1.1.1.1...
aCollectedResult[1] := nSamples_2;   // Put n-th Value of sequence into array (2nd here)
// Pattern: .2.2.2.2...
// ============================================================
// Result pattern: 12121212... (--> see scope view dots)
```

For an input signal with sine 5 kHz and 2.5 V amplitude, for example, the TwinCAT ScopeView provides the following results:

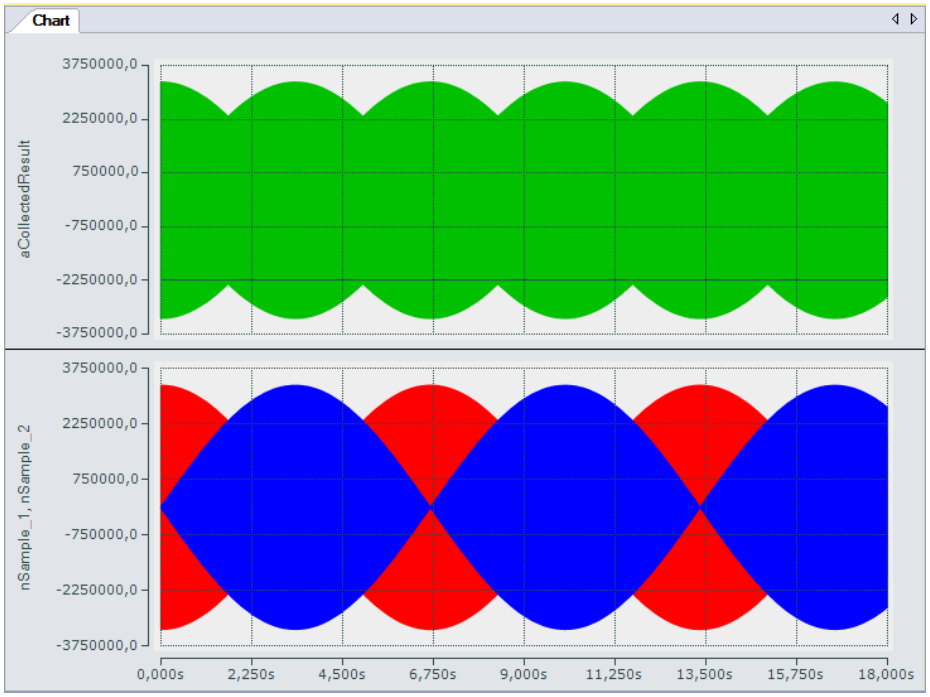


Fig. 31: Oversampling 20 ksps with 2 x EL3751 with input signals (below) and result signal (top)

The upper diagram shows the total signal and the two input signals (nSample_1, nSample_2), with a time shift of 50 µs relative to each other, within 18 s in compressed form. The total input signal (nCollectedResult) indicates the interlacing of the two input signals.

The following diagram (enhanced through highlighting) shows how the input signals (nSample_1, nSample_2) contribute to the structure of the total input signal:

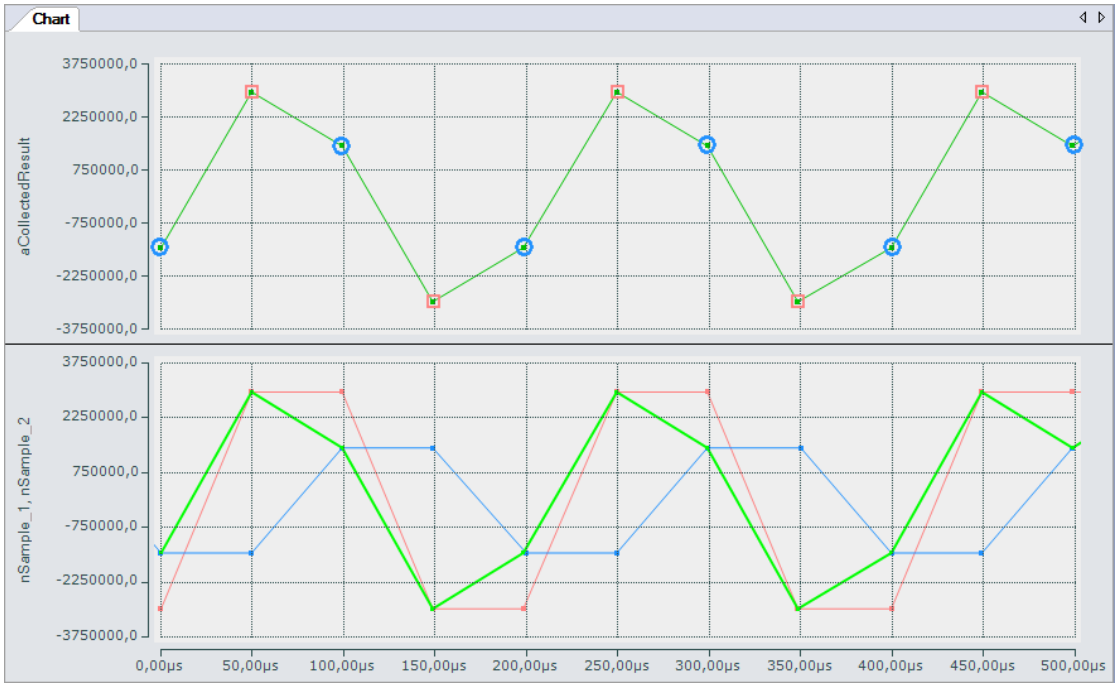


Fig. 32: Oversampling 20 ksps with 2 x EL3751 shows input value 1 and input value 2 alternately for a result value

Under certain conditions, both inputs can be combined into a single variable in a correspondingly fast task. For this purpose the sample program contains an additional task with 50 µs cycle time, which is required for representing the input signals in the SopeView and contains a variable (nCollected) to which both inputs are assigned alternately:

```
// 50 µs task
```

```
// ============================================================
// Junction of the two inputs
nCollected := SEL(nToggle, MAIN.nSamples_1_, MAIN.nSamples_2_);
nToggle := NOT nToggle;
```

The input variables required for the ScopeView are read in this task from the 100 µs task, so that the individual values can be represented at 50 µs intervals.

Variant with 2 x oversampling 10 = oversampling 20

If, for example, an oversampling factor of 10 is used for both input terminals, a field variable is used for the total measured value. A simple loop can be used for interlacing the input values, which reads the values sequentially into a field variable for the resulting result variable:

Variable declaration sample program 6b

```
PROGRAM MAIN
VAR
   aSamples_1          AT%I*      :ARRAY[0..9] OF DINT; // EL3751 input with no added shift time
   aSamples_2          AT%I*      :ARRAY[0..9] OF DINT; // EL3751 input with -50 µs added shift time
   aCollectedResult               :ARRAY[0..19] OF DINT;
// ====================================================
   nPos                           :BYTE;
END_VAR
```

Execution part:

```
// Example program 6b:
// 1 ms task
// ============================================================
FOR nPos := 0 TO 9 DO
   // Put 1st Value of sequence into array:
   aCollectedResult[2*nPos] := aSamples_1[nPos];
   // Put n-th value of sequence into array (2nd here):
   aCollectedResult[2*nPos+1] := aSamples_2[nPos];
END_FOR
```

Download TwinCAT 3 project / sample program 6b: https://infosys.beckhoff.com/content/1033/epp3504/Resources/4867891467.zip

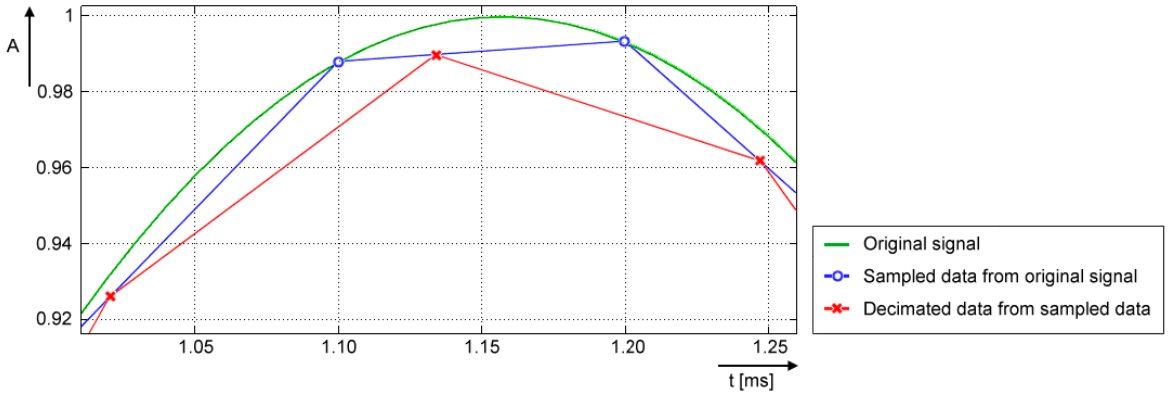
Sample program 6b returns the same result, except that the total input signal is only available in the form of a field variable with 20 elements.

5.3.6 Sample program 7 (general decimation in the PLC)

The EPP3504-0023/ ERP3504-0022 can only decimate their basic sampling rate f_{max} by integer multiples, see chapter "Decimation". To realize any other sampling rates ($f_{target} < f_{max}$) for a channel, you can proceed as follows, for example:

- Operate the box module /channel at the maximum sampling rate and transfer the data to the controller (PLC) via EtherCAT/oversampling

- In the PLC/ C++, on the time axis, convert to the desired sampling rate, e.g., by linear interpolation based on the timestamp for each input value (sample). Since the EPP3504-0023/ ERP3504-0022 units provide time-equidistant samples based on distributed clocks, this is easily possible.
  For example, a sinusoidal signal decimated with 50/44.1 = 1/0.882 can be represented as follows:



  - Green: corresponds to original analog signal (input), approx. 432 Hz
  - Blue (O): corresponds to sampling of the EPP3504-0023/ ERP3504-0022 with f_{max} of 10,000 sps; a sampling interval of 100 µs
  - Red (X): corresponds to signal converted by PLC to 8820 sps (factor 0.882) and thus a time interval of approx. 113.37.. µs
- Note: The term "decimation" is applied both to the calculation in the box module (see chapter "Decimation") and to the conversion in the PLC program. The following refers to the conversion in the PLC.
- Since the time interval of the desired sampling after decimation in the PLC is usually no longer an integer (finite) number, value/time pairs are used for representation in the PLC/Scope, i.e., an X time value is assigned to each Y value. Such value/time pairs can easily be displayed with TwinCAT ScopeView in XY mode. See also infosys.beckhoff.com:
  TwinCAT3 → TExxxx | TC3 Engineering → TE13xx | TC3 ScopeView → Configuration → XY-Graph
- The conversion also has consequences for further processing in PLC/C/ADS:
  - A PLC/EtherCAT/TwinCAT system tends to be set up such that a constant number of samples is processed per cycle. Usually this is now no longer the case: a different number of samples has to be processed from cycle to cycle (specified by the program variable *nResultNoOfSamples*).
  - While a time stamp per signal value has so far remained relatively insignificant, the method of conversion of the decimation process used here, however, means that the respective timestamp per signal value must be taken into account in an elementary manner.
- The non-constant number of samples is not visible in the TwinCAT XY Scope because some values are sporadically drawn twice, and this should be taken into account; it may be advisable to use an intermediate buffer for further processing.
- For orientation of the currently valid number of samples per task cycle, the program provides the variable *nResultNoOfSamples*. It indicates which values in the array variable contain valid values in a task cycle (indicates the field number - 1).

The following **sample program**, which also contains the XY representation in the TwinCAT Scope, serves as a guide. Due to the above-mentioned problem relating to the non-constant number of valid samples, the program returns the array pair *aVarDecResult_TS* and *aVarDecResult* for the Scope with the same number of elements as for the input value *aSamples_1* (*value* = *nOVS*). If there are fewer values in a task run, the last value is simply entered repeatedly (similar to "sample & hold"). The ScopeView was configured as follows for the recording:

Property		Value
ScopeNodeProperties	ViewDetauilLevel	ExtendedXYOnly
	Record time	00:00:00:05
ChartXYNodeProperties	Default Display Width	0,00:00:00,050:000
	Max Data Points	200000
XYChannelNodeProperties	Marks	On
	Mark Size	5
	Mark Color	(other than line color)

For an illustrative representation, the ScopeView recording was started first and then the program, which limits the decimated values to one second:

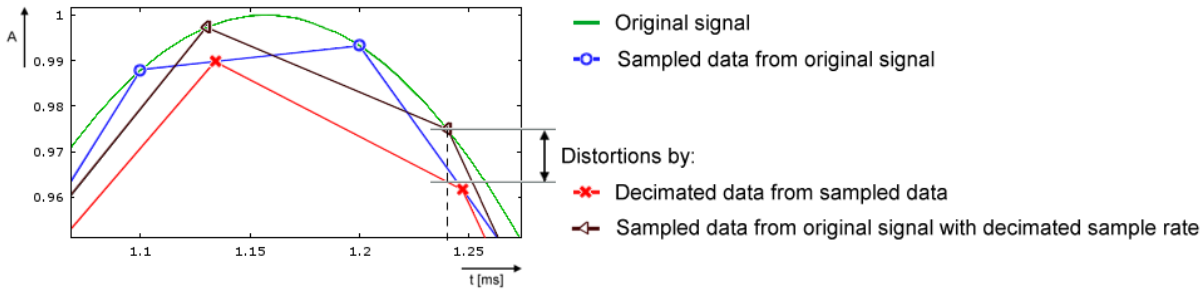
```
   IF nOVS_CycleCount = 1000000000 THEN
      ;
      bEnable := FALSE;// Stop after 1s just for recording
   ELSE
...
```

This line can, of course, be commented out for further adjustments:

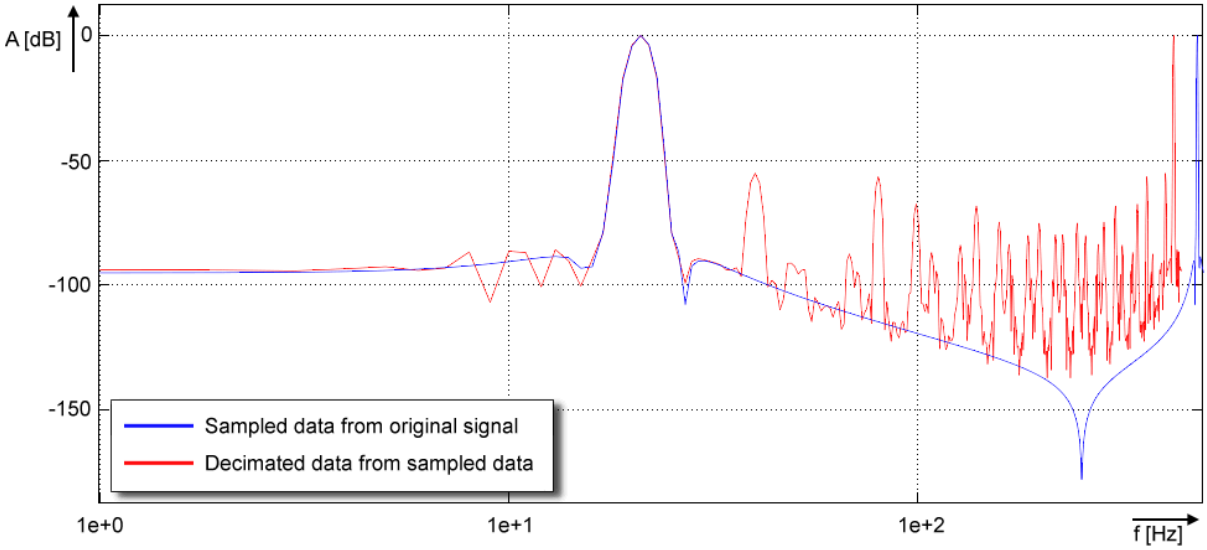
```
      //bEnable := FALSE;// Stop after 1s just for recording
```

Notes:

- the target sampling rate f_{target} should be close to the sample rate f_{max}, so that it is possible to evaluate a time interval between two decimated values. The desired decimation may require further parameters such as task cycle time, oversampling factor etc. to be adjusted both in the configuration and as variable initialization in the sample program (see figure "Process of variable decimation of the sample program", which illustrates the functionality of the program code).
- Basically, the conversion process in this sample program causes distortions in the result in relation to the original signal shape when decimating with fractional rational factors (see signal curve). In concrete terms, deviations from the original signal curve only occur in those sections where the time derivative value (the slope) is not constant. For example, input values of a sine signal in the non-linear sections are distorted by the interpolation performed in the program:



In the frequency spectrum, for example by a calculation with 20 Hz sinus signal, sampled with 500 sps and decimated to 441 sps, this is illustrated as follows:



- If no low-pass filtering corresponding to f_{target} is performed on the data stream, aliasing effects will occur! It is therefore advisable to perform low-pass filtering in the PLC, e.g. with the TC3 Controller Toolbox or the TC3 Filter Library, before the conversion/decimation is performed. Suitable filters can easily be created with the TE1310 FilterDesigner. For more information, see www.beckhoff.com:

Automation → TwinCAT 3 → TE1xxx | TC3 Engineering → TE1310 | TC3 Filter Designer
Alternatively, the filters available in the EPP3504-0023/ ERP3504-0022 can, of course, be set to the suitable low-pass frequency; the TwinCAT Filter Designer is also helpful for this.

- Entries of decimation factors within the program (*nDecimationValue*) should have a value > 1. The program code supports down sampling only.

E.g.: If a terminal such as ELM3602-0002 (2-channel IEPE evaluation) provides a data stream with oversampling of 50 ksps at 100 µs cycle time, this sample code can decimate to 44.1 ksps. In the sample program, the cycle ticks in the task configuration should be changed from 5 to 1 and the corresponding program variable nTaskCycle_ns from 500000 to 100000. See the following image section of ScopeView XY:

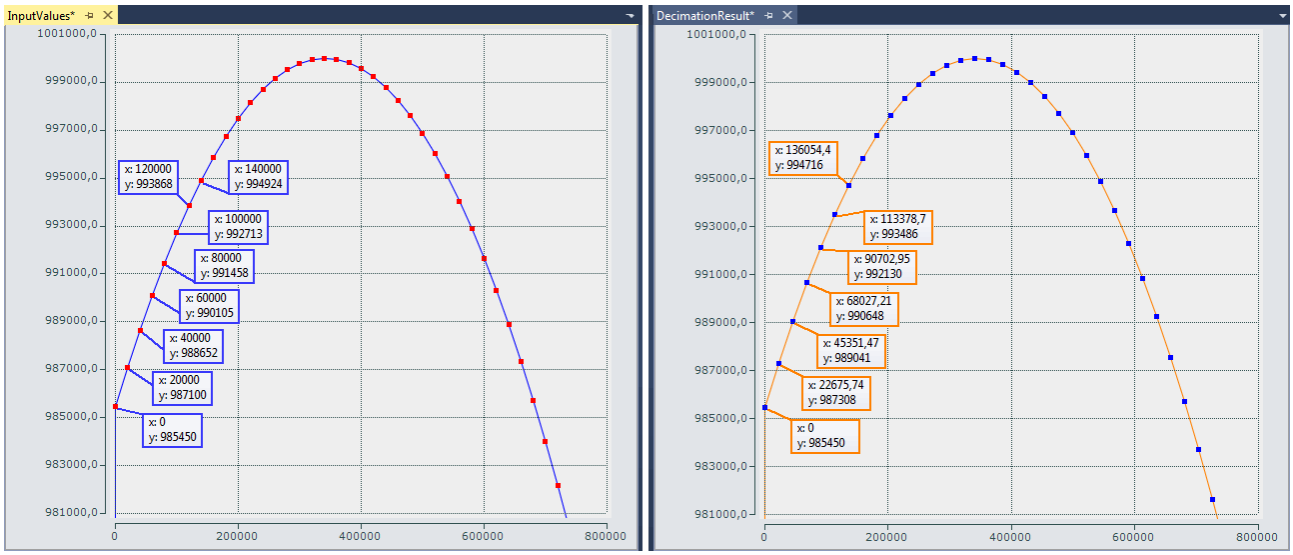


Fig. 33: Decimation from 20 µs (left) to 22.675.. µs (right) with ELM3602

The decimation factor is given by entering the value "50/44.1" for *nDecimationValue* in the sample. If this sample is used for the EL3751 with 500 µs cycle time and 5x oversampling, the sampling interval of 100 µs, which originates from the EL3751, is converted to approx. 113.378.. µs. This sample is designed accordingly.
The decimation in the program is freely selectable and must be configured with an oversampling factor and a task cycle time. The variable nOVS must contain the same oversampling factor as set in the process data configuration.

Download sample program 7:

- Configuration: IPC + EK1100 + **EL3751** + EL9011:
  https://infosys.beckhoff.com/content/1033/epp3504/Resources/5090848011.zip
- Configuration: IPC + EK1100 + **ELM3602-0002** + EL9011:
  https://infosys.beckhoff.com/content/1033/epp3504/Resources/5117137291.zip

Note: When using an EtherCAT box like the EPP35xx, the EtherCAT coupler EK1100 is omitted.

General information

The time at which the EtherCAT frames are passed to the box module is subject to fluctuations, referred to as EtherCAT frame jitter. If these fluctuations are large in relation to the cycle time, it is possible that data is fetched late from the box module, and dropouts/duplications may occur in the scope display. Such effects can be diagnosed with TwinCAT EtherCAT diagnostics. In the sample program for the ELM3602, the variable *nEqualTimeStampsCnt* is available for this kind of verification. The variable is incremented if such a failure occurs. It can be remedied by changing the DC ShiftTime of the box module; see the EtherCAT system documentation.

Declaration part:

```
// THIS CODE IS ONLY AN EXAMPLE - YOU HAVE TO CHECK APTITUDE FOR YOUR APPLICATION
PROGRAM MAIN
VAR CONSTANT
    // User decimation factor e.g. 50 to 44.1 kSps:
    nDecimationValue       :LREAL := 50/44.1; // 50/20;
```

```
    nOVS                    :BYTE := 5;         // Oversampling factor
    nTaskCycle_ns           :UDINT := 500000;  // PlcTask configured cycle time in ns

    nOVSTimeInterval_ns     :UDINT := LREAL_TO_UDINT(nTaskCycle_ns/nOVS); // OVS interval
    nDecTimeInterval_ns     :LREAL := nDecimationValue * nOVSTimeInterval_ns; // Decimation interval
END_VAR
VAR
   aSamples_1 AT%I*         :ARRAY[0..nOVS-1] OF DINT;     // Link to the terminal PDO
   aOVS_SampleSets          :ARRAY[0..(2*nOVS)-1] OF DINT; // 2 OVS sample sets

   nVarDecResult            :DINT;  // The calculated interpolated value
   tVarDecResult            :LREAL; // Decimation timestamp
   aVarDecResult            :ARRAY[0..nOVS-1] OF DINT; // Decimation result values
   aVarDecResult_TS         :ARRAY[0..nOVS-1] OF LREAL; // Decimation result timestamps

   nResultNoOfSamples       :BYTE; // This is for the user for further processing

   nDivVar                  :INT;  // Value for selection of the target input element
   tDecVar_InTaskCycle      :LREAL:=0; // Time span for all decimation timestamps within a task cycle
   i                        :BYTE:=0; // Common loop counter
   nDX                      :LREAL; // X-Difference: target input element to decimation element
   nDY                      :DINT;  // Y-Difference: two values for interpolation
   sVal                     :LREAL; // Slope for calculation of new value
   bEnable                  :BOOL:=FALSE; // Start/Stop conversion to decimation values
   nOVS_CycleCount          :ULINT := 0;  // Time value for every OVS sample
   // Values for testing
   bTEST_VALUES_ENABLED     :BOOL := FALSE; // No input value needed, if TRUE
   nPhi                     :LREAL := 1.4; // Start angle for sinus simulation
   // For visualization only:
   aOVS_Samples             :ARRAY[0..nOVS-1] OF DINT;  // 2 OVS sample sets (value)
   aOVS_Samples_TS          :ARRAY[0..nOVS-1] OF ULINT; // 2 OVS sample sets (timestamp)
END_VAR
```

Execution part:

```
// 500 µs Task
FOR i:= 0 TO nOVS-1 DO
   // Shift OVS set to left and update on right:
   aOVS_SampleSets[i] := aOVS_SampleSets[i+nOVS];   // Transfer "samples set" to the left side
   IF bTEST_VALUES_ENABLED THEN
      // Simulate values:
      aOVS_SampleSets[i+nOVS] := LREAL_TO_DINT(1000000 * SIN(nPhi));
      nPhi := nPhi + 0.01;//0.003141592653;
   ELSE
      // Fill current new samples set on right:
      aOVS_SampleSets[i+nOVS] := aSamples_1[i];
   END_IF
END_FOR
IF bEnable THEN
   nResultNoOfSamples := 0; // Use for further processing

   FOR i := 0 TO nOVS-1 DO
      nDivVar := TRUNC_INT(tDecVar_InTaskCycle/nOVSTimeInterval_ns);
      // Check, if new value is in grid
      IF (nDivVar = i) THEN
         nResultNoOfSamples := nResultNoOfSamples + 1;

         // Calc slope by the left and right element values (dy/dx):
         nDY := aOVS_SampleSets[i+1] - aOVS_SampleSets[i];
         sVal := DINT_TO_LREAL(nDY)/nOVSTimeInterval_ns;
         // Get the time (difference) from the left side element start to the desired time point:
         nDX := tDecVar_InTaskCycle
                - TRUNC_INT(tDecVar_InTaskCycle/nOVSTimeInterval_ns)
                * UDINT_TO_LREAL(nOVSTimeInterval_ns);
         // Calc timestamp
         tVarDecResult := nDX + ULINT_TO_LREAL(nOVS_CycleCount);
         // Calc new value:
         nVarDecResult :=
                  LREAL_TO_DINT(DINT_TO_LREAL(aOVS_SampleSets[i]) + sVal * nDX);

         // next decimation time step
         tDecVar_InTaskCycle := tDecVar_InTaskCycle + nDecTimeInterval_ns;
         tDecVar_InTaskCycle := tDecVar_InTaskCycle
                - INT_TO_UDINT(TRUNC_INT(tDecVar_InTaskCycle/nTaskCycle_ns))
                * nTaskCycle_ns;
      END_IF
      // Fill timestamp and new value allocated to the field element of its timestamp
      aVarDecResult_TS[i] := tVarDecResult;
      aVarDecResult[i] := nVarDecResult;
      // For visualization of the original input:
```

```
      aOVS_Samples[i] := aOVS_SampleSets[i];
      aOVS_Samples_TS[i] := nOVS_CycleCount;
      // Count the task cycle timestamp
      nOVS_CycleCount := nOVS_CycleCount + nOVSTimeInterval_ns;
   END_FOR
END_IF
IF nOVS_CycleCount = 1000000000 THEN
   bEnable := FALSE;// Stop after 1s just for recording
   IF NOT bEnable THEN
      bEnable := TRUE; // OVS-Samples transferred complete into both array sets
   END_IF
END_IF
```

Also see about this

- Features [▶ 000]

5.3.7 Sample program 8 (diagnosis messages)

Download TwinCAT 3 project: https://infosys.beckhoff.com/content/1033/epp3504/Resources/4279234443.zip

Note on loading the program: Preparation to start the sample program (tpzip file/ TwinCAT 3) [▶ 89]

Program description

This sample program reads several CoE Objects of the box module and yet 0x10F3 „Diagnosis History“ [▶ 69] that contains user specific diagnosis data:
Diagnosis message No.01...16 (0x10F3:06...0x10F3:15). Format of a message (consider little endian):

[dddd cccc ffff mmmm tttttttttttttttt pppp $_{(i)}$ **kk** $_{(i)}$ **]**

dddd = DiagCode: z.B. (00 E0): 0xE000 standard Beckhoff Message

cccc = ProductCode (21 50): 0x5021 = Code for ELM

ffff = Flags, amongst others indication of the number (i) of parameters (pppp kk) to be given.
E.g. (02 00) = 0x0002; bit 4 is set, when not in DC operation

mmmm = Message ID – respective text can be found here: basic principles of diag messages [▶ 226]

tttttttttttttttt = TimeStamp

pppp $_{(i)}$ = Datatype of the parameters, e.g. (05 00) = 0x0005 for datatype UINT8

kk $_{(i)}$ = parameter value

e.g. 2 x UINT8 parameters as indicated by ffff (Flags), with values 0x3C and 0x89
= “05003C050089“

The further procedure is described in section TwinCAT Quickstart, TwinCAT 3, Starting the controller [▶ 152].

5.3.8 Sample program 9 (measuring range combination)

Note on this chapter: The use of EL3751/ELM3xxx terminals also applies accordingly to EPP35xx.

In some applications it can be of interest to measure a value with very fine resolution in a small range, but still detect high deflections. If it is an AC/DC signal that has to be resolved around 0, the following approach can be used: Two inputs of an ELM terminal are electrically connected to simultaneously measure the signal, but with different measuring ranges.

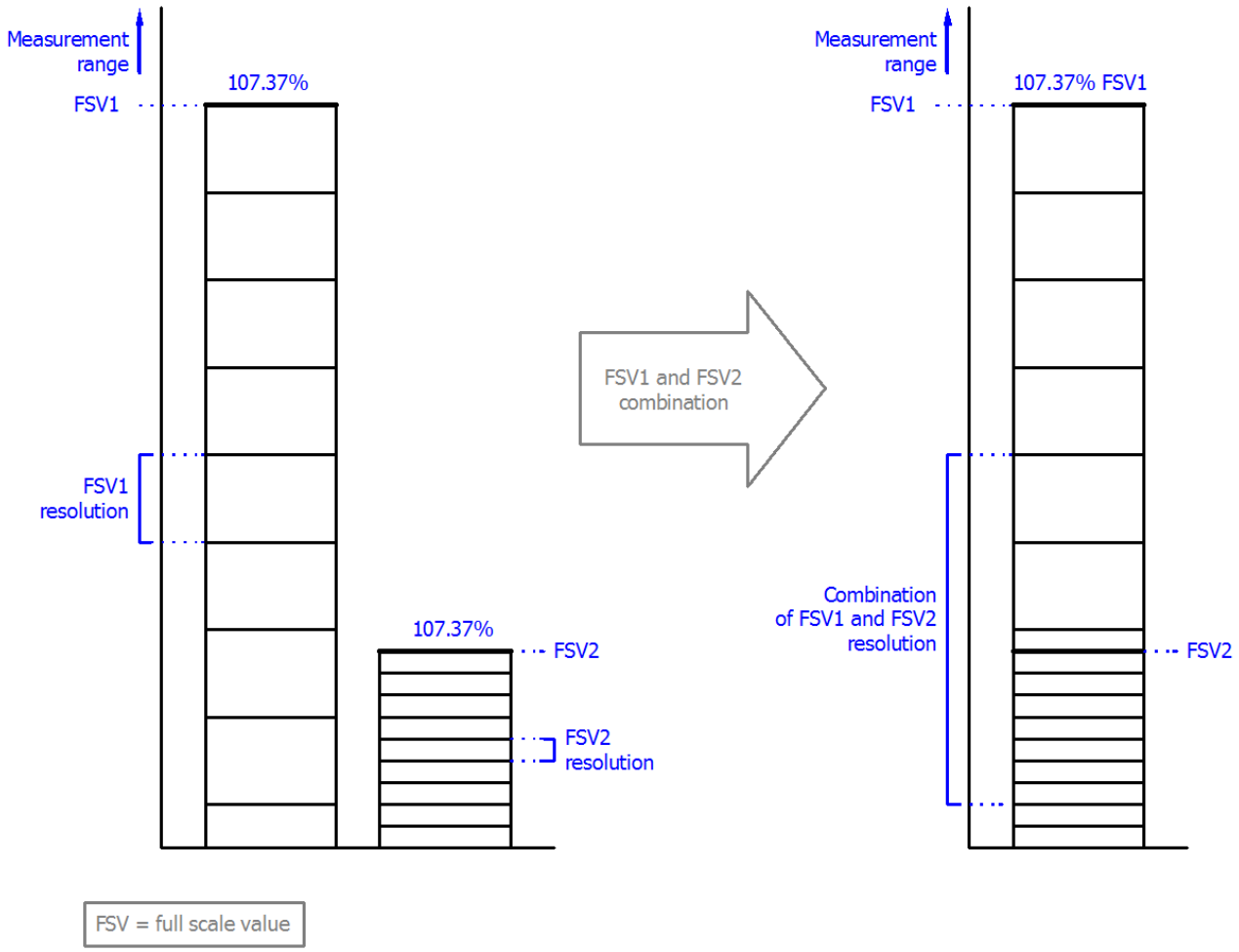


Fig. 34: Principle of combining two measuring channels with FSV1 and FSV2

The dynamic range of a typical 24-bit voltage or current measurement range with the absolute PDO end value of 2^{23} (bit 24 is sign) is $20 \cdot \log(2^{23}) \approx 138.5$ dB (without consideration of measurement uncertainties). Now it is possible to connect two (or more) inputs of a measuring system of the same measurement type with different measurement range end values (FSV1, FSV2, FSVn) in parallel to increase the dynamic range. The measured input value is then logged with two measuring ranges FSV1 *and* FSV2 through combination of two inputs. If FSV2 < FSV1 is selected and thus a lower resolution of FSV2 than FSV1, the low resolution of FSV2 is available if the magnitude of the measured input value is <= FSV2, and the measured input value can also be acquired for the larger range up to <= FSV1.

Note: The general definition is used to calculate the dynamic range:

Dynamic range = largest measured value / smallest unit

For output in dB accordingly with $20 \cdot \log(\text{FSV} / \text{Resolution}_{\text{FSV}})$. In this sample, using a combination of FSV1 and FSV2, the calculation is as follows:

Dynamic range = $20 \cdot \log(\text{FSV1} / \text{Resolution}_{\text{FSV2}})$.

The following sample program is based on a parallel connection of two input channels of the ELM3602-0002:

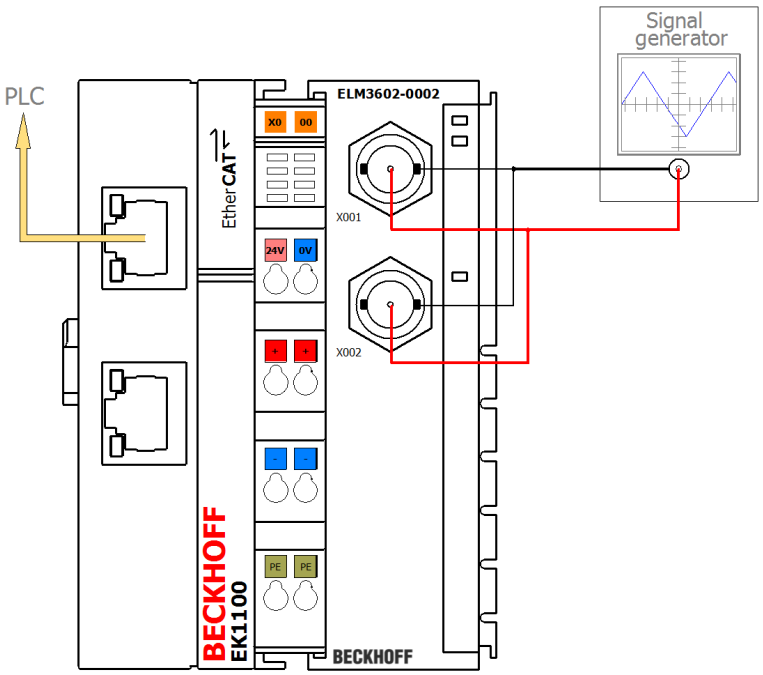


Fig. 35: Possible structure for the "Measurement range combination" sample program

Program description / function

The FSV1 of channel 1 is selected as ± 5 V, the FSV2 of channel 2 as ±80 mV. The program takes the measured input value from either channel 1 or channel 2 for a common variable depending on the magnitude of the unsigned amount of the measured input value: Initially, the limit value of 107% of the FSV2 (8388607) is verified.

In the CoE object directory, the following settings should be applied in the in the PAI settings objects, according to the variables *nFSV_HI* and *nFSV_LO*:

0x8000:01 → ±5 V

0x8010:01→ ±80 mV

Scaling for both channels: "Extended Range"; no filters active (corresponds to the default setting of the terminal).

Variables declaration:

```
PROGRAM MAIN
VAR CONSTANT
   nFSV_PDO              :  REAL := 7812500;
   nMAX_PDO              :  REAL := 8388607;

   nEXT_F                : REAL := nMAX_PDO/nFSV_PDO;
   nFSV_HI               :  REAL := 5;    // V
   nFSV_LO               :  REAL := 0.08; // V
   nStep_HI              :  REAL := nFSV_HI/nFSV_PDO;
   nStep_LO              :  REAL := nFSV_LO/nFSV_PDO;
END_VAR
VAR
   nSamplesIn1    AT%I*  :  DINT;
   nSamplesIn2    AT%I*  :  DINT;
   nValueCombi           :  LINT;
   nValueCombi_LREAL     :  LREAL;
   nKF                   :  REAL := nFSV_HI/nFSV_LO;
   nLimit                :  REAL := nMAX_PDO;
   nPDO1_REAL            :  LREAL;
   nPDO2_REAL            :  LREAL;
// Voltage values:
   nVoltage1             :  LREAL;
   nVoltage2             :  LREAL;
   nVoltageComb          :  LREAL;
END_VAR
```

Program code:

```
nPDO1_REAL := DINT_TO_LREAL(nSamplesIn1);
nPDO2_REAL := DINT_TO_LREAL(nSamplesIn2);
IF ABS(nPDO2_REAL) >= nLimit THEN
   nValueCombi_LREAL := nPDO1_REAL*nKF;
ELSE
   nValueCombi_LREAL := nPDO2_REAL;
END_IF
nValueCombi := LREAL_TO_LINT(nValueCombi_LREAL);
nVoltage1 := nPDO1_REAL * nFSV_HI/nFSV_PDO;
nVoltage2 := nPDO2_REAL * nFSV_LO/nFSV_PDO;
nVoltageComb := nValueCombi_LREAL * nFSV_LO/nFSV_PDO;
```

An application of this sample with a ±5 V FSV1 and a ±80 mV FSV2 and an input signal of ±5.68 V shows the voltage curve at input 1, input 2 and both combined inputs as a continuous range in the lowest recording. In the recording of input 2 the range of +/- overflow is marked (negative/ positive clipping):

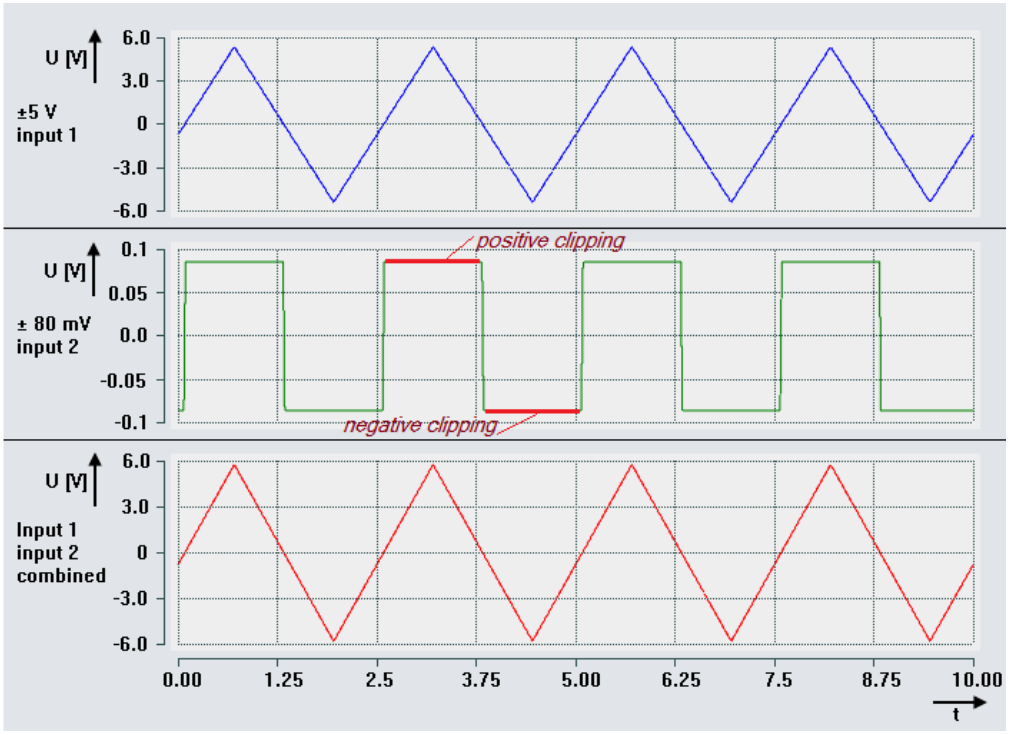


Fig. 36: Combination of two channels of the ELM3602-0002 with ±5 V and ±80 mV measuring range

With an applied delta voltage of approx. 86 mV ±5 mV, the transition range is indicated by the voltage characteristic of input 2 (values < 0 V):

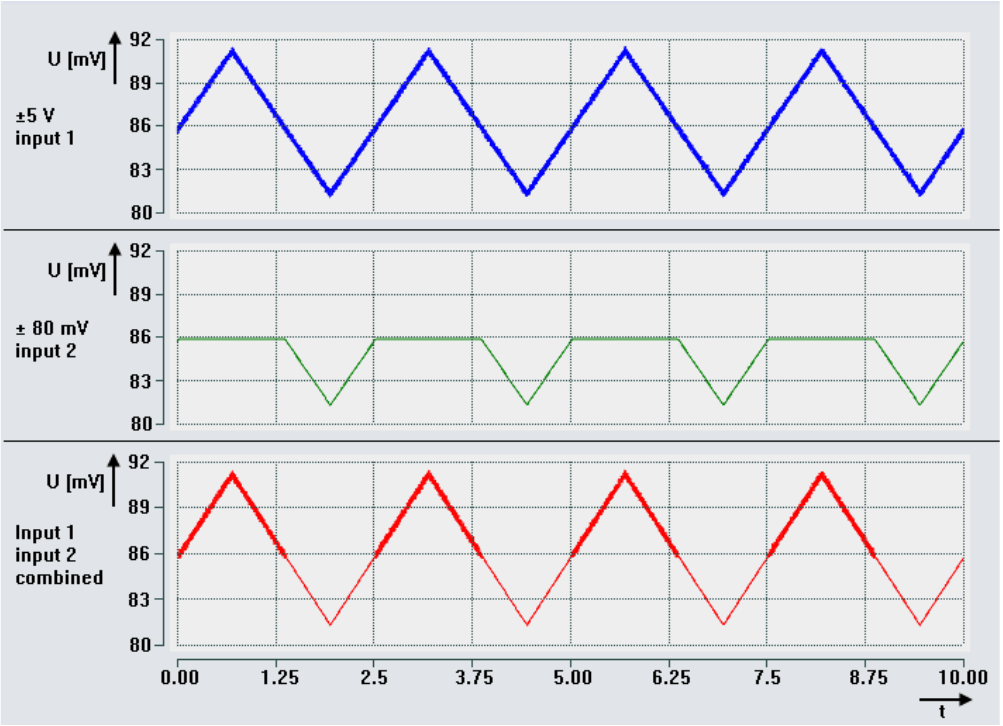


Fig. 37: Combination of two channels of the ELM3602-0002: Supply of a delta voltage in the positive transition range

The following applies to the (preset) extended range of both channels (without taking any measurement uncertainties into account):

If the dynamic range for the ± 5 V measuring range is approx. 20 · log (5.368 / 6.4E-7) ≈ 138.47 dB, the combination of two channels of the terminal can be used to increase the dynamic range to approx. 20 × log (5.368 / 1.024E-8) ≈ 174.39 dB (with the limitation of a coarser resolution in the range of approx. ± 85.9 mV to ± 5.37 V).

Please note that under these conditions the terminal always displays errors via the error LED and the error bit and outputs error messages to the TwinCAT environment due to regularly occurring overflow of a measuring channel.

5.3.9 Sample program 10 (FB for real time diagnosis)

The following function block can be used as a template for a real time diagnosis application of an EtherCAT box module analog input in TwinCAT PLC. It must be placed between the box module and the application and evaluates the diagnostic variables coming from the box module. The measurement values will be unchanged passed through.

The function block is written for the ELM3602-0002 with oversampling = 5 and should be understood as a functional example and must be adapted if necessary to

- other terminals or box modules if necessary other value data types
- other oversampling values

It can be extended with data-processing code or further particular diagnostics or assigned to a completely different type of a box module (analog output EL4xxx, Encoder EL5xxx, ...).

The function block between the box module and the PLC can be schematically illustrated as follows:

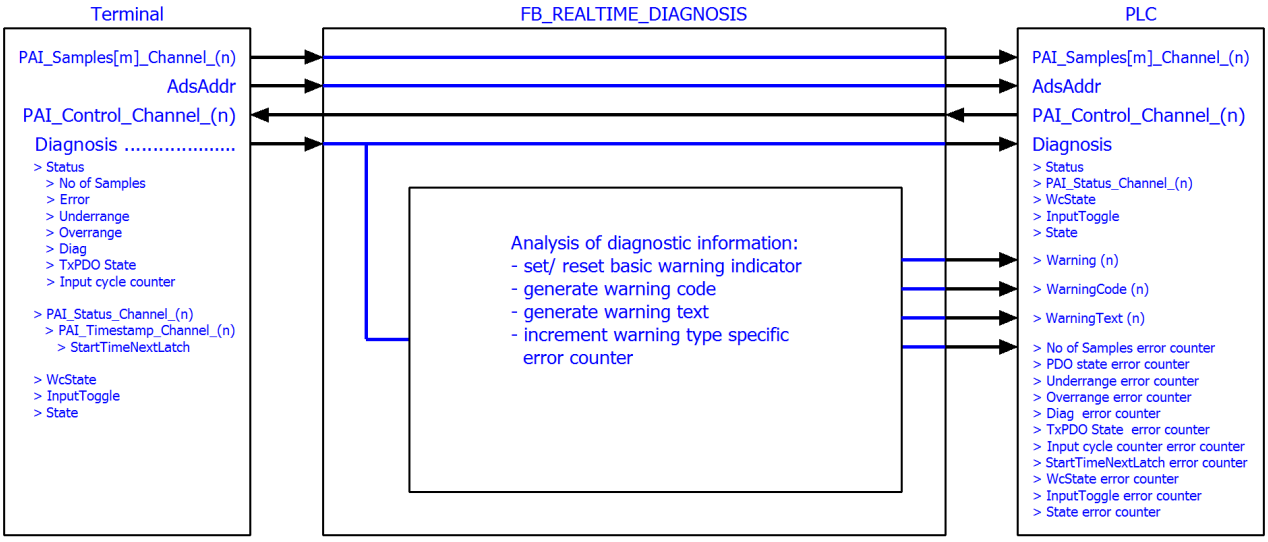
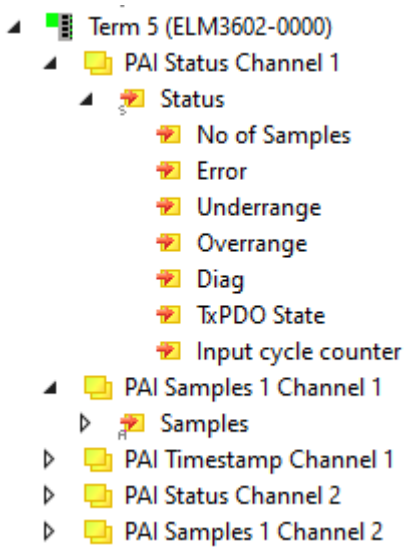


Fig. 38: Function block as an example for analysis of diagnostics information of the box module

Simplified linking via structure variable

This example program takes the opportunity to describe a TwinCAT function that simplifies the linking of complex PDO structures.

This function block would have to be linked to all real time variables of the box module: inputs and outputs; here e.g., for the ELM3602:



This time-consuming process can be simplified and accelerated by structuring in TwinCAT 3. Therefore, in this chapter two alternative variants in TwinCAT 3.1 are presented, as with a few clicks a structure can be defined in the PLC which corresponds to the **process image** of the box module.

The respective variant of the function block FB_REALTIME_DIAGNOSIS is included in the two example programs. It contains variables with an application-specific data type. This is a structure created by TwinCAT 3. Because the structure generated by TwinCAT directly maps the PDO structure of the box module, it is not necessary that a suitable structure must be elaborately created, or individual variables must be linked to individual data types. Only a link at a higher level (Status, Samples, Control, ...) is required.

This and all configurations are already included in the respective example program:

- Example program (variant A – using the “Plc” tab of the box module): https://infosys.beckhoff.com/content/1033/epp3504/Resources/7161530379.zip
- Example program (variant B – using of “Create SM/PDO Variables” by the advanced settings of the box module): https://infosys.beckhoff.com/content/1033/epp3504/Resources/7161533067.zip

Variant A, “Plc” tab:

In general, the generation of this special PDO data type is activated via the PLC settings of the box module (tab “Plc"): there the check box "Create PLC Data Type" is set ("Copy" then transfers this character string to the clipboard):

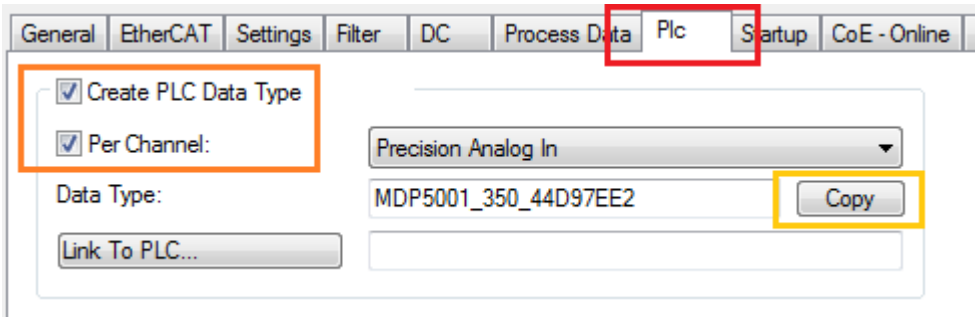


Fig. 39: Creation of PDO variables (TwinCAT version >= V3.1.4024.0)

The setting "Per Channel” can be set if not for all but for one only the structure shall be created.

The address assignments for inputs (%ATI*) and outputs (%ATQ*) are already within this generated structure. Inputs and outputs are therefore summarized in this structure.

The variables declaration within the function block FB_REALTIME_DIAGNOSIS then contains:

```
stELM3602Special                : MDP5001_350_EB559ACD;
```

Read access is provided to the inputs of the box module via the substructure *MDP5001_350_Input* and write access to the outputs via the substructure *MDP5001_350_Output* of the structure *stELM3602Special*.

Variant B, “Create SM/PDO variables“:

Commonly, the generation of this specific PDO data type incl. the PDO element will be activated via the EtherCAT settings of the box module: within the advanced settings under “General”/ "Behavior" the checkbox "Create SM/PDO Variables“ in “Process Data” is to set:

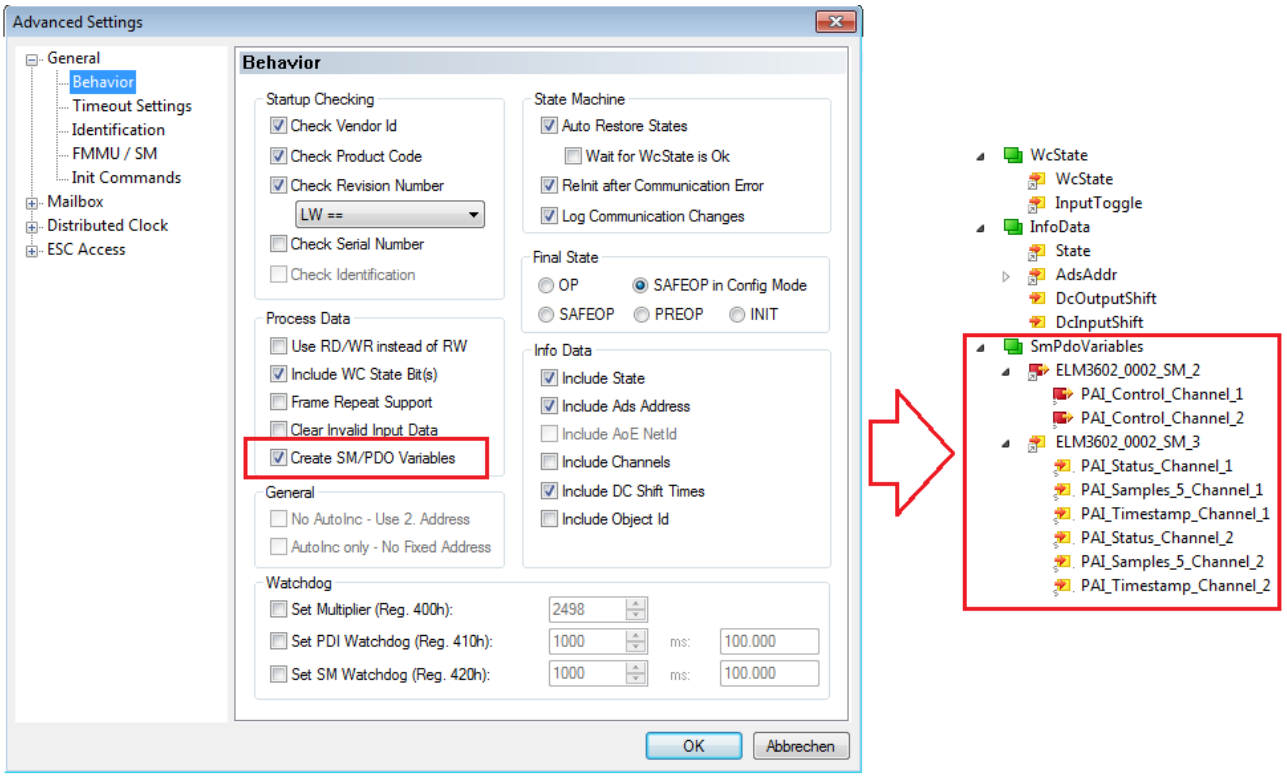


Fig. 40: Creation of the SmPdoVariables (TwinCAT version >= V3.1.4022.30)

The data type is visible by selecting the object and can be copied to the clipboard there:

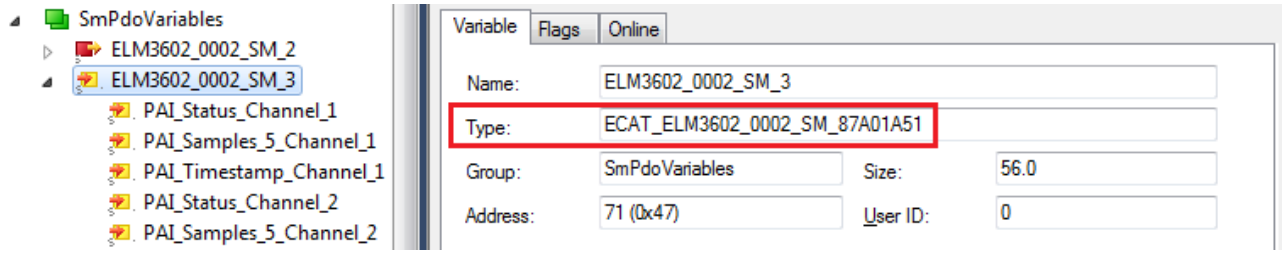


Fig. 41: Seek the generated data type of SmPdoVariables

The variables declaration within the function block FB_REALTIME_DIAGNOSIS then contains:

```
st_SM2              AT%Q*     : ECAT_ELM3602_0002_SM_3412CB6A;
st_SM3              AT%I*     : ECAT_ELM3602_0002_SM_87A01A51;
```

The read access to the inputs of the application is provided via the structure *st_SM3* and write access to the outputs via the structure *st_SM2*. These data structures corresponds to the automatically added new PDO element "SmPdoVariables".

5.3.10 Sample program 11 (scripts for generation and transformation of filter coefficients)

Download link:

https://infosys.beckhoff.com/content/1033/epp3504/Resources/12455432203.zip

For explanations of application see chapter "Exemplary calculation of IIR/FIR filter coefficients".

5.3.11 Sample program 12 (R/W signature of calibration)

The box module features an advanced calibration mechanism to store, among other things, an individual signature with 256 bytes, which results from the calibration data. In this way the customer could provide a calibration with a specific signature, e.g. to detect unauthorized internal manipulation of the calibration data; see also chapter “Display of data changes”.

The function block described below can be used as a basis for an implementation in TwinCAT on a PLC. To simplify matters, only a CRC16 was used in this sample to serve a "signature" limited to two bytes. At a commented point in the FB implementation, another signature algorithm can be implemented, which can be up to 256 bytes long.

The sample function block is included in the TwinCAT 3 archive, which is available for download together with a visualization:

https://infosys.beckhoff.com/content/1033/epp3504/Resources/8823639307.zip

Explanatory notes for the visualization "Calibration_Signature_RW"

The input variables of the ADS address and the "InputToggle" must be linked again if another terminal or Box (than ELM3602) is used for the sample. This must be entered in the field after starting the sample program. Alternatively, it can be entered before the start as initialization of the input variable "sTerminalTypeIn" of the function block "FB_VisuUpdate":

```
sTerminalTypeIn      : T_MaxString := 'ELM3602';
```

After the program start

The function block "FB_CalibrationSignature" is called in read mode by the visualization when channel +/- or interface +/- or "read" is actuated and in write mode when "write" is actuated. If, after reading, the calculated and the read signature match, *bCmpResult* becomes TRUE (no inequality). After a write access the entry remains in the read CoE and can be checked by reading (a write access does not change the state of *bCmpResult*).

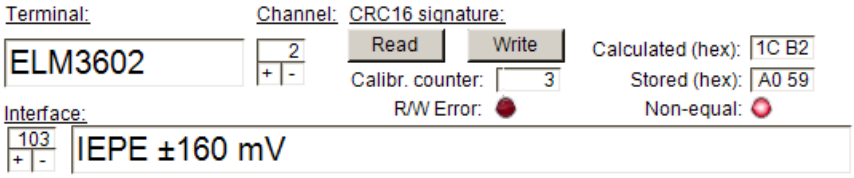


Fig. 42: Visualization of the sample implementation: Calibration signature

The variable *bError* (visualization representation: "R/W Error") provides information about a general error that has occurred when accessing the box module as well as the failure to find stored information of the box module (either the entry in the GVL is missing or the box module is not present).

Explanatory notes for FB_CalibrationSignature

The interface of the function block is structured as follows:

```
VAR_INPUT
   bInitialize                : BOOL := FALSE; // TRUE = Initialised
   bEnable                    : BOOL := FALSE; // Activate module
   tAmsNetIdArr               : AMSADDR;       // Ads address of the terminal/ box
   nIfSlectCoE                : WORD;          // Interface number of the CoE
   nChSelectCoE               : WORD := 1;     // Channel number
   eOption                    : E_CALSIG_OPTIONS; // Access get/set (read/write)
   stCoEPAIInfoDataCalCnt     : ST_CoE;        // Cal. counter object (EPP3504-0023/ ERP3504-0022)
END_VAR
VAR_OUTPUT
   bDone                      : BOOL; // Process end
   bCmpResult                 : BOOL; // Signature comparision: TRUE = equal
   nInterfaceUserCalCnt       : WORD; // Value of calibration counter
   bError                     : BOOL; // Case of error
   bCancel                    : BOOL; // Cancel
   nErrorId                   : UDINT;// Error number (all sources)
   anSigDataOutCoE            : ARRAY[0..(GVL_CoE.nSigLen-1)] OF BYTE; // Signature stored
   anSigDataOutCalc           : ARRAY[0..(GVL_CoE.nSigLen-1)] OF BYTE; // Signature calculated
END_VAR
```

For initialization the "Net Id" and "Port No." must be transferred to the variable "tAmsNetIdArr" of the FB instance. In addition, the CoE object for reading the calibration counter must be transferred via 'stCoEPAIInfoDataCalCnt', since this is different for the EPP3504-0023/ ERP3504-0022 box module.

A call is made with "bEnable := TRUE" for activation and with specification of the interface number (nIfSlectCoE) that applies to the box module to be addressed, the channel (nChSelectCoE) and for reading the stored signature "eOption := E_CALSIG_OPTIONS.get" or for writing "eOption := E_CALSIG_OPTIONS.set".

Then the function block is called until the output variable "bDone" is TRUE.

The outputs anSigDataOutCalc, anSigDataOutCoE, nInterfaceUserCalCnt and bCmpResult will provide content according to the selected option and the calculated/stored data of the box module.

To attempt to clear an error that has occurred in the case of "bError" = TRUE, the FB can be called with "bInit := FALSE" (e.g. if the channel number or the interface number has been corrected according to the addressed box module). The "nErrorId" can be used for evaluation.

In the function block, the signature calculation can be changed/extended at the following point:

```
// Calculate signature
// ============== User code here ==============
// Example: simple CRC:
nCrc := nIfSlectCoE + nChSelectCoE; // Default setting of start value
nCrc := F_DATA_TO_CRC16_CCITT(ADR(aData), nDataLen, nCrc); // Calculate "signature"
memset(ADR(anSigDataOutCalc), 16#FF, GVL_CoE.nSigLen);
memcpy(ADR(anSigDataOutCalc), ADR(nCrc), 2); // <- Dependant be the type of encryption
// ============================================
```

5.3.12 Sample program 13: Reading the BIC from the CoE

The Beckhoff Identification Code (BIC) is used for the unambiguous visual and electronic identification of Beckhoff products and was introduced continually into the ongoing device production from 2020 onwards. Refer also to the introductions in the chapter "Version identification of EtherCAT devices"/"Beckhoff Identification Code (BIC)" [▶ 11] Beckhoff Identification Code (BIC) [▶ 11]. The BIC contains several components, in particular the unambiguous BTN.

The BIC is also stored electronically in the ESI EPROM in all Beckhoff EtherCAT devices and can be read there by the EtherCAT Master (e.g. TwinCAT). A reading function is available for this in the TC3 EtherCAT lib from 2020 onwards.

Some of these EtherCAT devices are so-called intelligent slaves with a local microcontroller, which offers a so-called CoE directory for parameterization. The BIC can be mapped there by the firmware in index 0x10E2 for reading. In the course of continuous product maintenance, this function is gradually being introduced into the EtherCAT devices.

The function block described here, for example, reads all entries from the CoE object 0x10E2 and copies them into a structure variable field "*patManFactSpecIdCode*" provided for the purpose. In the first step, the function block reads the number of stored BIC entries and in the second step the individual data sets.

Note about the modular devices: if several sub-devices with so-called sub-BICs to be identified are installed in an EtherCAT device, 0x10E2:1 bears the BIC of the main device, while the BICs of the sub-devices are located in the subsequent indices.

Function block FB_GET_BIC

This function block is intended for an environment of a TwinCAT 3 project; i.e. a project must first be present or created. See further explanations within chapter "Sample programs"/" Preparation to start the sample program (tpzip file/ TwinCAT 3)" [▶ 89].

The declarations of the function block are as follows:

Inputs:

```
    userNetId            : T_AmsNetId; // NetId of EtherCAT device to be read
    userSlaveAddr        : UINT;   // Address of EtherCAT device to be read
    bExecute             : BOOL;   // Execute fb by rising edge
```

Outputs:

```
    bDone                : BOOL;   // TRUE = FB Execution done
   nNumOfSubIndizies     : BYTE;   // Number of read BICs via Sub-Ids
   // Array with struct of BIC entries:
    patManFactSpecIdCode :
      ARRAY[0..nMAXINDEXSUBINDIZES] OF POINTER TO T_MAN_FACT_SPEC_ID_CODE; // BICs data
    bError               : BOOL;
```

The function block is executed with a rising edge at the input "*bExecute*". The EtherCAT address "*userSlaveAddr*" of the box module and the "*userNetId*" are to be transferred. Successful execution is indicated by "*bDone*" (TRUE = successfully executed). After successful execution, the entries from the CoE object 0x10E2 exist in the pointer field "*patManFactSpecIdCode*". The number of read entries is given by "*nNumOfSubIndizies*". The following figure shows a filled structure with test data:

patManFactSpecIdCode	ARRAY [0..nMAXINDEXSUBINDIZES] OF POINTER ...	
patManFactSpecIdCode[0]	POINTER TO T_MAN_FACT_SPEC_ID_CODE	16#FFFF9F80B722D588
patManFactSpecIdCode[0]^	ARRAY [0..nMAXINDEXSUBINDIZES] OF POINTER ...	
Article_number	STRING(FB_GET_BIC.aLengthOfDataEntry[0])	'1P367926'
BTN	STRING(FB_GET_BIC.aLengthOfDataEntry[1])	'SBTNi9g102f6'
Article_description	STRING(FB_GET_BIC.aLengthOfDataEntry[2])	'1KELM3002_0000_____'
Quantity	STRING(FB_GET_BIC.aLengthOfDataEntry[3])	'Q5_trs'
Batch_number	STRING(FB_GET_BIC.aLengthOfDataEntry[4])	'2P142019365930'
ID_serial_number	STRING(FB_GET_BIC.aLengthOfDataEntry[5])	'515859384972'
Variant_number	STRING(FB_GET_BIC.aLengthOfDataEntry[6])	'30PV322, 3*A749_PHIYUCEGVKCOIJWW'

Fig. 43: Test data to illustrate the content of *patManFactSpecIdCode[0]*^ of the FB

The data type needed for FB_GET_BIC is:

```
TYPE T_MAN_FACT_SPEC_ID_CODE:
STRUCT
      Article_number       :STRING(FB_GET_BIC.aLengthOfDataEntry[0]);
      BTN                  :STRING(FB_GET_BIC.aLengthOfDataEntry[1]);
      Article_description  :STRING(FB_GET_BIC.aLengthOfDataEntry[2]);
      Quantity             :STRING(FB_GET_BIC.aLengthOfDataEntry[3]);
      Batch_number         :STRING(FB_GET_BIC.aLengthOfDataEntry[4]);
      ID_serial_number     :STRING(FB_GET_BIC.aLengthOfDataEntry[5]);
      Variant_number       :STRING(FB_GET_BIC.aLengthOfDataEntry[6]);
END_STRUCT
END_TYPE
```

With the stored data lengths within FB_GET_BIC:

```
aLengthOfDataEntry    : ARRAY[0..nNumOfDataIds] OF BYTE := [8, 12, 32, 6, 14, 12, 32];
```

The function block is available as a .tpzip file in the following download (as a .zip file) and also contains the necessary library references (Tc2_EtherCAT, Tc3_DynamicMemory), the necessary data structure and a call in MAIN:
https://infosys.beckhoff.com/content/1033/epp3504/Resources/9880941579.zip

6 Features

NOTICE

This short documentation does not contain any further information within this chapter. For the complete documentation please contact the Beckhoff sales department responsible for you.

7 Commissioning on EtherCAT Master

7.1 General Commissioning Instructions for an EtherCAT Slave

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

Diagnosis in real time: WorkingCounter, EtherCAT State and Status

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

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

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

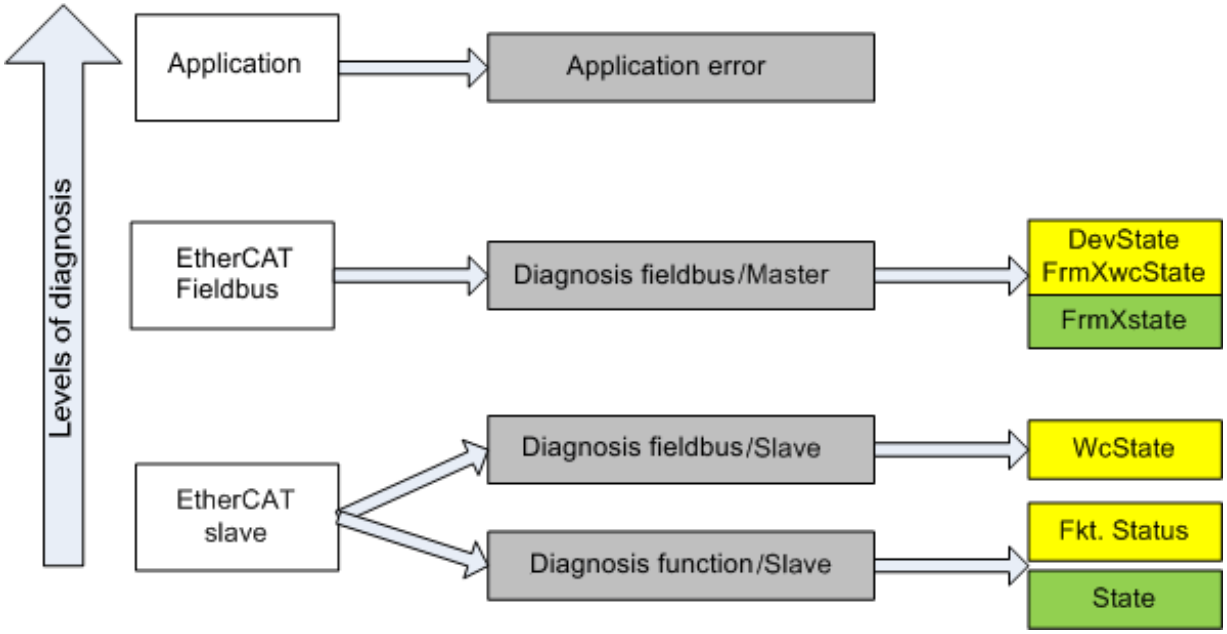


Fig. 44: Selection of the diagnostic information of an EtherCAT Slave

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

Colour	Meaning
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.

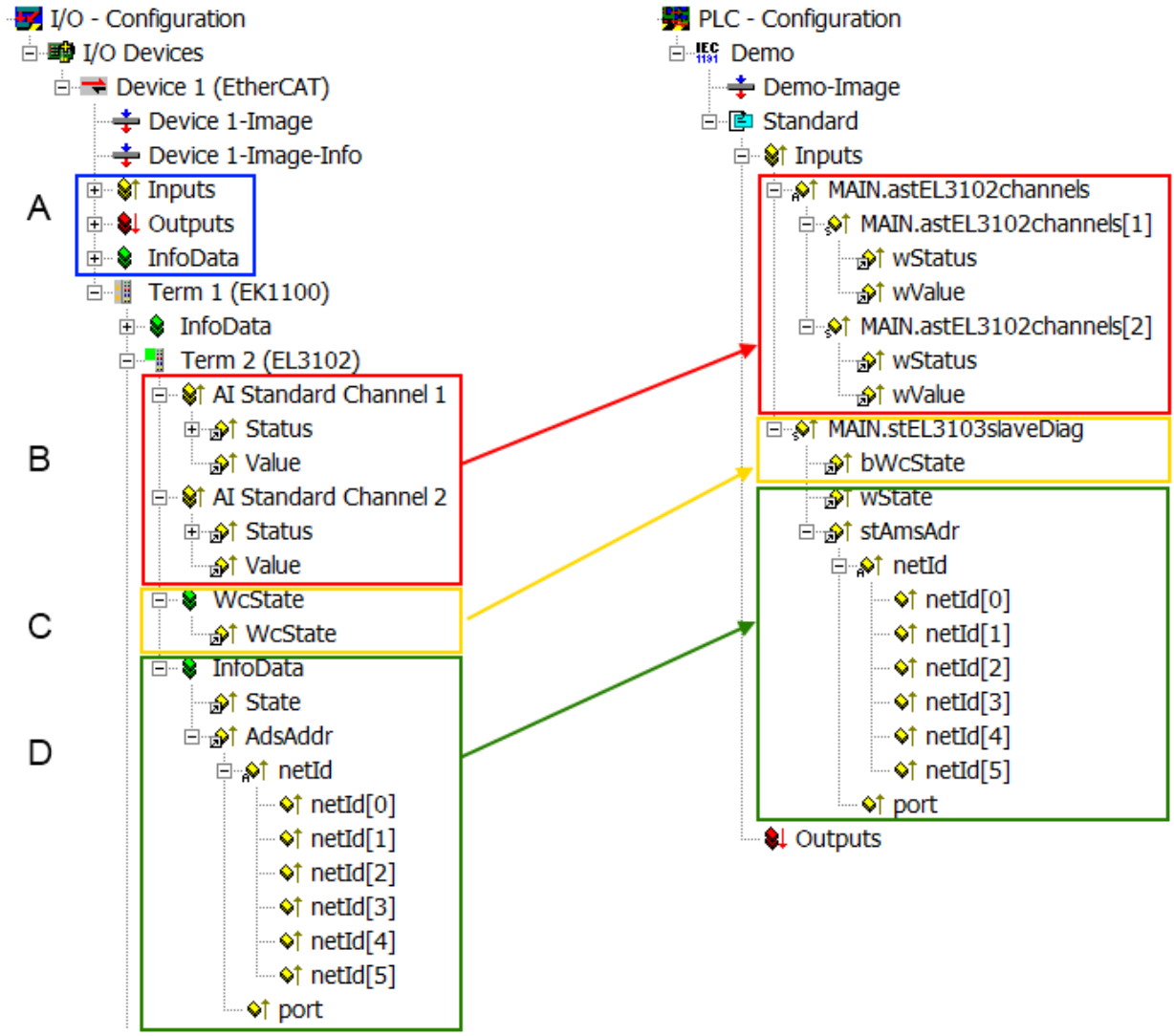


Fig. 45: Basic EtherCAT Slave Diagnosis in the PLC

The following aspects are covered here:

Code	Function	Implementation	Application/evaluation
A	The EtherCAT Master's diagnostic information updated cyclically (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

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

NOTICE

Diagnostic information

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

CoE Parameter Directory

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

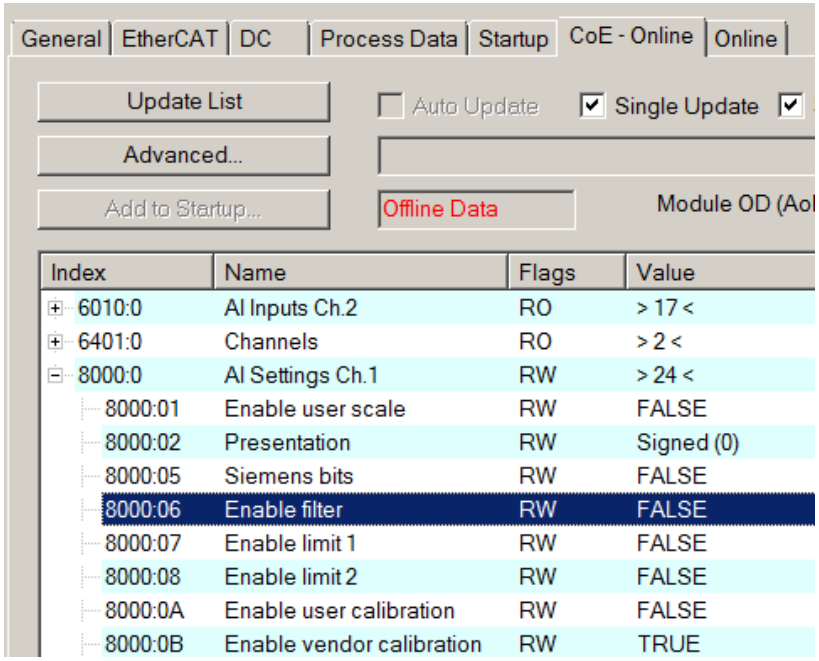


Fig. 46: EL3102, CoE directory

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.

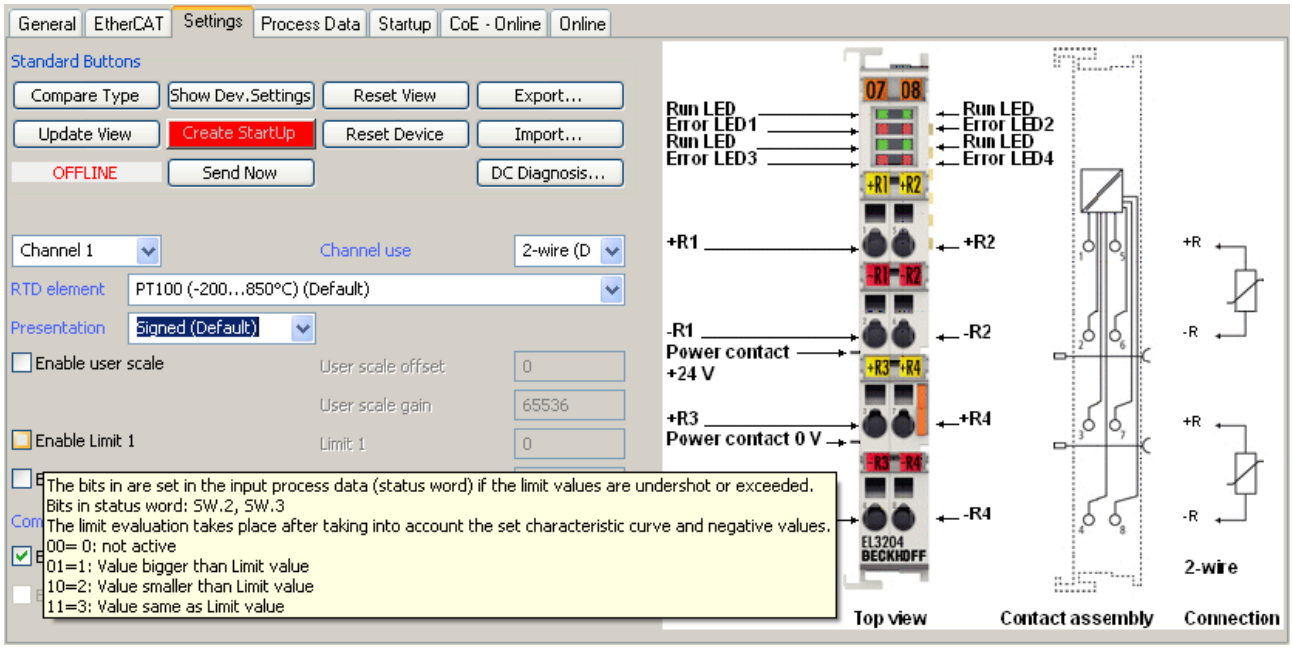


Fig. 47: Example of commissioning aid for a EL3204

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

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

The target state wanted by the user, and which is brought about automatically at start-up by TwinCAT, can be set in the System Manager. As soon as TwinCAT reaches the status RUN, the TwinCAT EtherCAT Master will approach the target states.

Standard setting

The advanced settings of the EtherCAT Master are set as standard:

- EtherCAT Master: OP
- Slaves: OP
  This setting applies equally to all Slaves.

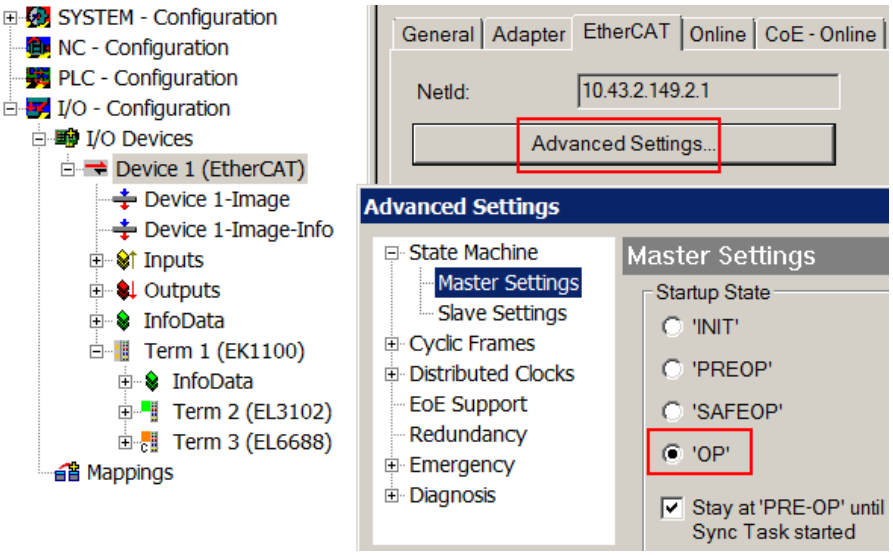


Fig. 48: Default behaviour of the System Manager

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

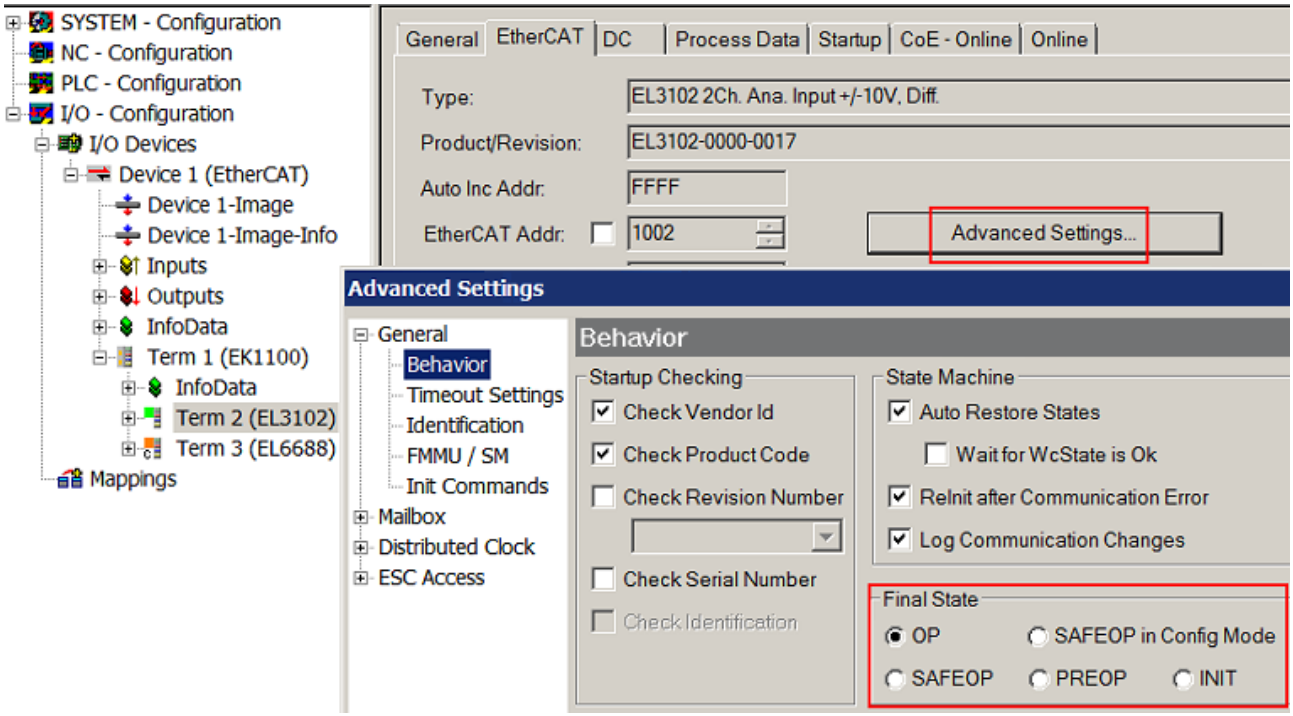


Fig. 49: Default target state in the Slave

Manual Control

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

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

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

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

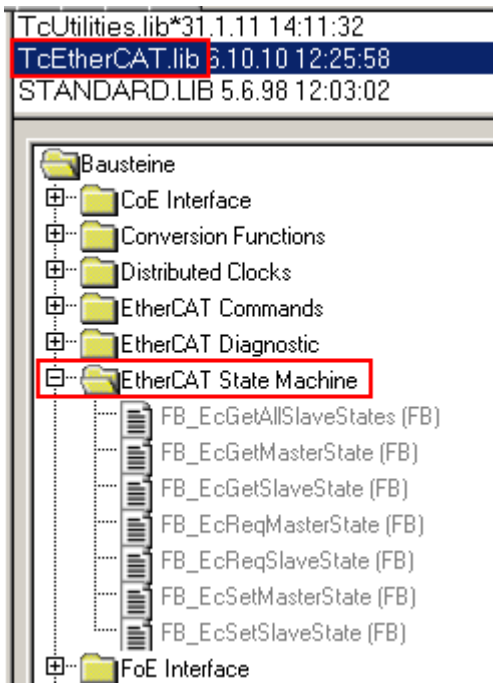


Fig. 50: PLC function blocks

Note regarding E-Bus current

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

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

General | Adapter | EtherCAT | Online | CoE - Online

NetId: 10.43.2.149.2.1 Advanced Settings...

Number	Box Name	Address	Type	In Size	Out S...	E-Bus (..
1	Term 1 (EK1100)	1001	EK1100			
2	Term 2 (EL3102)	1002	EL3102	8.0		1830
3	Term 4 (EL2004)	1003	EL2004		0.4	1730
4	Term 5 (EL2004)	1004	EL2004		0.4	1630
5	Term 6 (EL7031)	1005	EL7031	8.0	8.0	1510
6	Term 7 (EL2808)	1006	EL2808		1.0	1400
7	Term 8 (EL3602)	1007	EL3602	12.0		1210
8	Term 9 (EL3602)	1008	EL3602	12.0		1020
9	Term 10 (EL3602)	1009	EL3602	12.0		830
10	Term 11 (EL3602)	1010	EL3602	12.0		640
11	Term 12 (EL3602)	1011	EL3602	12.0		450
12	Term 13 (EL3602)	1012	EL3602	12.0		260
13	Term 14 (EL3602)	1013	EL3602	12.0		70
14	Term 3 (EL6688)	1014	EL6688	22.0		-240 !

Fig. 51: Illegally exceeding the E-Bus current

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

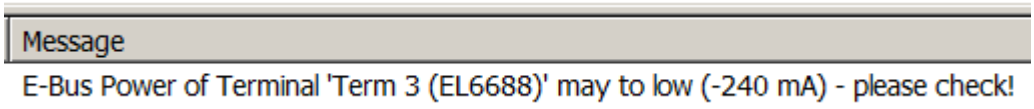


Fig. 52: Warning message for exceeding E-Bus current

NOTICE

Caution! Malfunction possible!

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

7.2 TwinCAT Quick Start

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

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

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

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

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

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

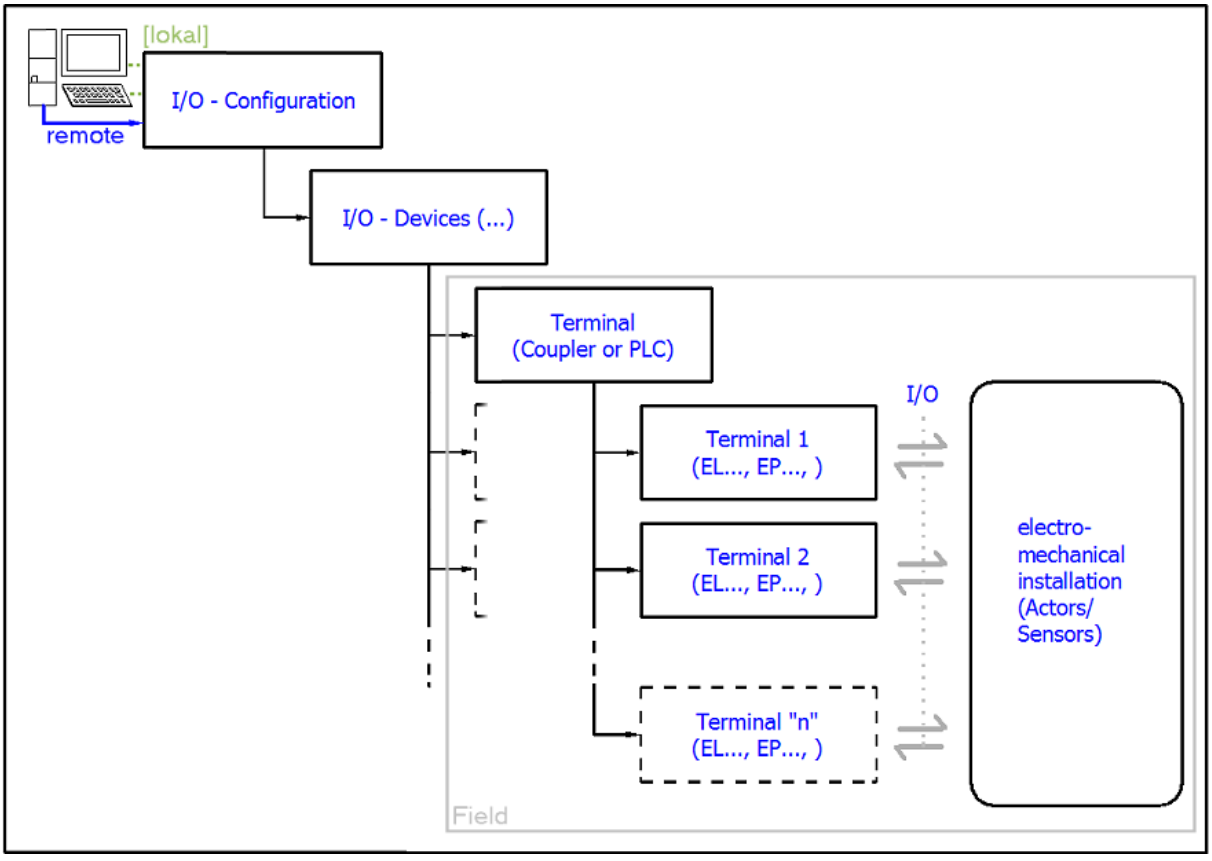


Fig. 53: Relationship between user side (commissioning) and installation

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

Example configuration (actual configuration)

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

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

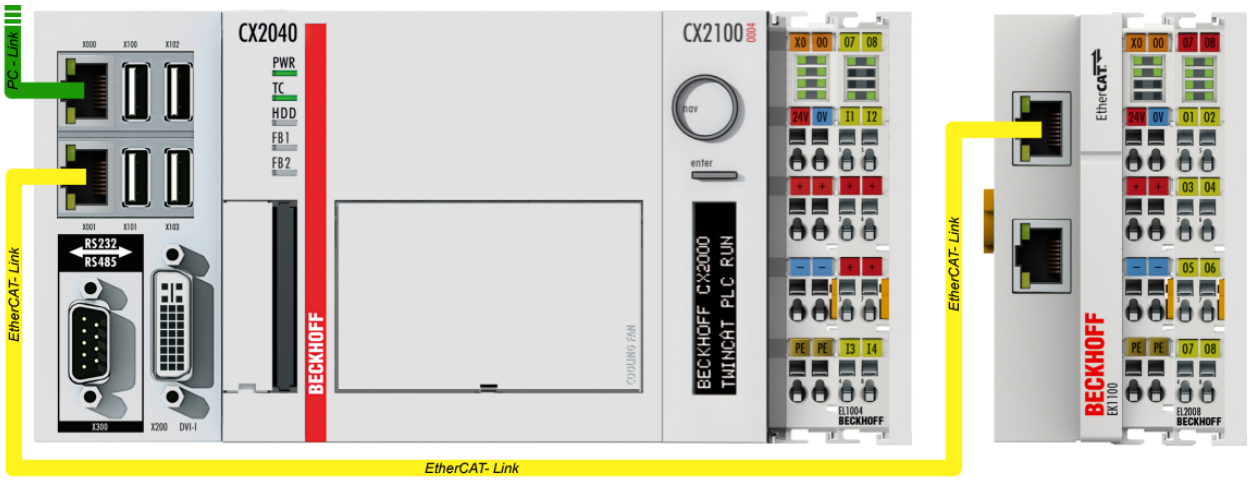


Fig. 54: Control configuration with Embedded PC, input (EL1004) and output (EL2008)

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.

7.2.1 TwinCAT 2

Startup

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

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

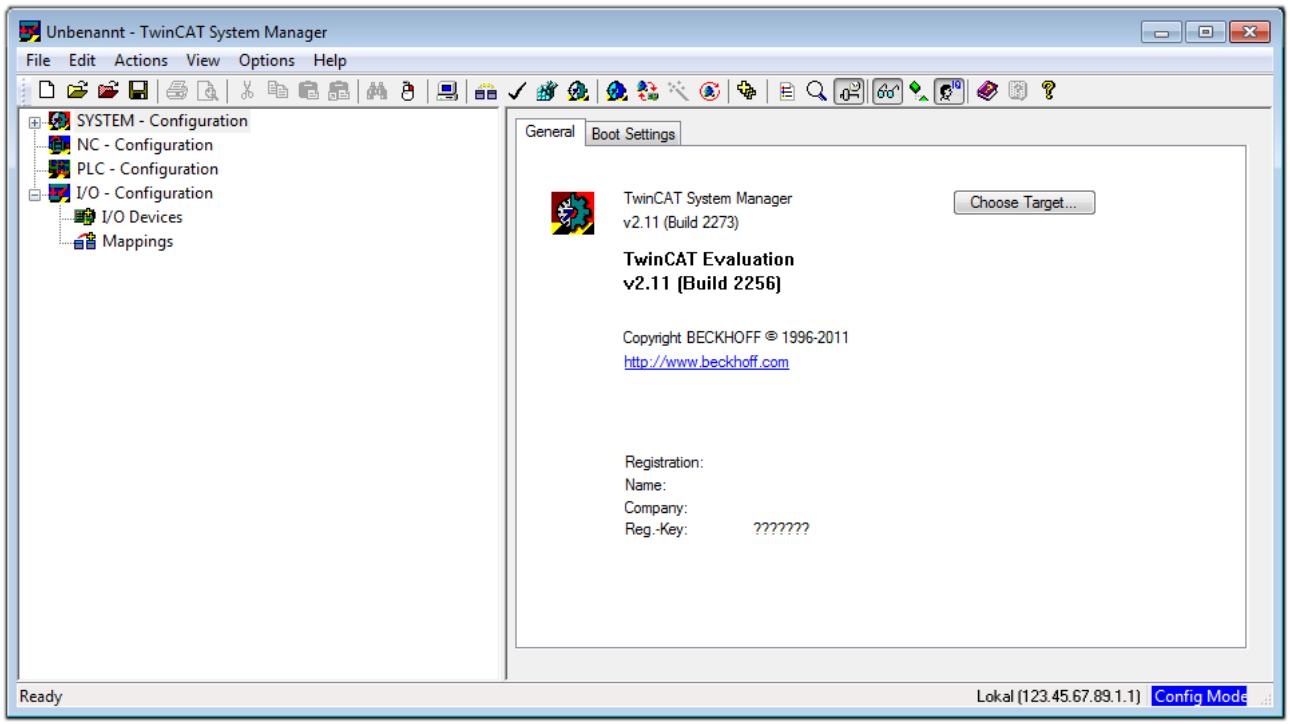


Fig. 55: Initial TwinCAT 2 user interface

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system, including the user interface (standard) is installed on the respective PLC, TwinCAT can be used in local mode and thus the next step is “Insert Device [▶ 131]”.

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

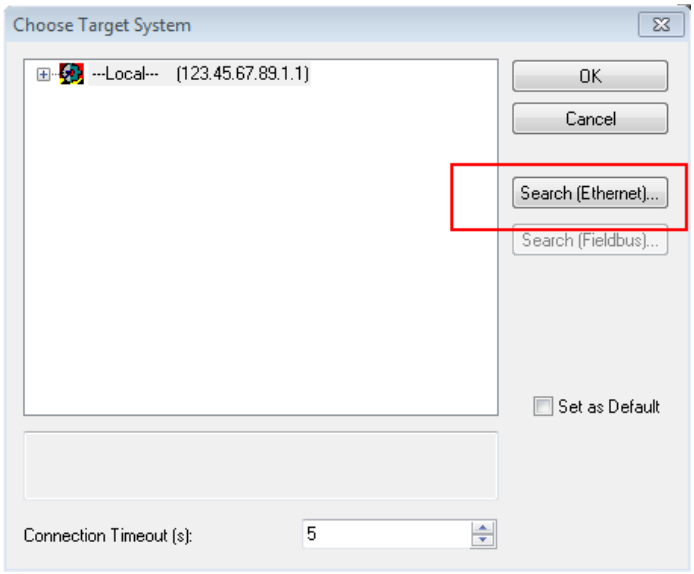


Fig. 56: Selection of the target system

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

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

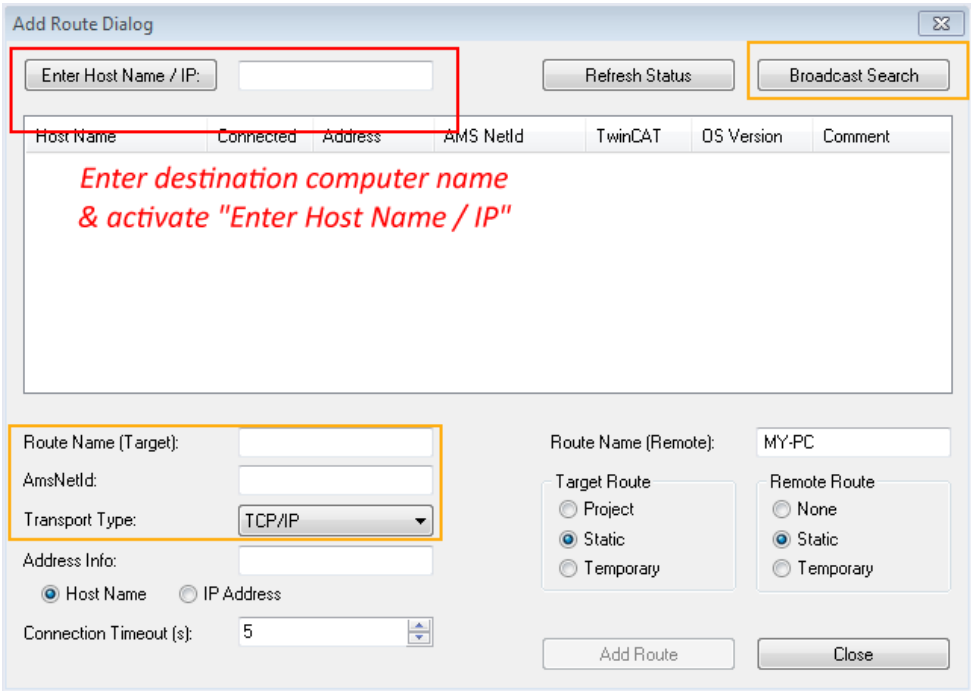
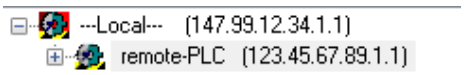


Fig. 57: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

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

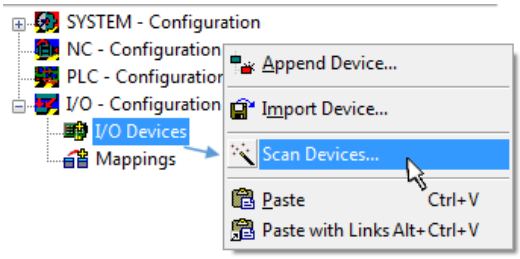


Fig. 58: Select “Scan Devices...”

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

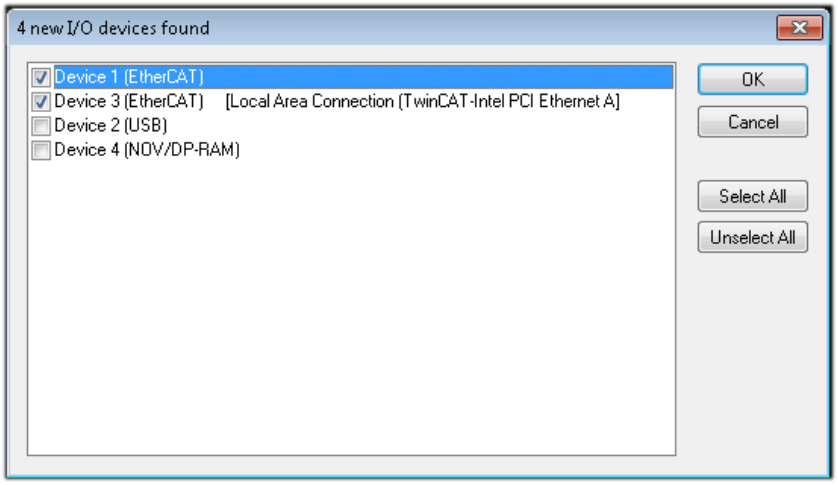


Fig. 59: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the example configuration [▶ 128] described at the beginning of this section, the result is as follows:

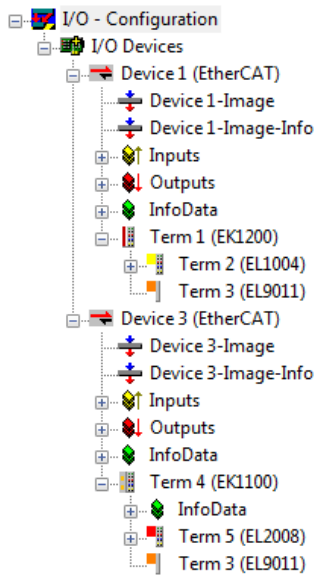


Fig. 60: Mapping of the configuration in the TwinCAT 2 System Manager

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

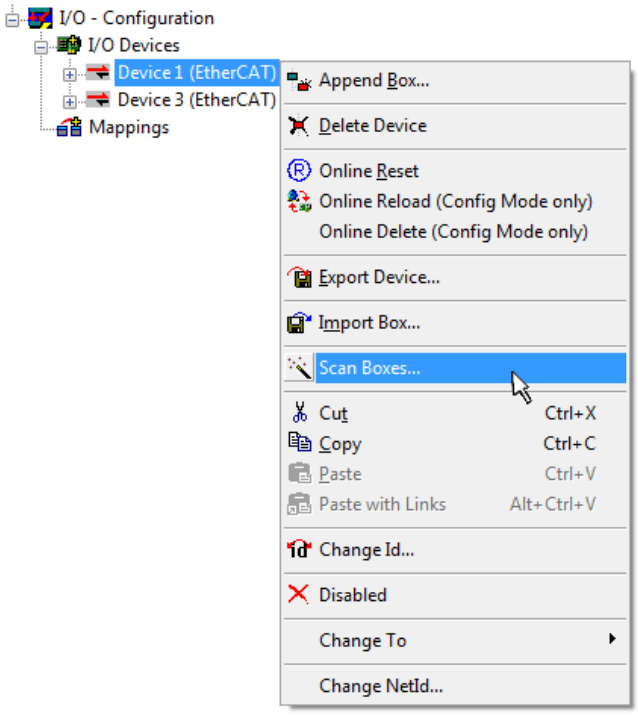


Fig. 61: Reading of individual terminals connected to a device

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

Programming and integrating the PLC

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

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

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

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

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

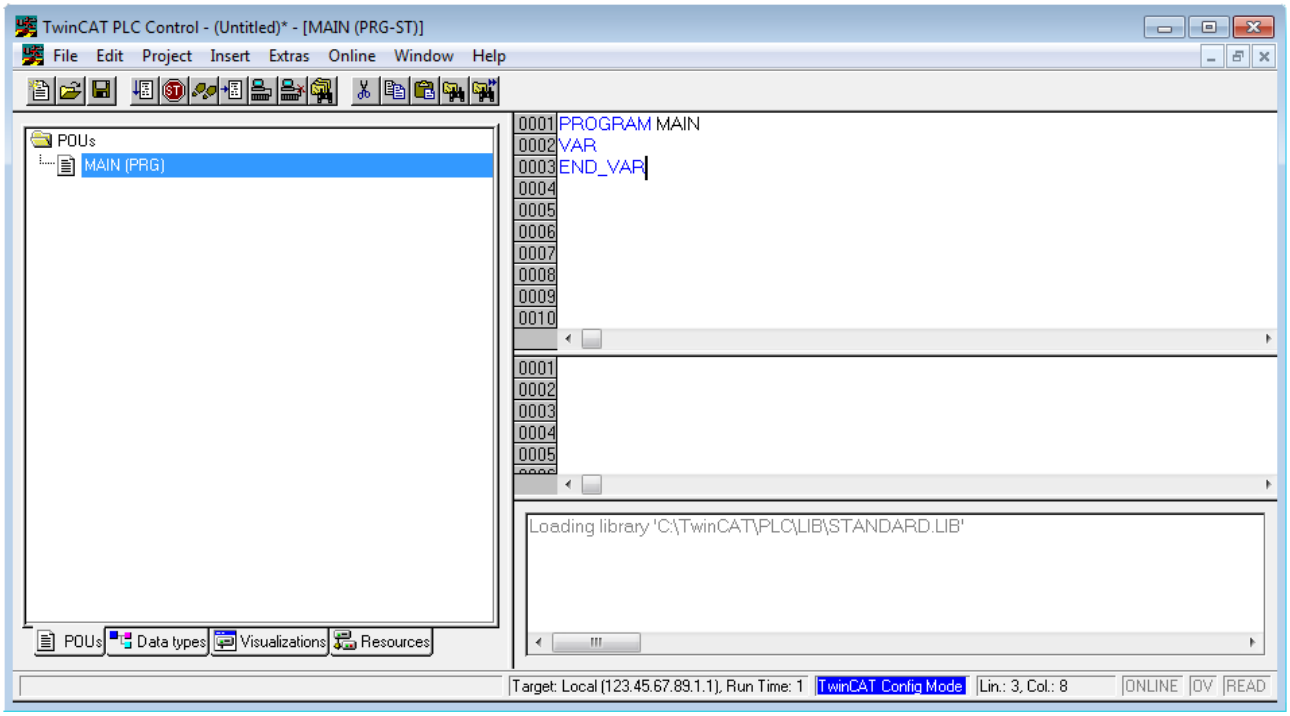


Fig. 62: TwinCAT PLC Control after startup

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

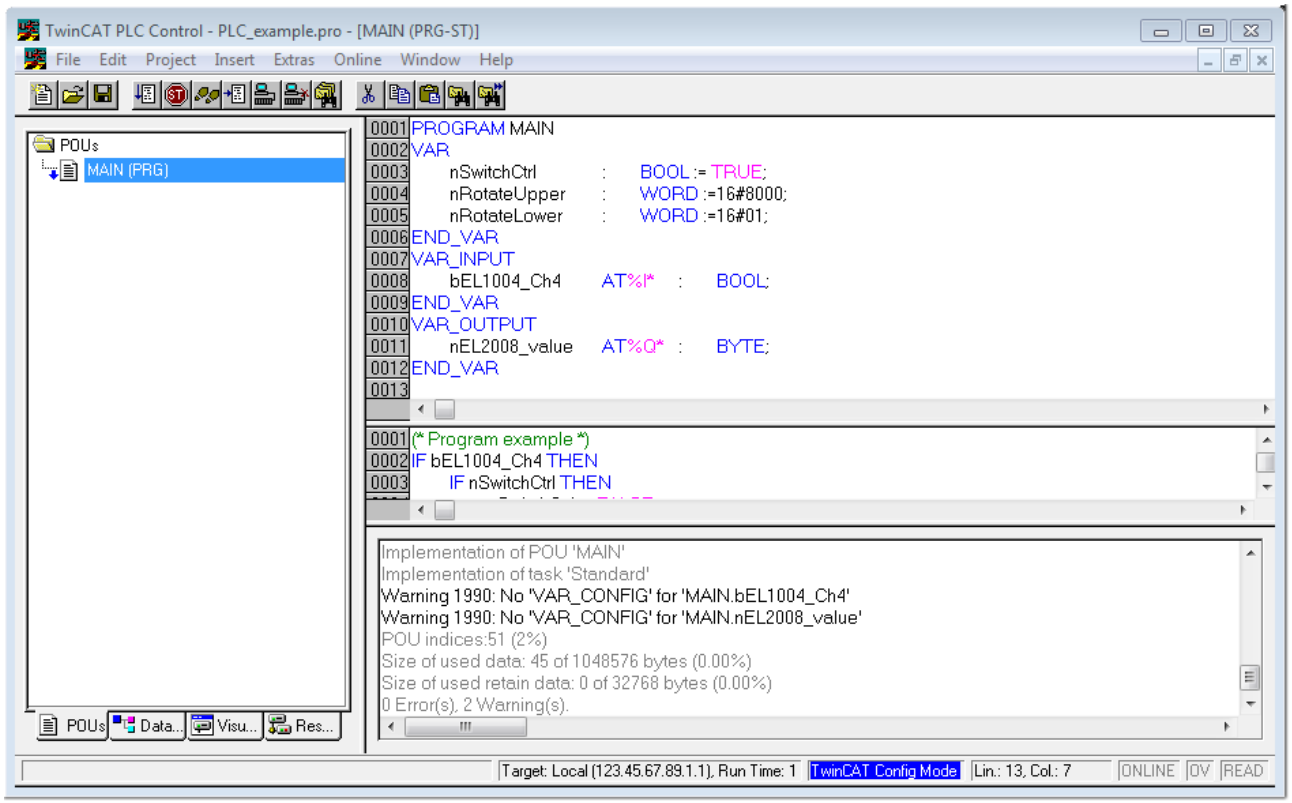


Fig. 63: Example program with variables after a compile process (without variable integration)

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

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

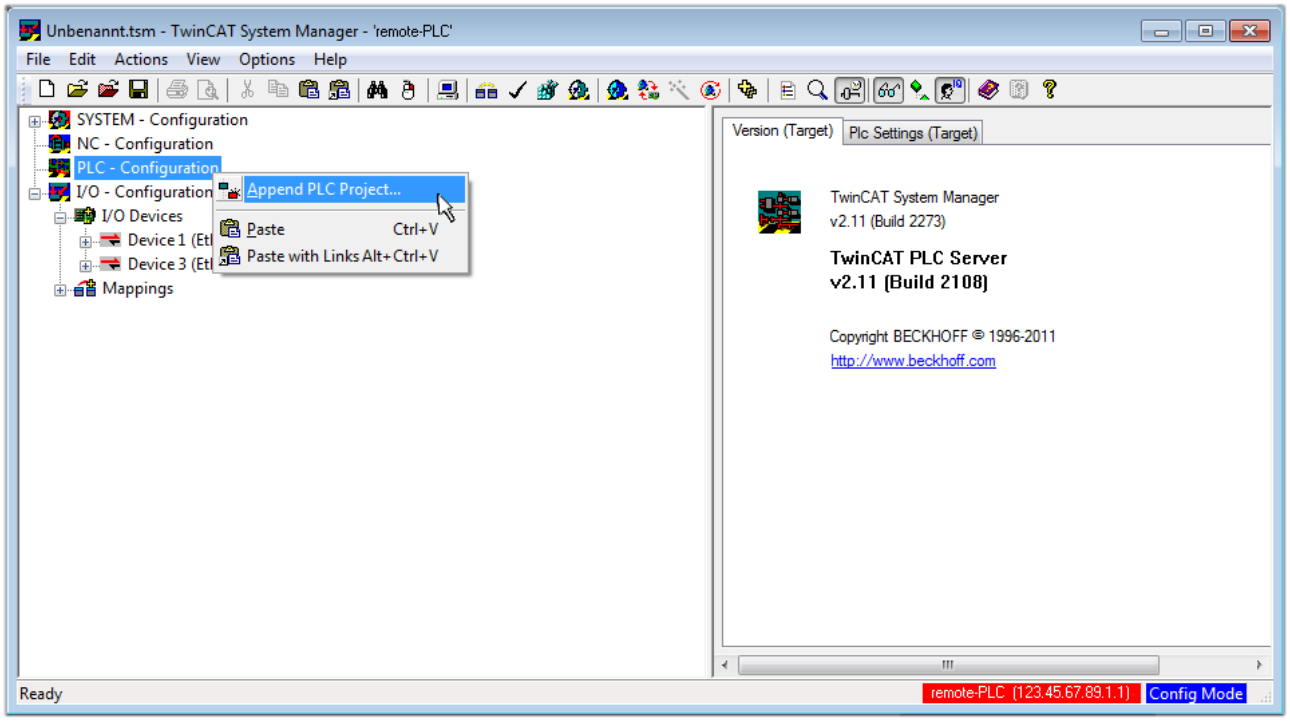


Fig. 64: Appending the TwinCAT PLC Control project

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:

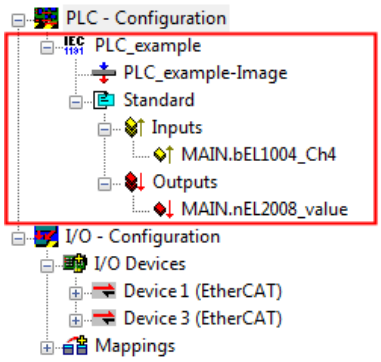


Fig. 65: PLC project integrated in the PLC configuration of the System Manager

The two variables “bEL1004_Ch4” and “nEL2008_value” can now be assigned to certain process objects of the I/O configuration.

Assigning variables

Open a window for selecting a suitable process object (PDO) via the context menu of a variable of the integrated project “PLC_example” and via “Modify Link...” “Standard”:

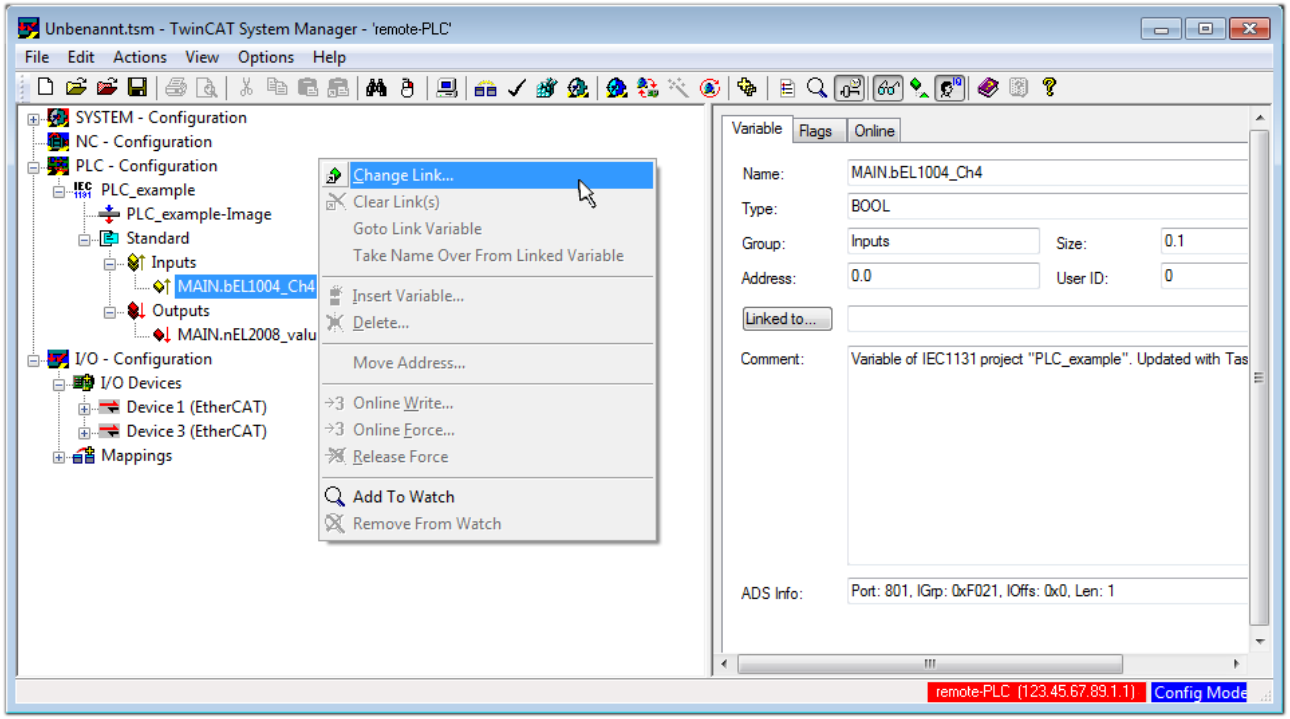


Fig. 66: Creating the links between PLC variables and process objects

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

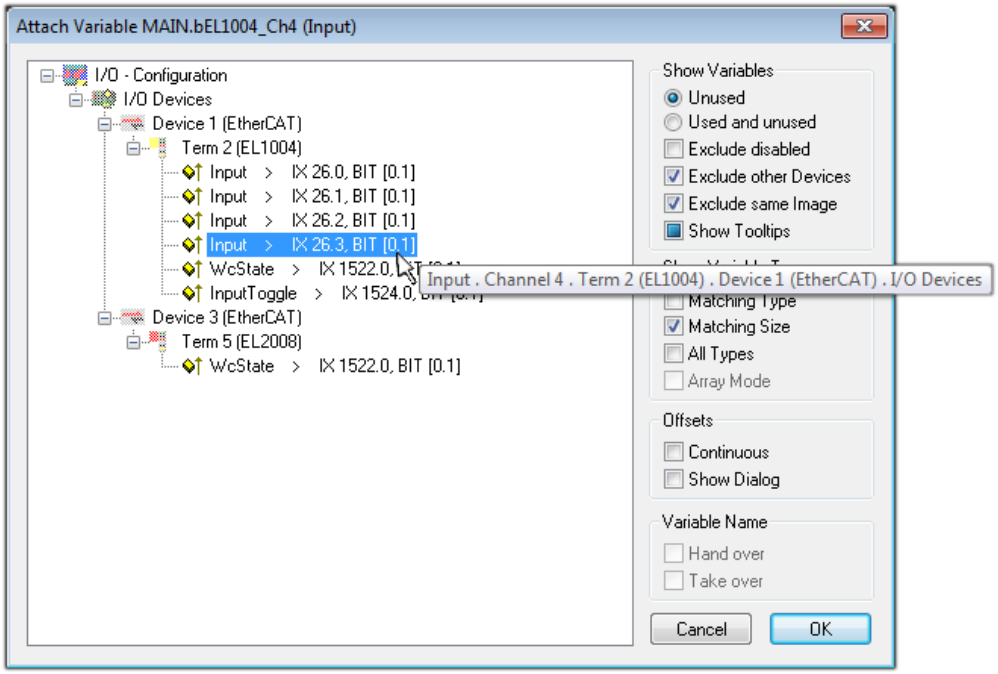


Fig. 67: Selecting BOOL-type PDO

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

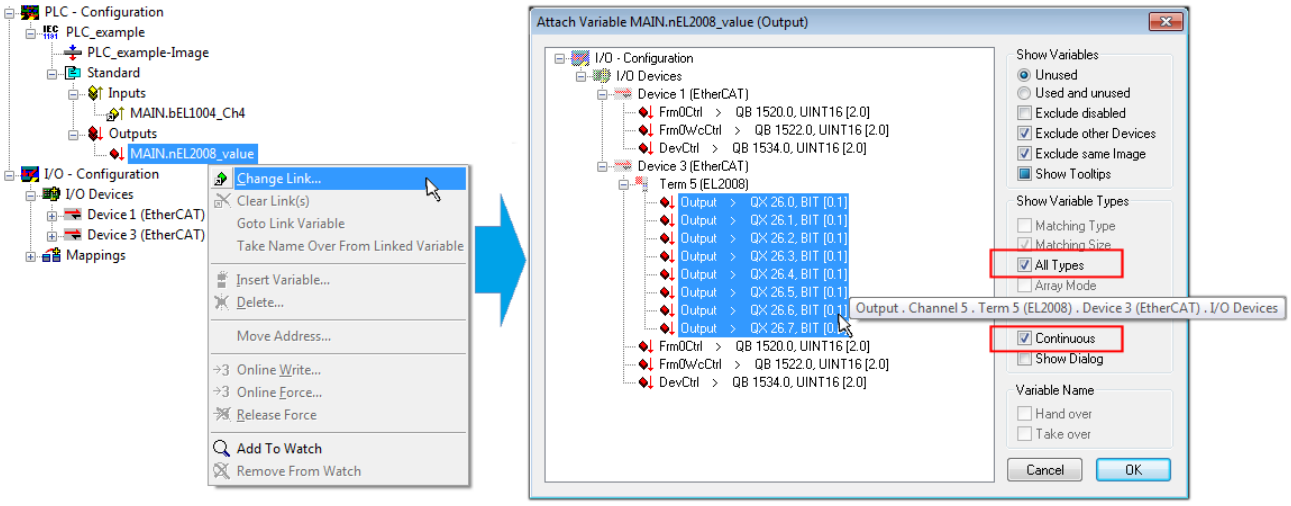


Fig. 68: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

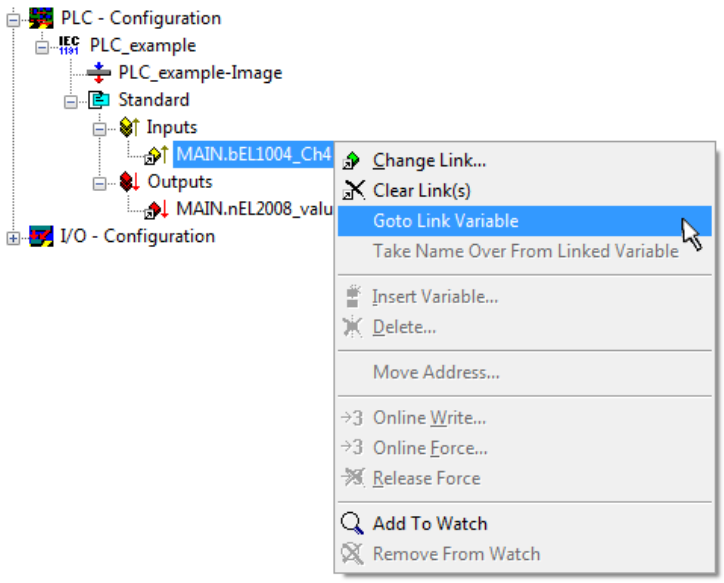
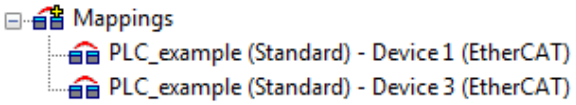


Fig. 69: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

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

This can be visualized in the configuration:



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

Activation of the configuration

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

A few seconds later, the real-time status

is displayed at the bottom right in the System Manager. The PLC system can then be started as described below.

Starting the controller

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

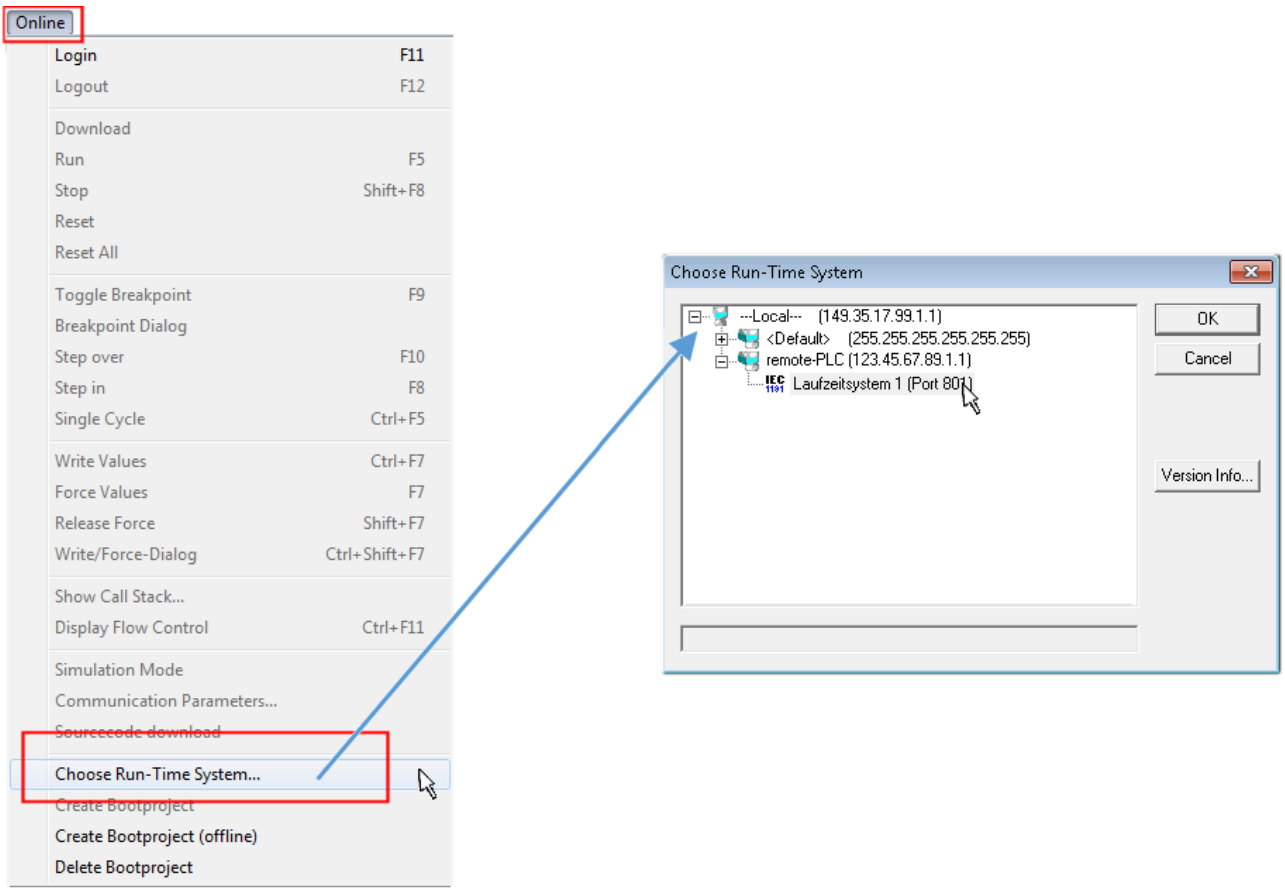


Fig. 70: Choose target system (remote)

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

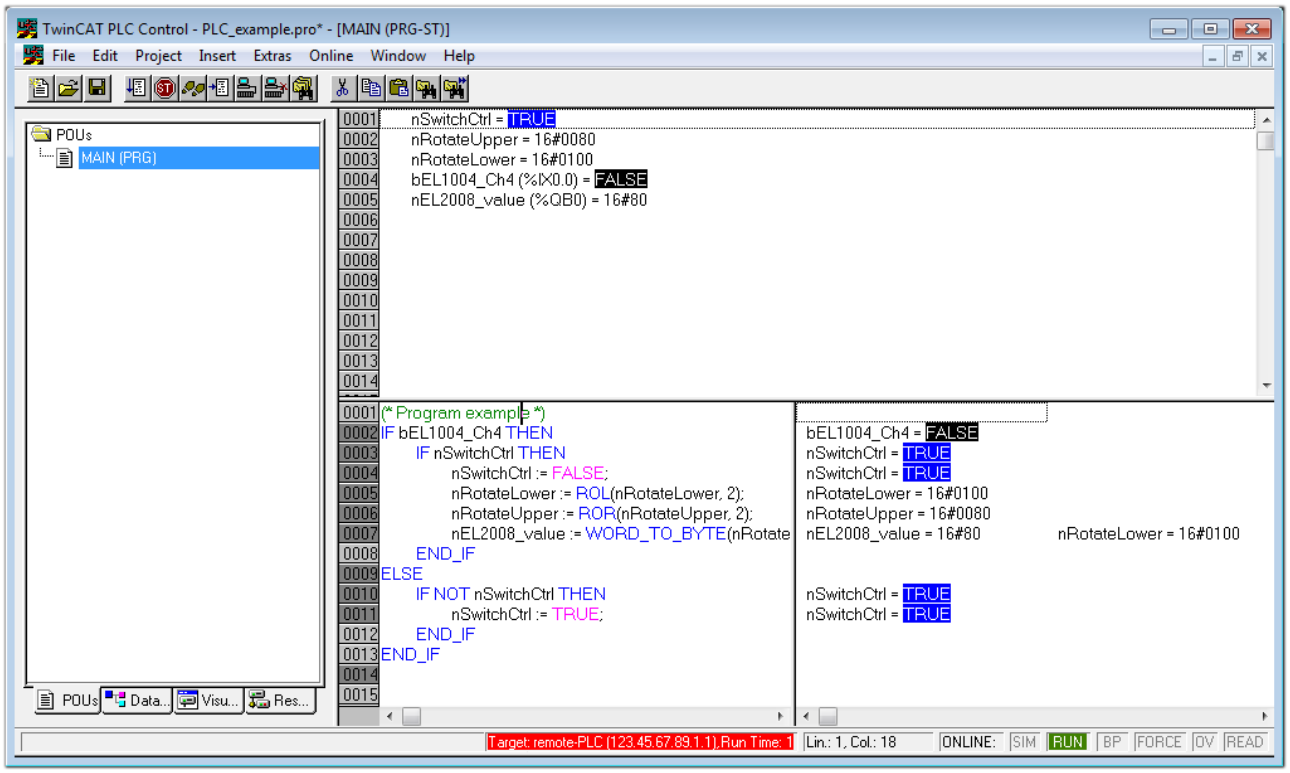


Fig. 71: PLC Control logged in, ready for program startup

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

7.2.2 TwinCAT 3

Startup

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

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

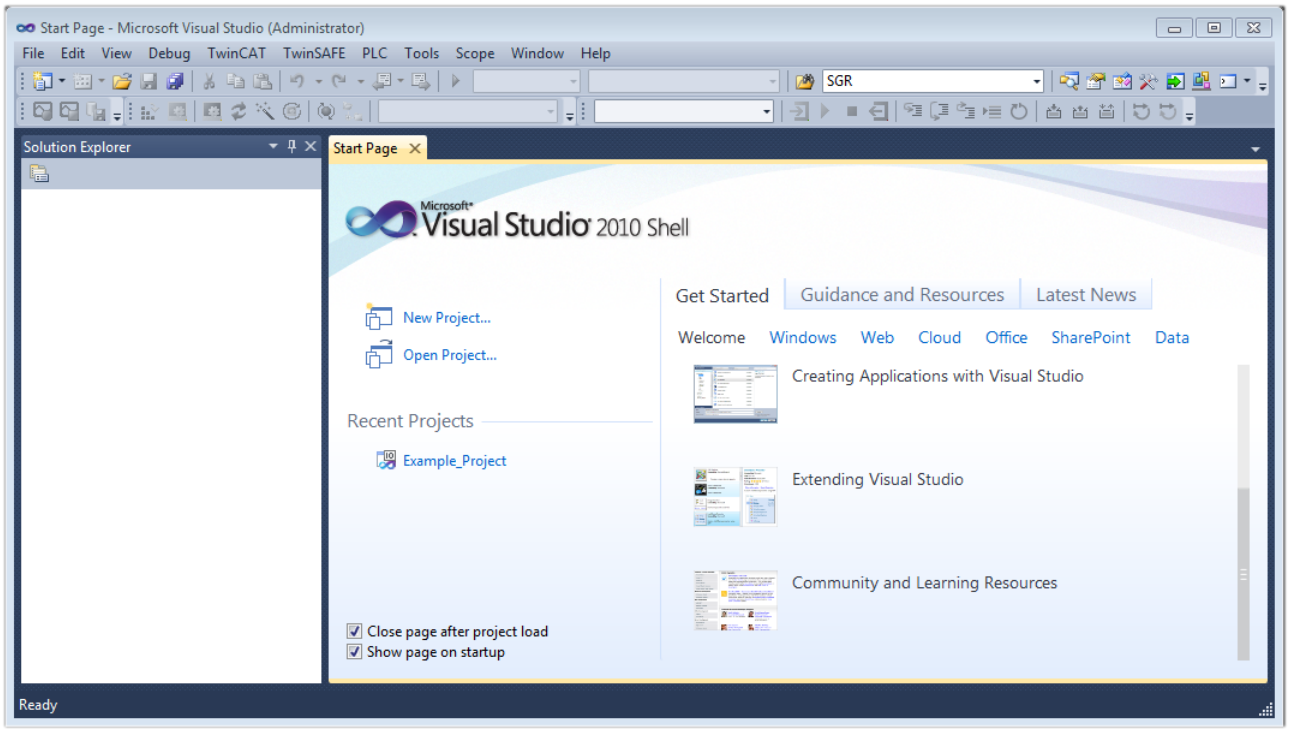


Fig. 72: Initial TwinCAT 3 user interface

First create a new project via New TwinCAT Project... (or under “File”→“New”→ “Project…”). In the following dialog, make the corresponding entries as required (as shown in the diagram):

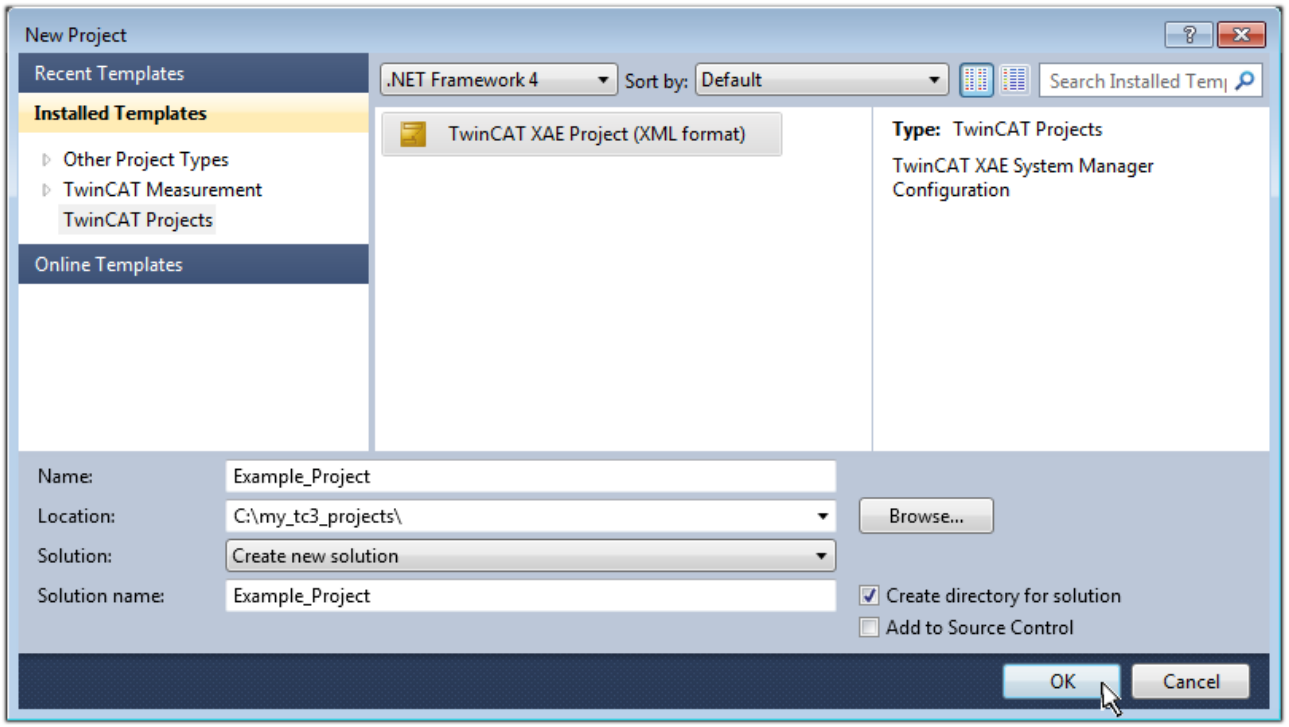


Fig. 73: Create new TwinCAT 3 project

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

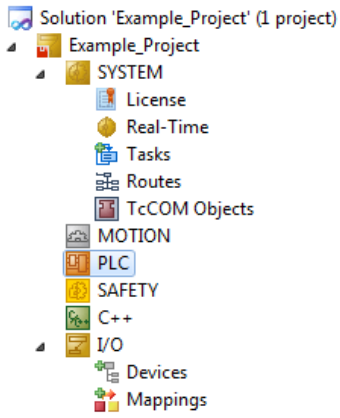
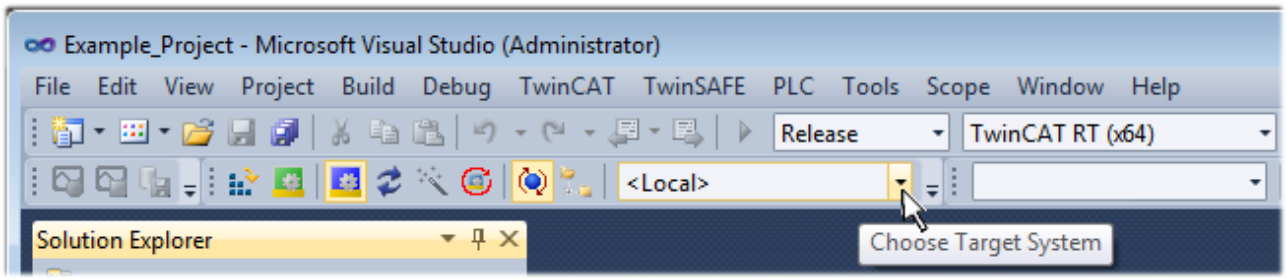


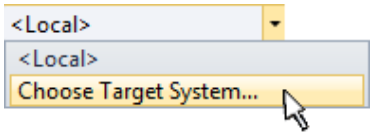
Fig. 74: New TwinCAT 3 project in the project folder explorer

Generally, TwinCAT can be used in local or remote mode. Once the TwinCAT system including the user interface (standard) is installed on the respective PLC (locally), TwinCAT can be used in local mode and the process can be continued with the next step, "Insert Device [▶ 142]".

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



expand the pull-down menu:



and open the following window:

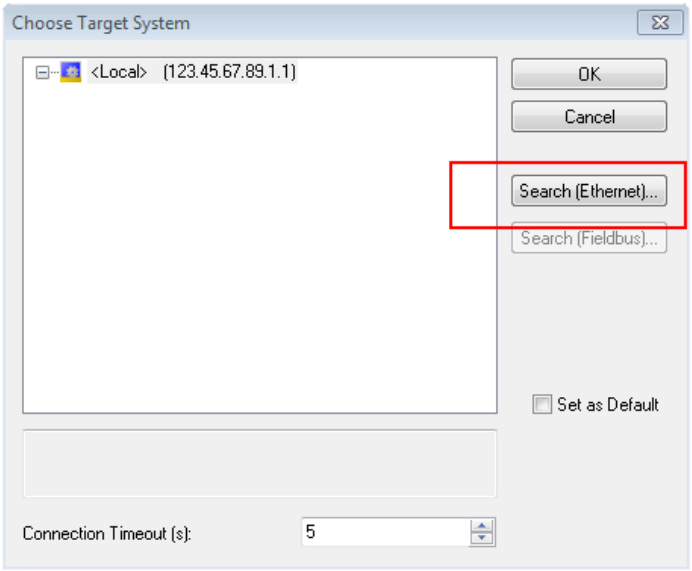


Fig. 75: Selection dialog: Choose the target system

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

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

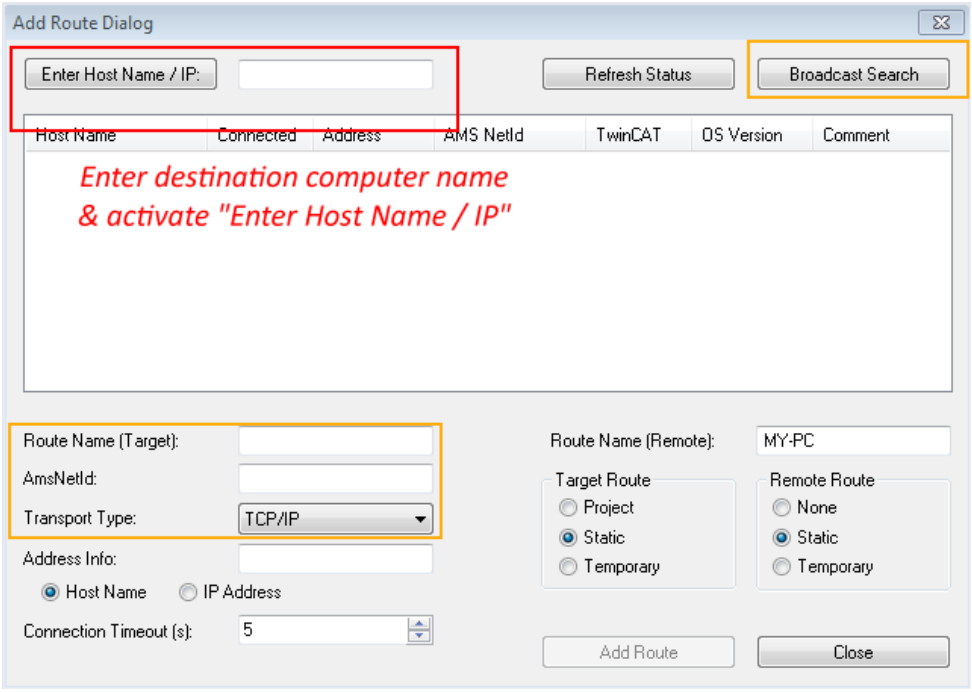
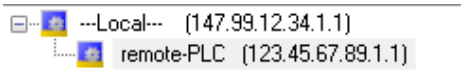


Fig. 76: specify the PLC for access by the TwinCAT System Manager: selection of the target system

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



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

Adding devices

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

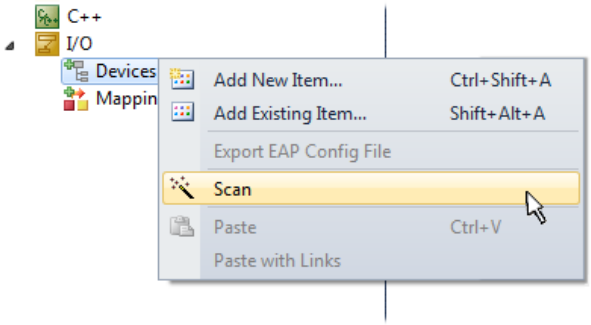


Fig. 77: Select “Scan”

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

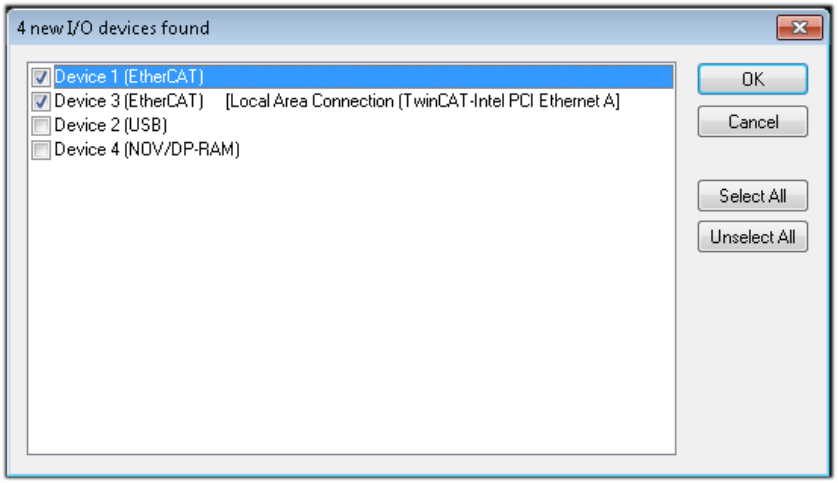


Fig. 78: Automatic detection of I/O devices: selection of the devices to be integrated

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

Based on the example configuration [▶ 128] described at the beginning of this section, the result is as follows:

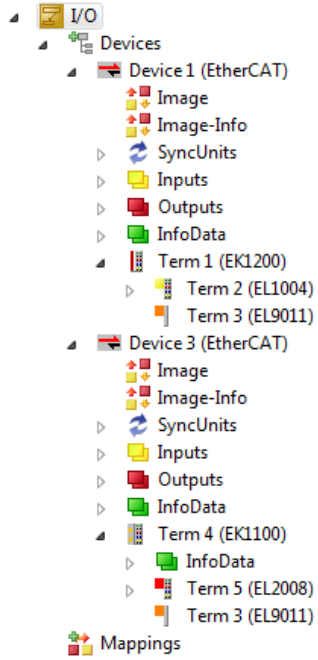


Fig. 79: Mapping of the configuration in VS shell of the TwinCAT 3 environment

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

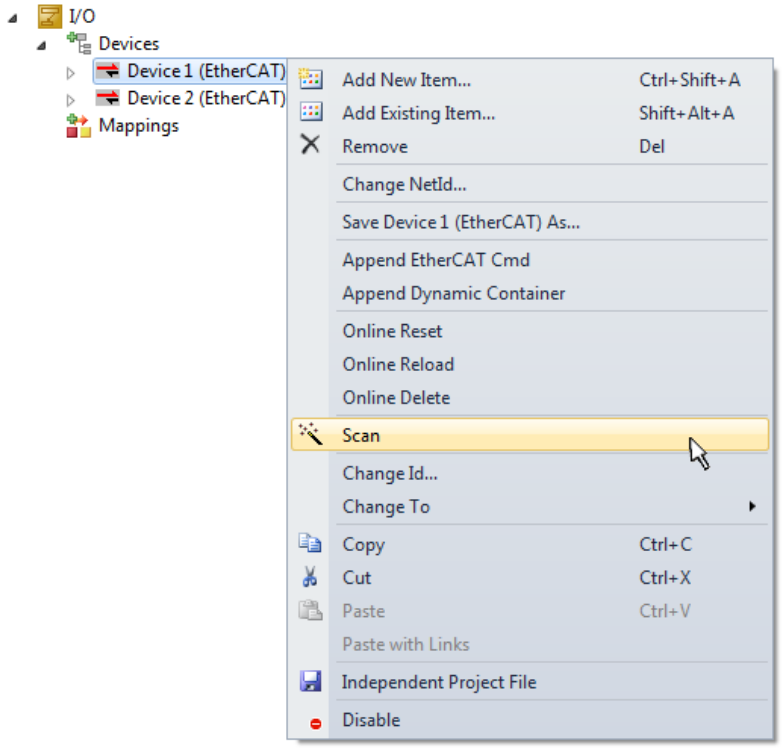


Fig. 80: Reading of individual terminals connected to a device

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

Programming the PLC

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

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

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

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

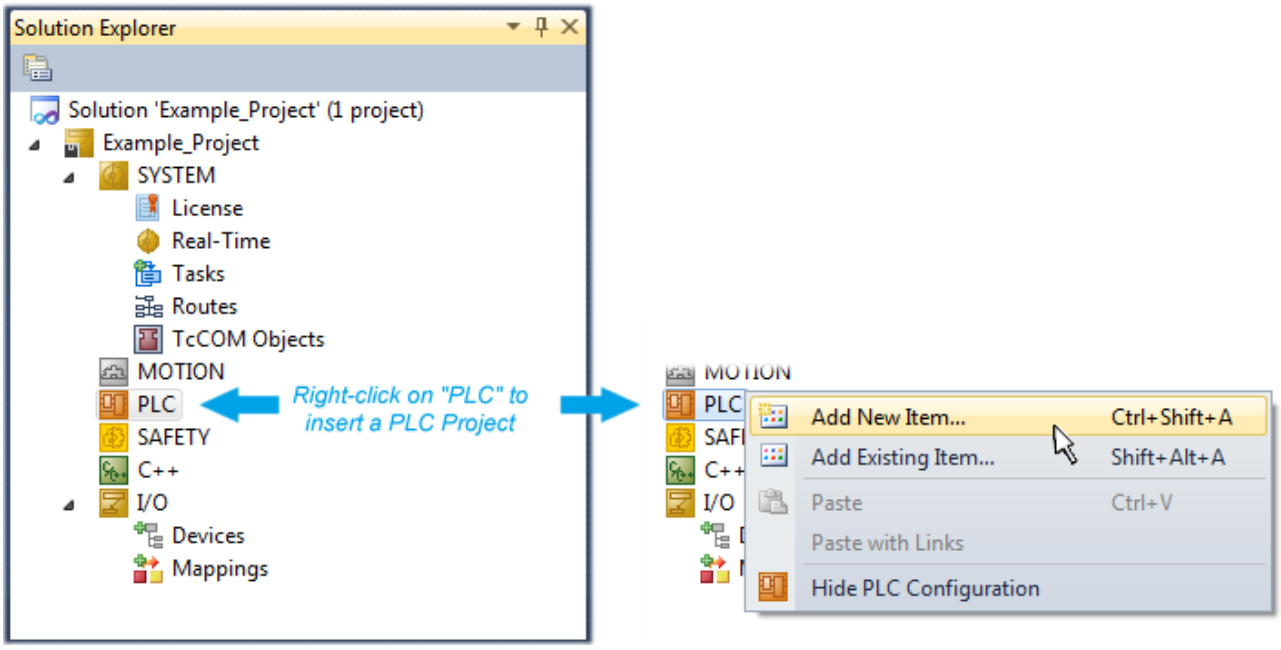


Fig. 81: Adding the programming environment in “PLC”

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

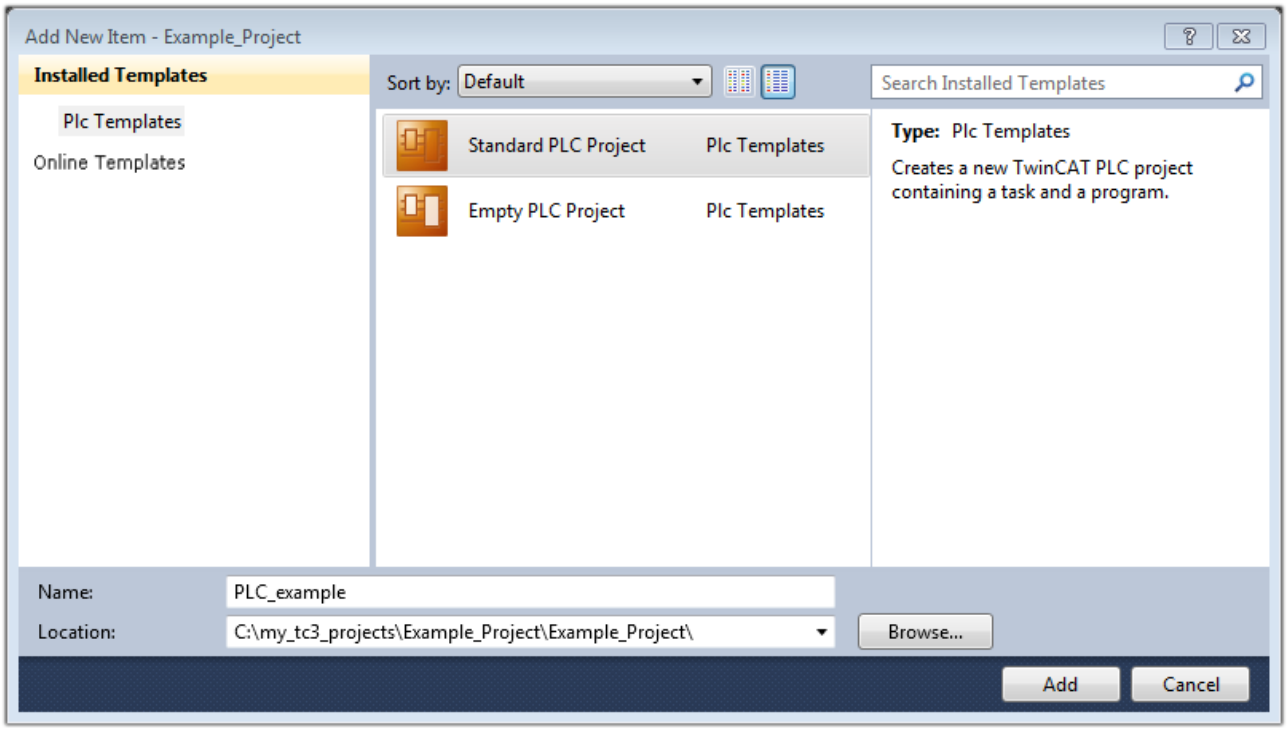


Fig. 82: Specifying the name and directory for the PLC programming environment

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

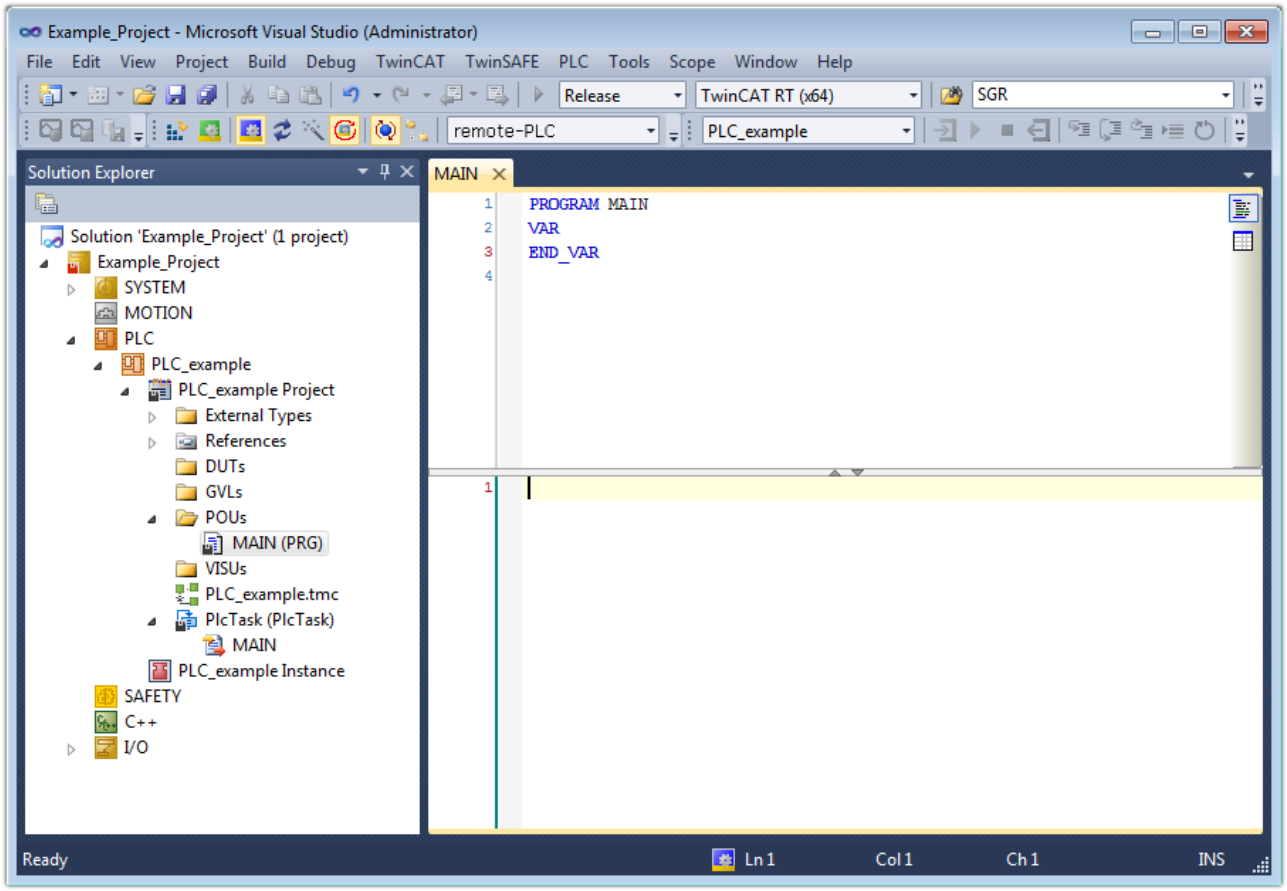


Fig. 83: Initial “Main” program for the standard PLC project

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

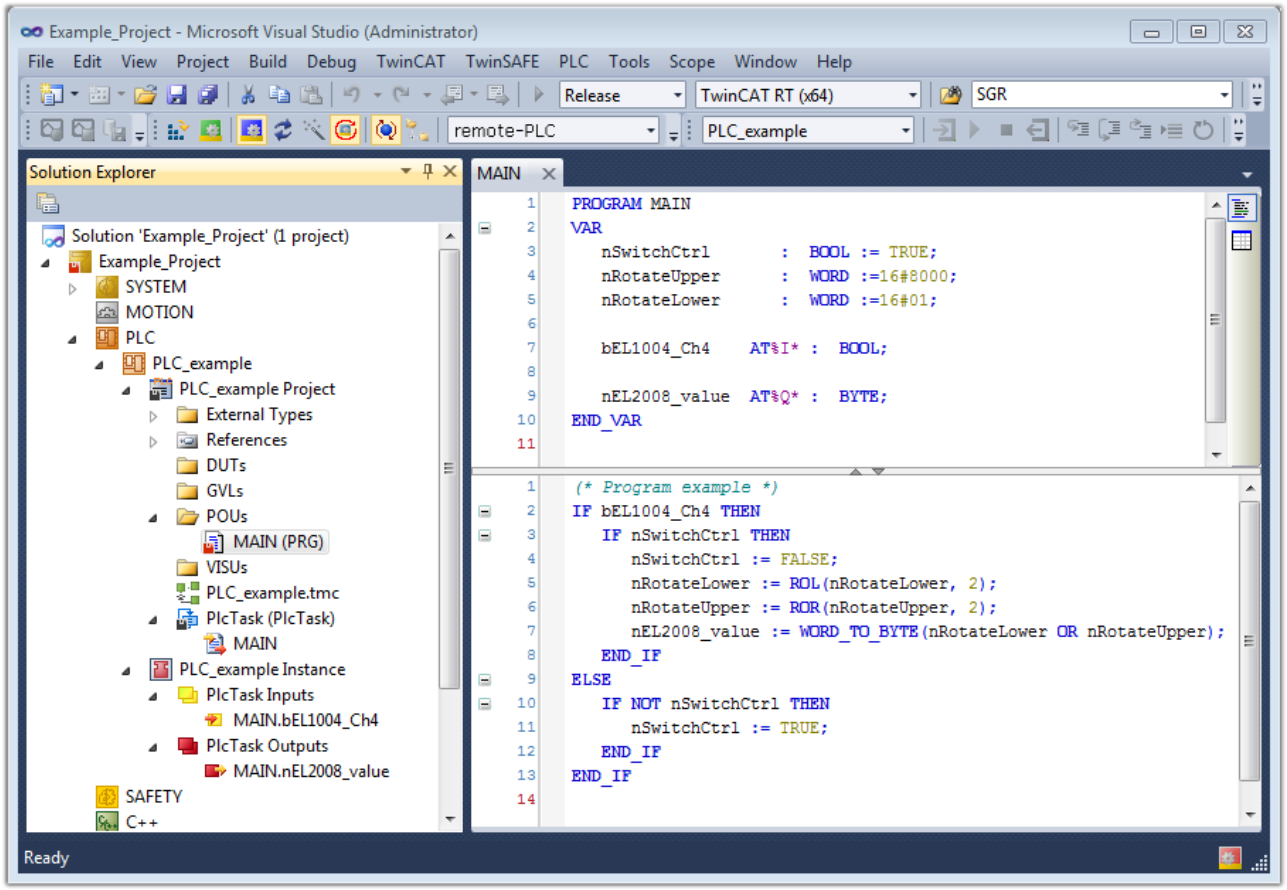


Fig. 84: Example program with variables after a compile process (without variable integration)

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

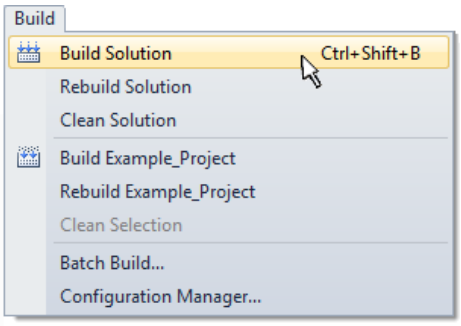
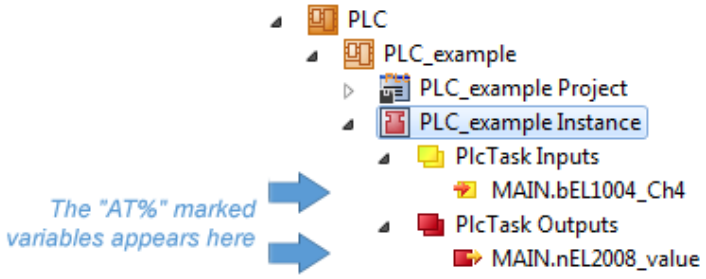


Fig. 85: Start program compilation

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



Assigning variables

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

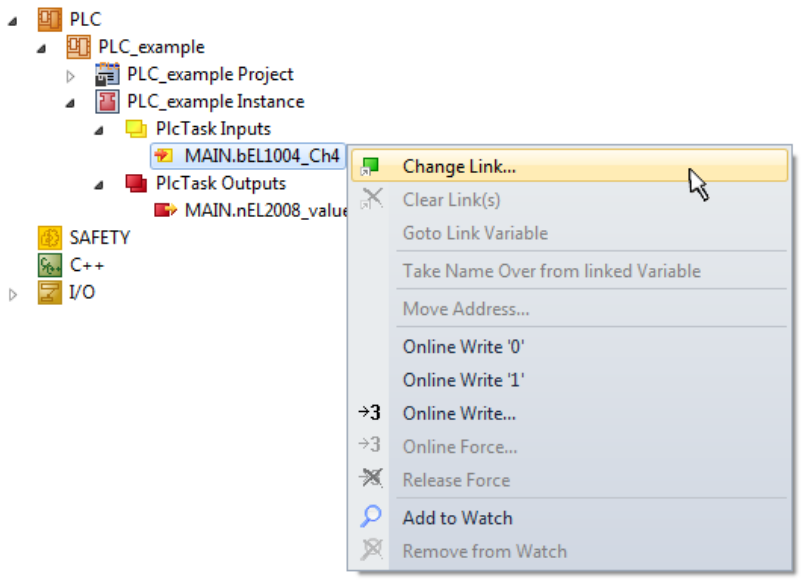


Fig. 86: Creating the links between PLC variables and process objects

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

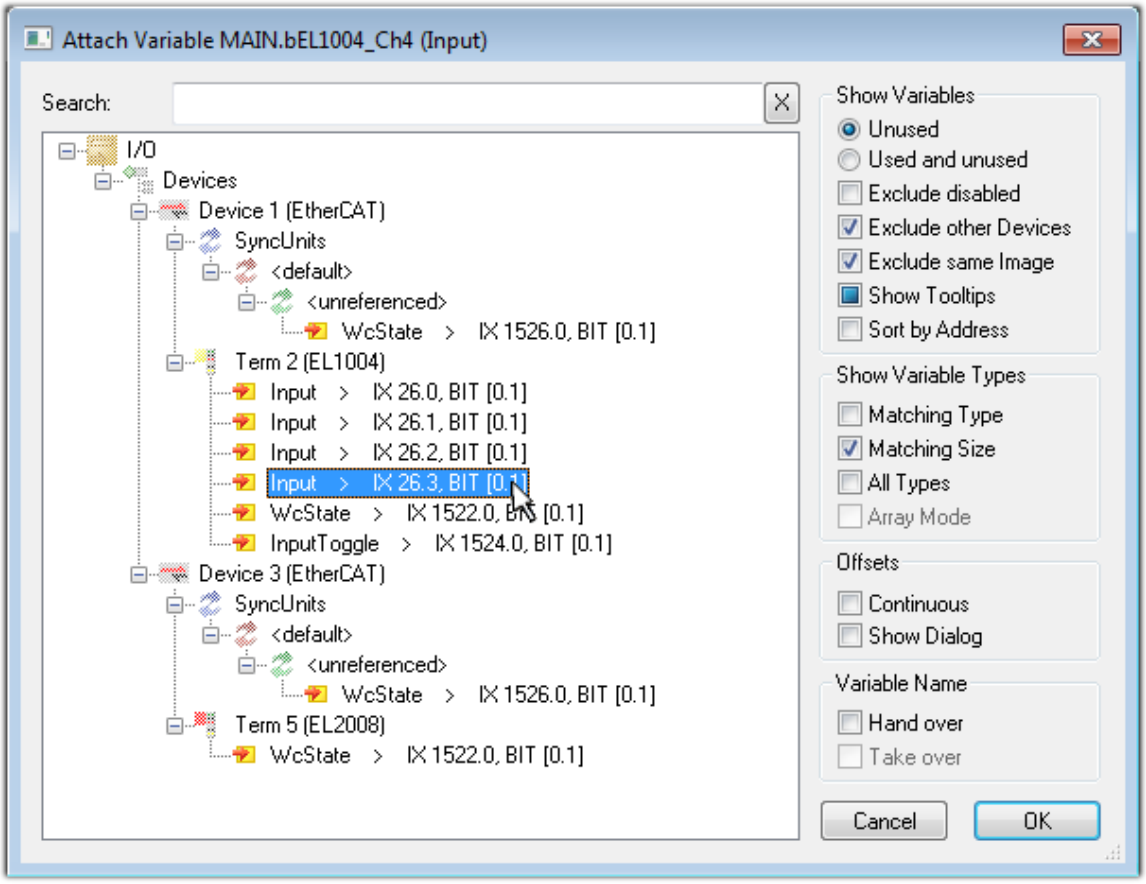


Fig. 87: Selecting BOOL-type PDO

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

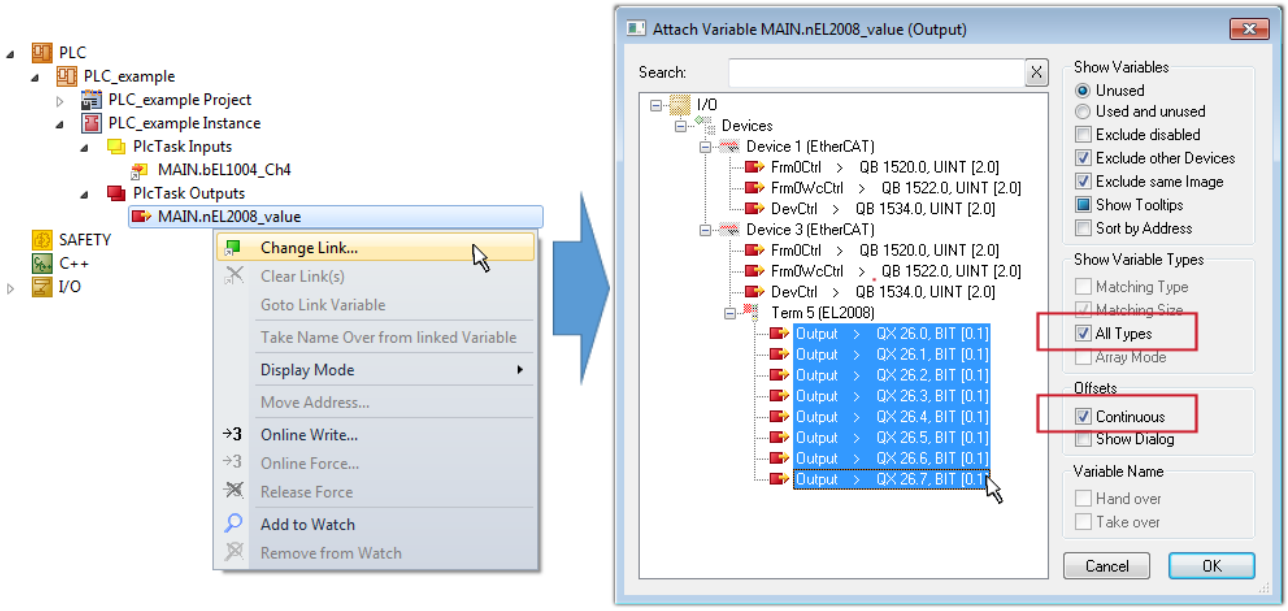


Fig. 88: Selecting several PDOs simultaneously: activate “Continuous” and “All types”

Note that the “Continuous” checkbox was also activated. This is designed to allocate the bits contained in the byte of the “nEL2008_value” variable sequentially to all eight selected output bits of the EL2008 Terminal. It is thus possible to subsequently address all eight outputs of the terminal in the program with a byte corresponding to bit 0 for channel 1 to bit 7 for channel 8 of the PLC. A special symbol ( ) on the yellow or red object of the variable indicates that a link exists. The links can also be checked by selecting “Goto Link Variable” from the context menu of a variable. The opposite linked object, in this case the PDO, is automatically selected:

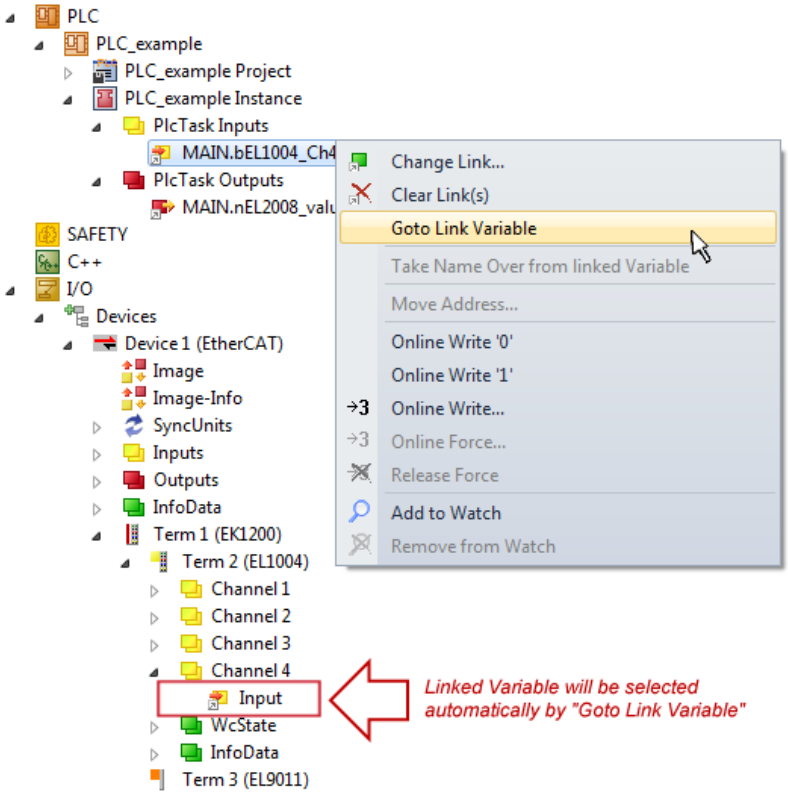


Fig. 89: Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example

The process of creating links can also be performed in the opposite direction, i.e. starting with individual PDOs to a variable. However, in this example, it would not be possible to select all output bits for the EL2008, since the terminal only makes individual digital outputs available. If a terminal has a byte, word,

integer or similar PDO, it is also possible to allocate this to a set of bit-standardized variables. Here, too, a “Goto Link Variable” can be executed in the other direction, so that the respective PLC instance can then be selected.

Note on type of variable assignment

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

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

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

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

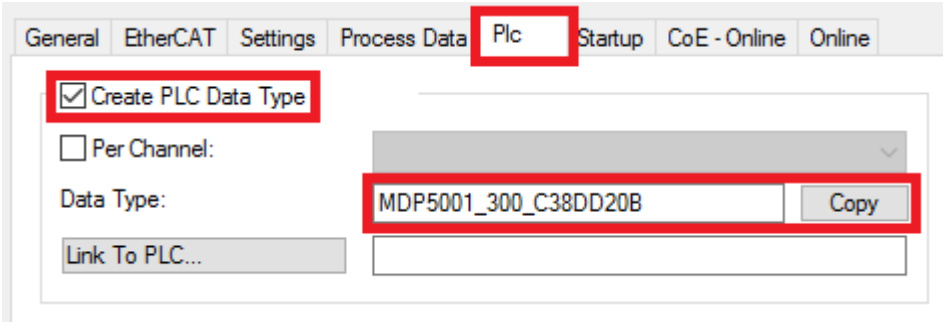


Fig. 90: Creating a PLC data type

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

```
MAIN
1  PROGRAM MAIN
2  VAR
3      EL3001  : MDP5001_300_C38DD20B;
4  END_VAR
```

Fig. 91: 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.

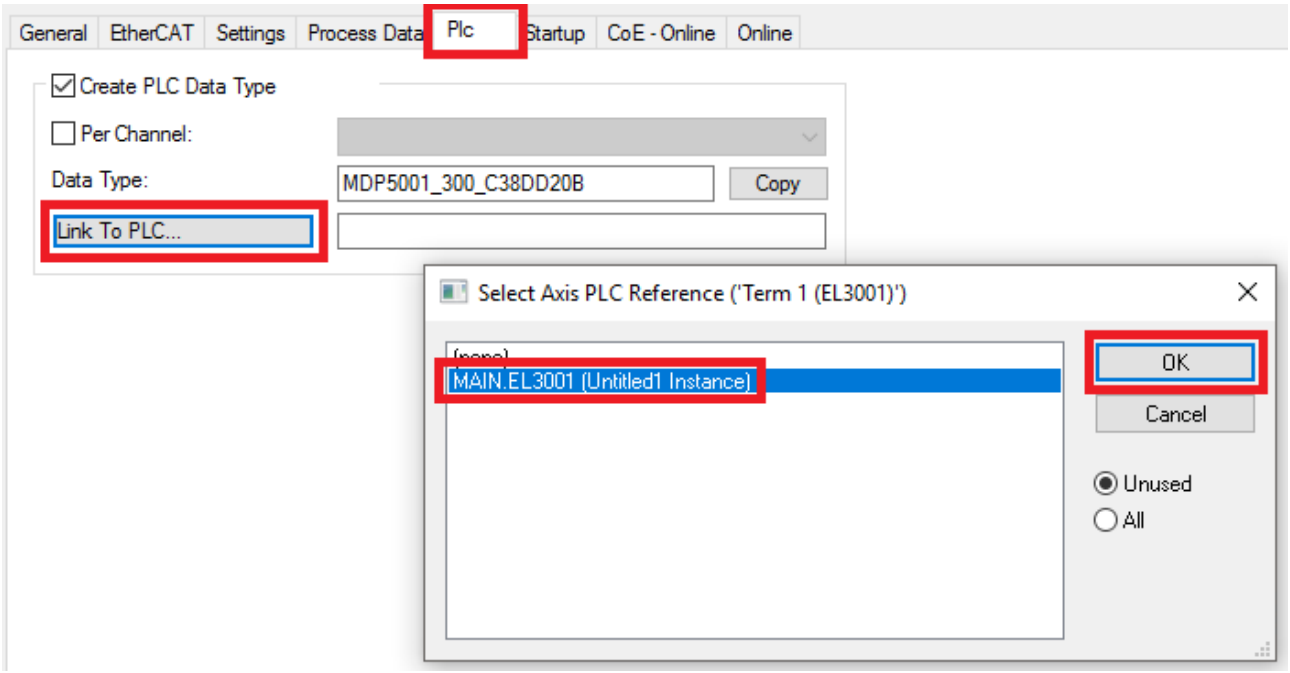


Fig. 92: Linking the structure

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

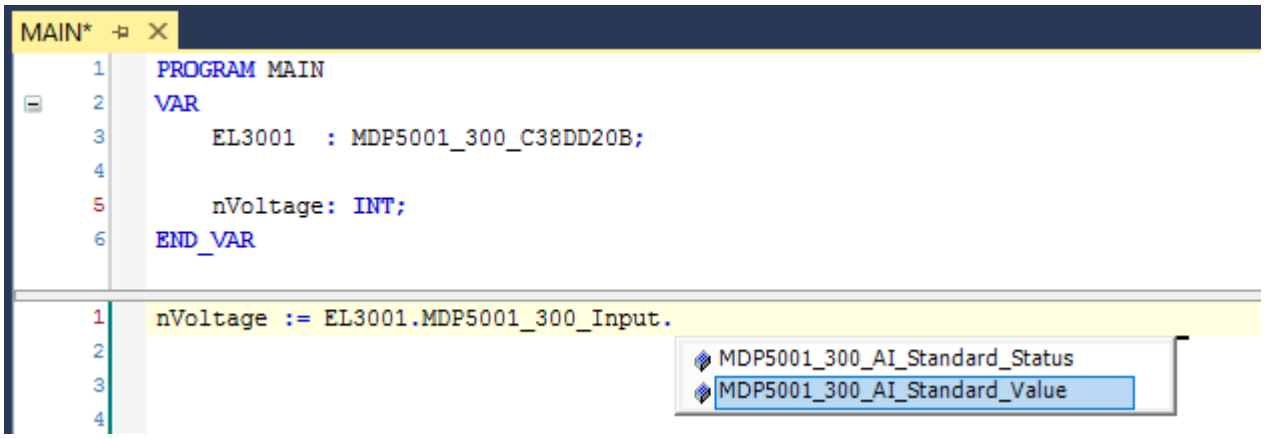
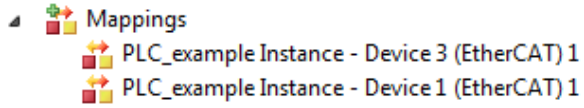


Fig. 93: Reading a variable from the structure of the process data

Activation of the configuration

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



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

Starting the controller

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

Fig. 94: TwinCAT 3 development environment (VS shell): logged-in, after program startup

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

7.3 TwinCAT Development Environment

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

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

Details:

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

Additional features:

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

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

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

7.3.1 Installation of the TwinCAT real-time driver

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

This can be done in several ways.

A: Via the TwinCAT Adapter dialog

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

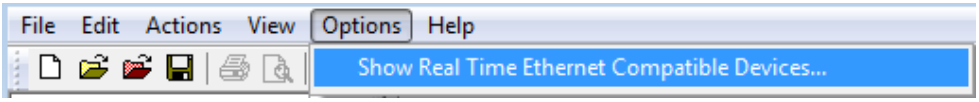


Fig. 95: System Manager “Options” (TwinCAT 2)

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

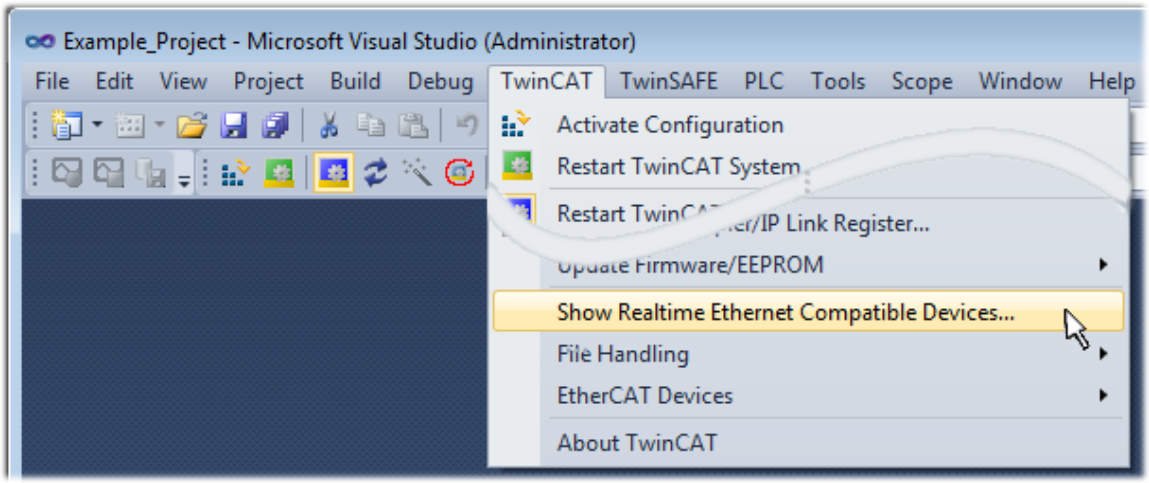


Fig. 96: Call up under VS Shell (TwinCAT 3)

B: Via TcRteInstall.exe in the TwinCAT directory

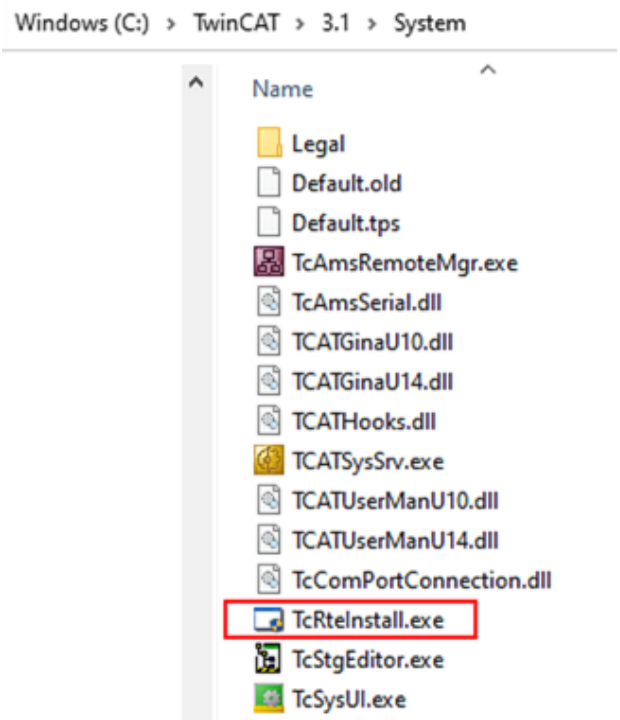


Fig. 97: TcRteInstall in the TwinCAT directory

In both cases, the following dialog appears:

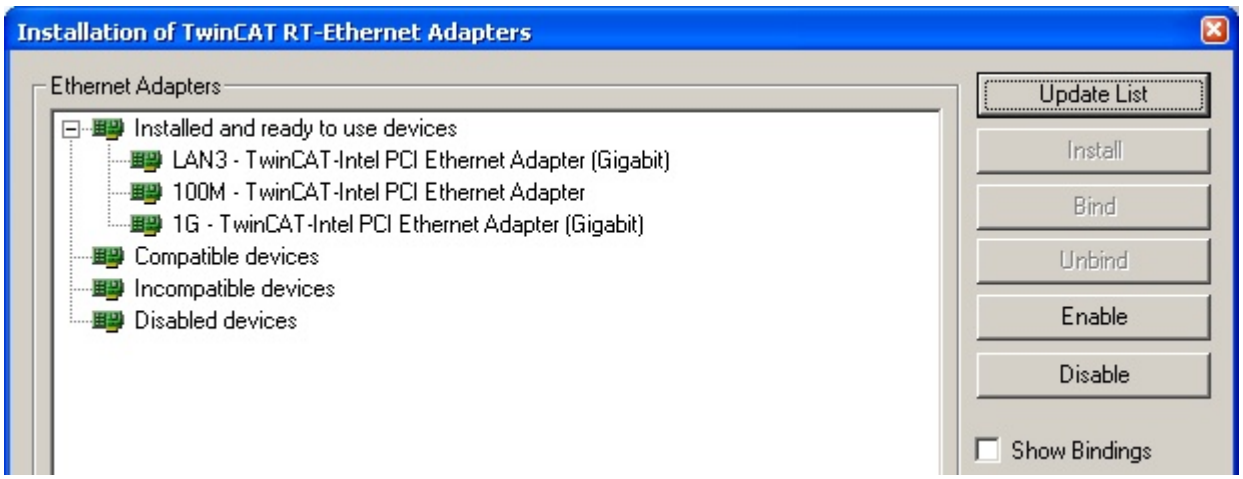


Fig. 98: Overview of network interfaces

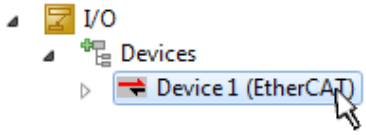
Interfaces listed under “Compatible devices” can be assigned a driver via the “Install” button. A driver should only be installed on compatible devices.

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

Fig. 99: EtherCAT device properties (TwinCAT 2): click on “Compatible Devices…” of tab “Adapter”

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



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

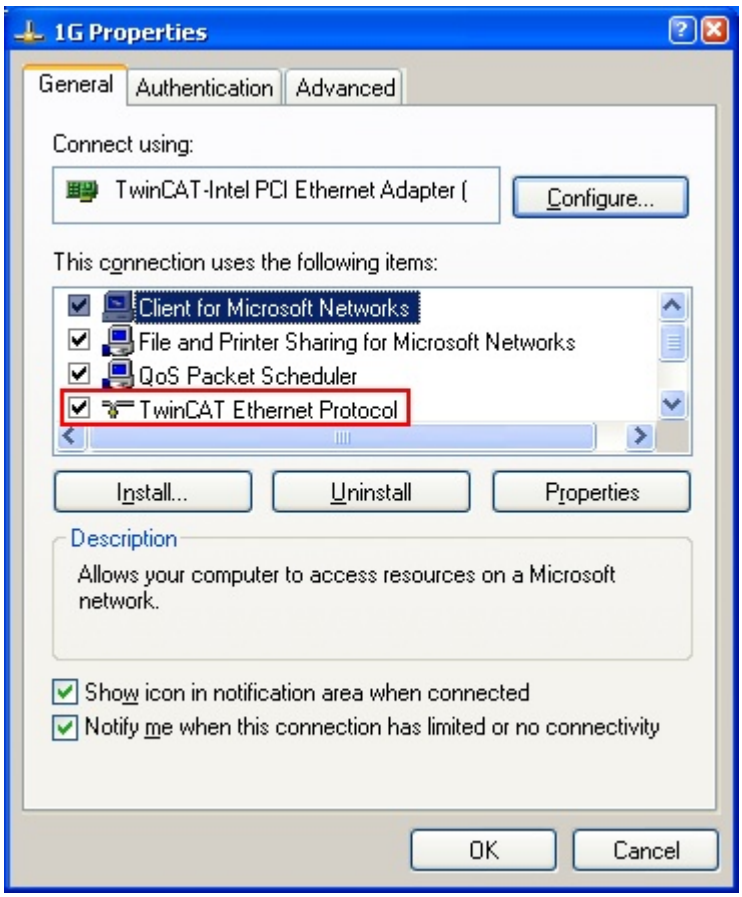


Fig. 100: Windows properties of the network interface

A correct setting of the driver could be:

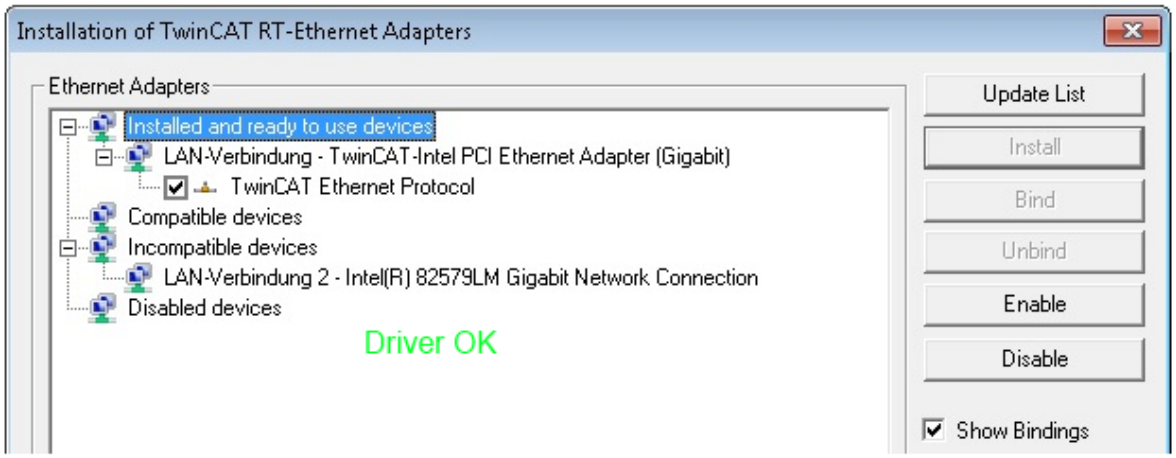


Fig. 101: Exemplary correct driver setting for the Ethernet port

Other possible settings have to be avoided:

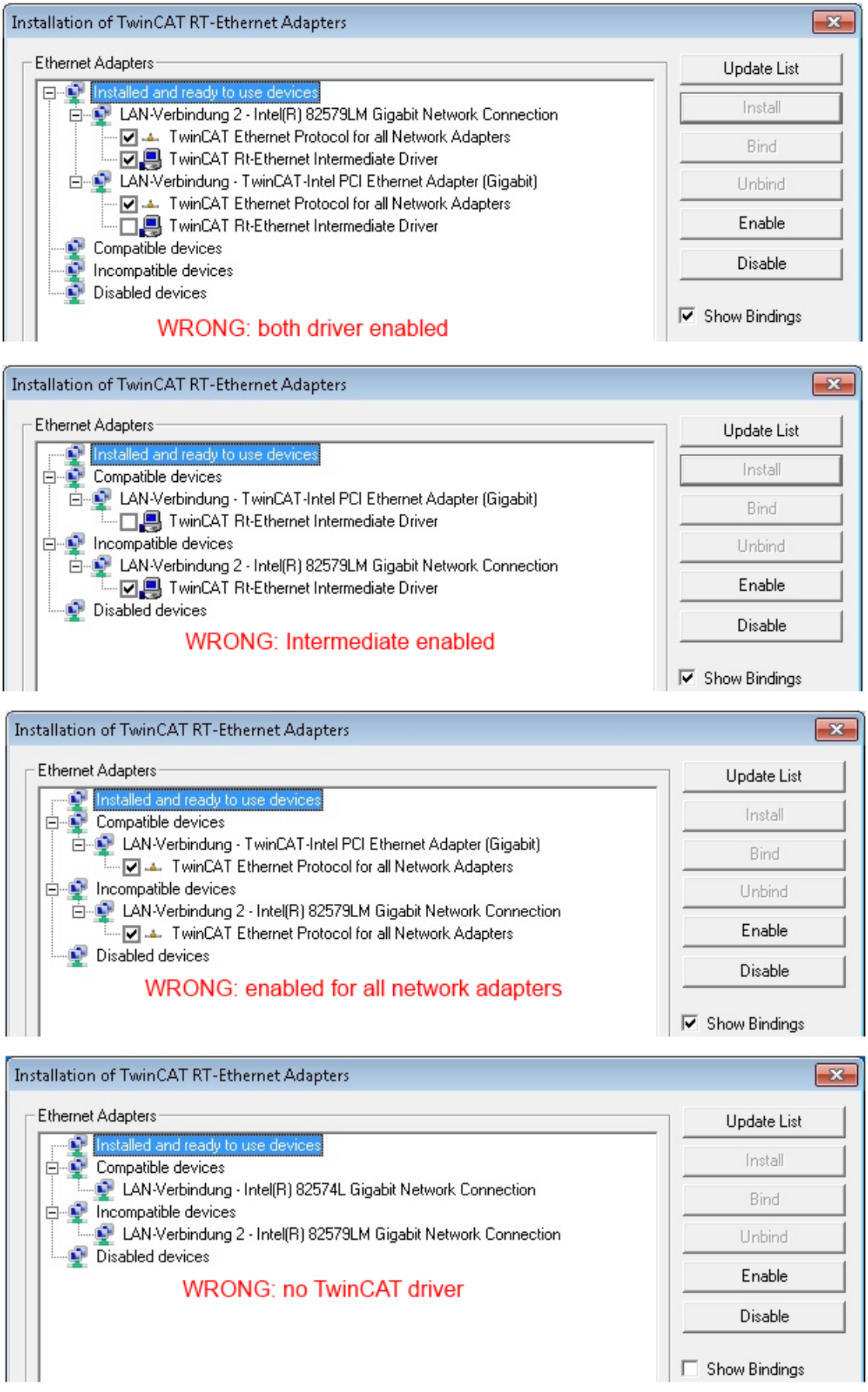


Fig. 102: Incorrect driver settings for the Ethernet port

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.

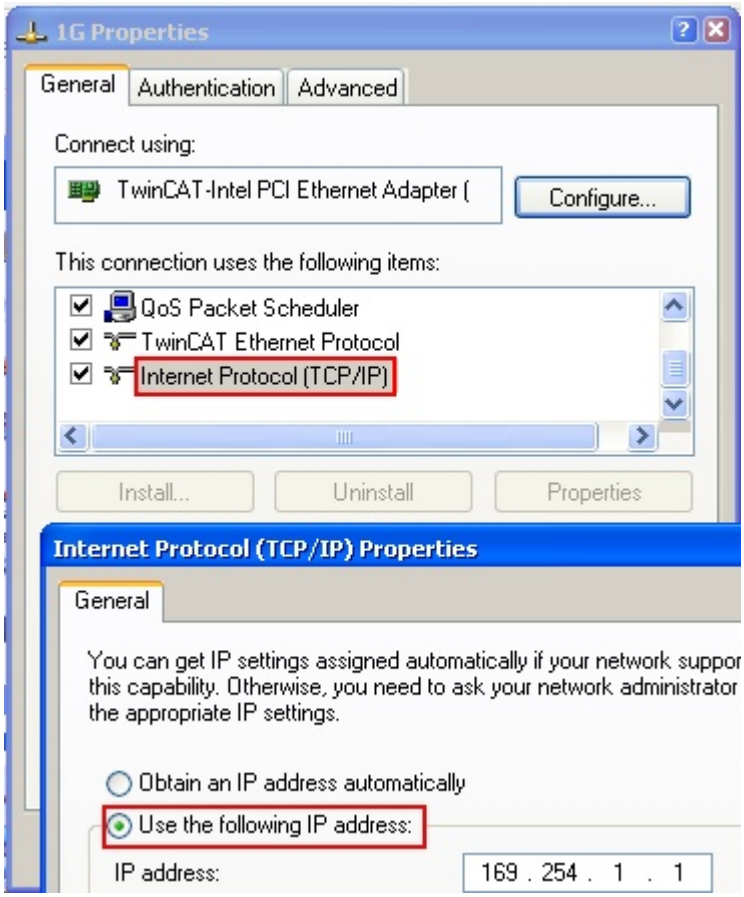


Fig. 103: TCP/IP setting for the Ethernet port

7.3.2 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 [▶ 163] 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”

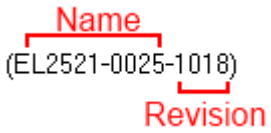


Fig. 104: Identifier structure

The order identifier consisting of name + type (here: EL2521-0025) describes the device function. The revision indicates the technical progress and is managed by Beckhoff. In principle, a device with a higher revision can replace a device with a lower revision, unless specified otherwise, e.g. in the documentation. Each revision has its own ESI description. See further notes [▶ 11].

Online description

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

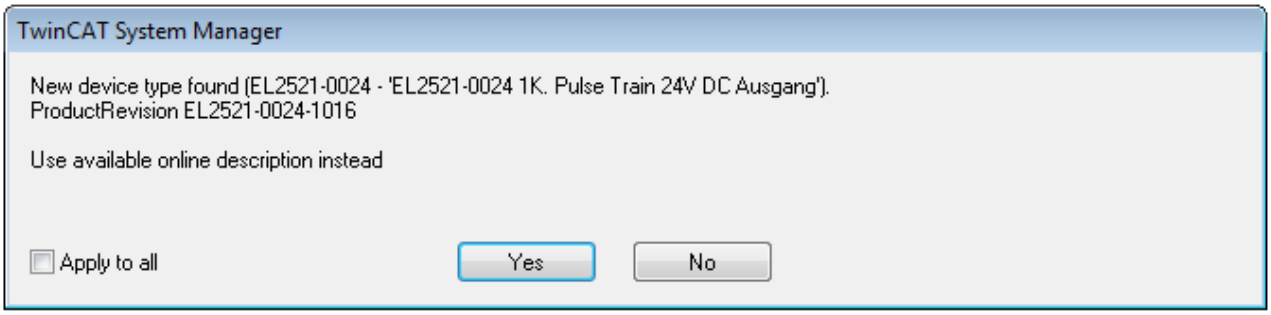


Fig. 105: OnlineDescription information window (TwinCAT 2)

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

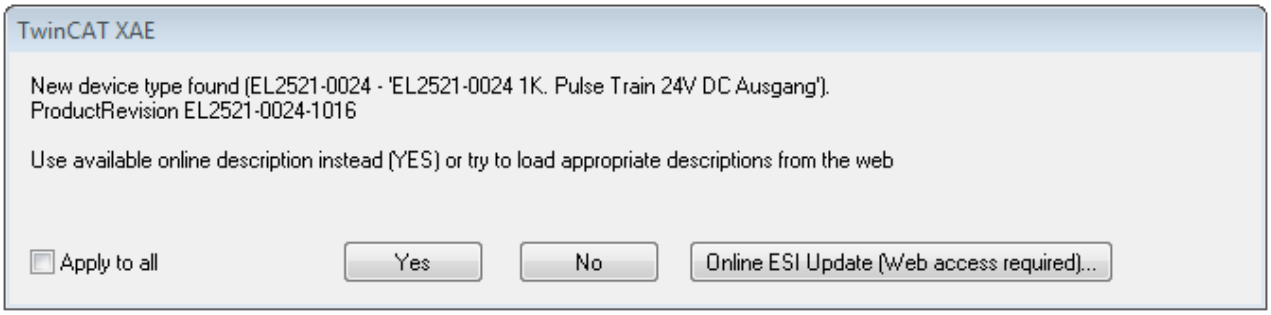


Fig. 106: Information window OnlineDescription (TwinCAT 3)

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

NOTICE

Changing the "usual" configuration through a scan

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

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

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

Refer in particular to the chapter "General notes on the use of Beckhoff EtherCAT IO components" and for manual configuration to the chapter "Offline configuration creation [▶ 165]".

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. 107: 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)*.

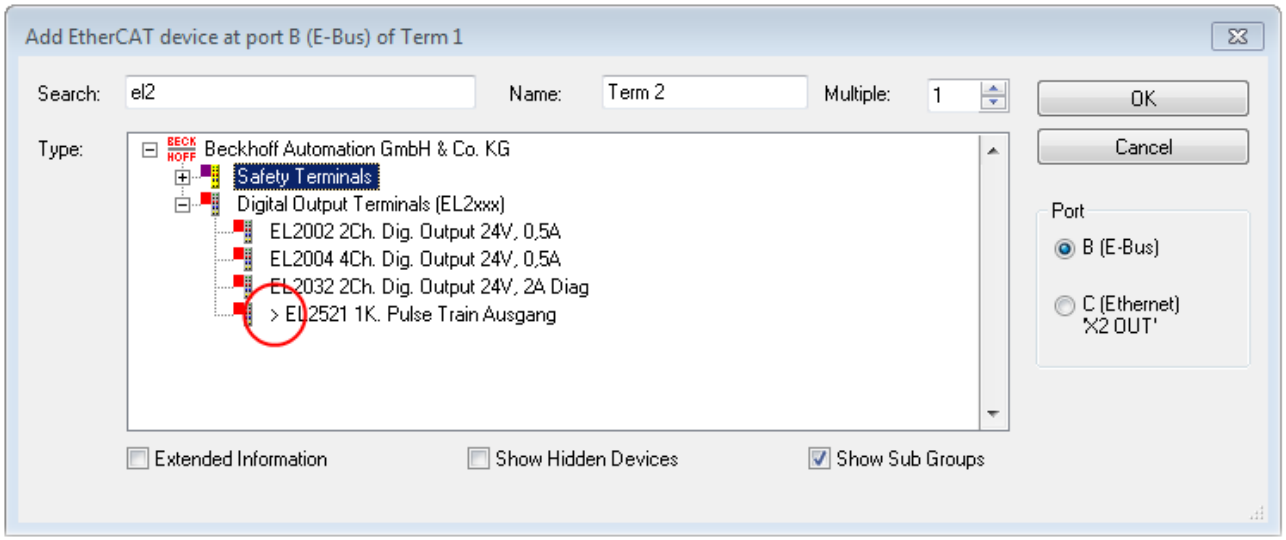


Fig. 108: Indication of an online recorded ESI of EL2521 as an example

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.

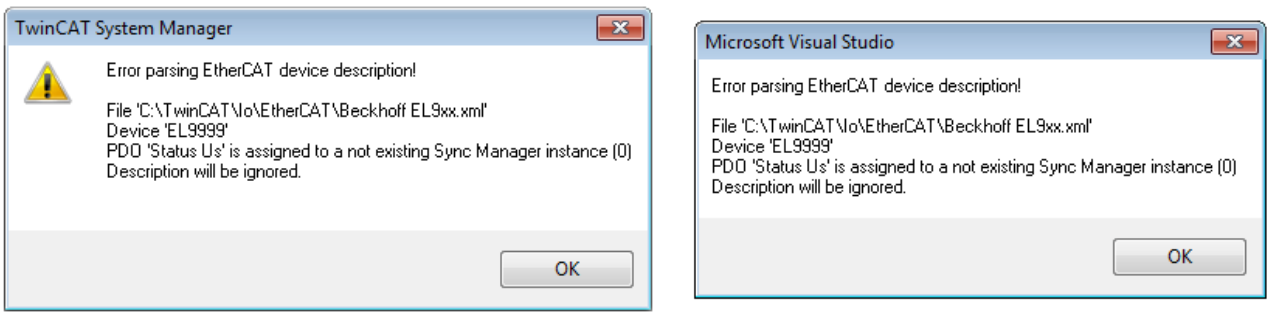


Fig. 109: Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3)

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

7.3.3 TwinCAT ESI Updater

The ESI Updater 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.

7.3.3.1 TwinCAT 3

Step 1) Updating the ESI data sets

The Updater can be accessed under:
“TwinCAT” → “EtherCAT Devices” → “Update Device Description (via ETG Website)...”

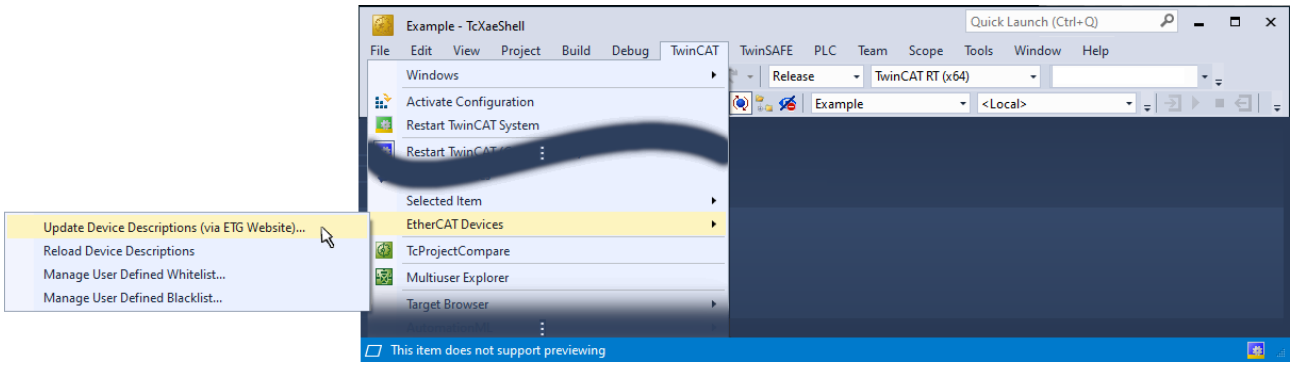
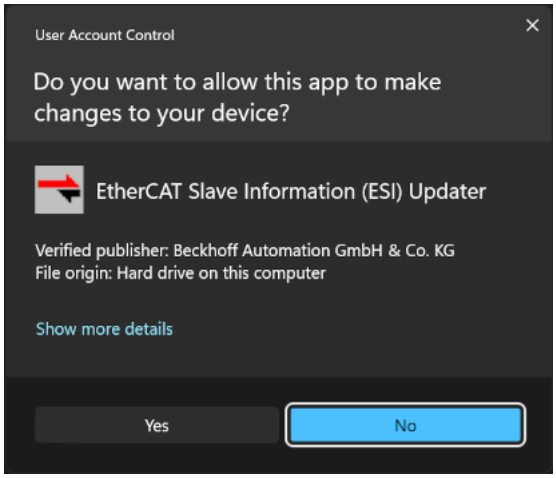


Fig. 110: Calling the ESI Updater (TwinCAT 3)

A prompt for Windows user account control appears after the call, which must be confirmed with [Yes]:



The "Updater" window then appears, showing the configuration data source:

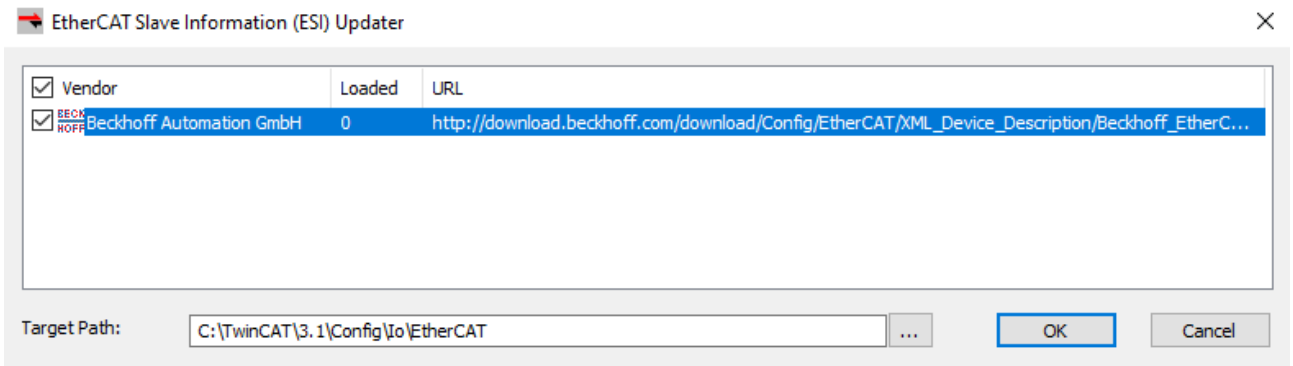
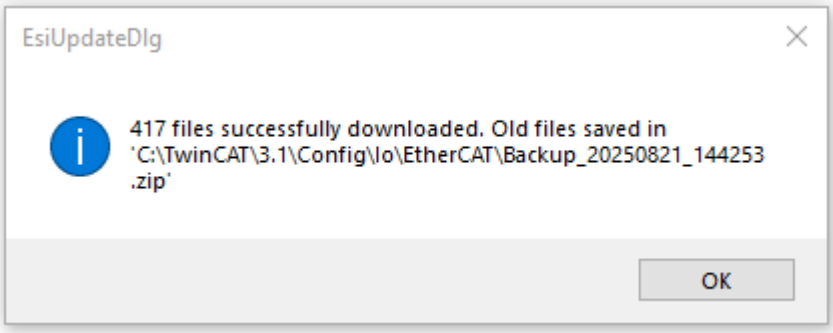


Fig. 111: (ESI) Updater

Select the desired line and click [OK] to start the download or update process. This may take a few minutes.

The completion of the download process is indicated by the message



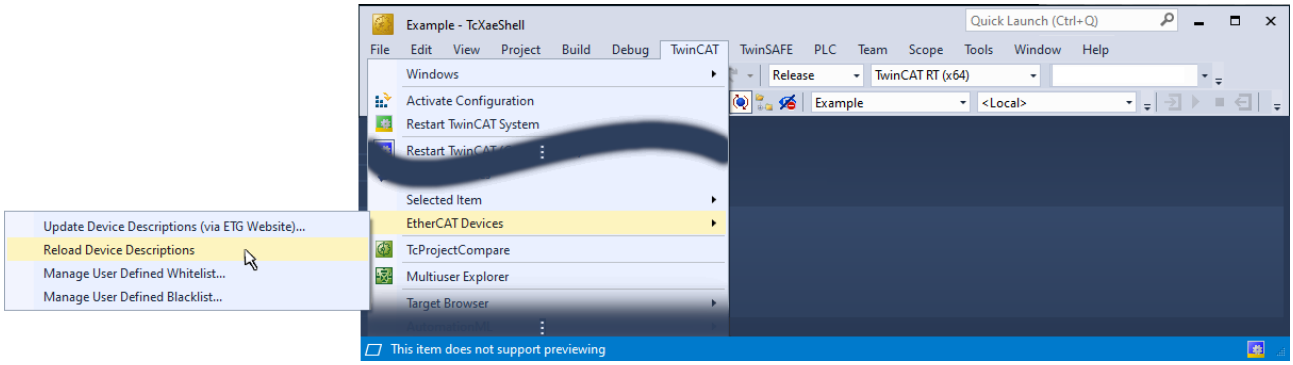
, which shows the number of downloaded files and the storage location or the backup zip folder. Click [OK] to close the window.

Note: Manually added ESI files are not moved to the backup zip folder and remain accessible in TwinCAT. Only the ESI files supplied by Beckhoff are managed automatically.

The (ESI) Updater window that is still open can now be closed with [X] (top right) or [Cancel].

Step 2) Update ESI cache

The ESI files must now be loaded into the application cache in order to be able to use them. This is done either by restarting TwinCAT or by calling up
"TwinCAT" → "EtherCAT Devices" → "Reload Device Descriptions":



The status of the cache rebuild can be seen in the lower status bar in TwinCAT.

7.3.3.2 TwinCAT 2

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

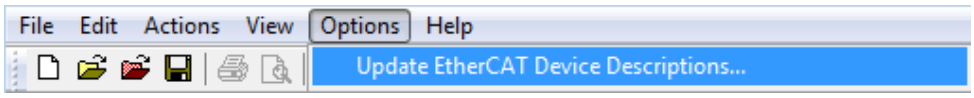


Fig. 112: Using the ESI Updater (>= TwinCAT 2.11)

The call is made via:
“Options” → “Update EtherCAT Device Descriptions”.

7.3.4 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” [▶ 159].

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 [▶ 171] (Ethernet port at the IPC)
- detecting the connected EtherCAT devices [▶ 172]. This step can be carried out independent of the preceding step
- troubleshooting [▶ 175]

The scan with existing configuration [▶ 176] can also be carried out for comparison.

7.3.5 OFFLINE configuration creation

Creating the EtherCAT device

Create an EtherCAT device in an empty System Manager window.

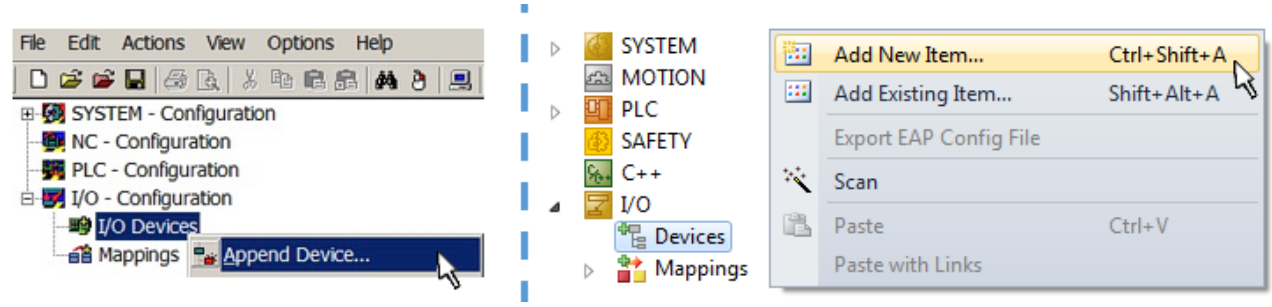


Fig. 113: Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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”.

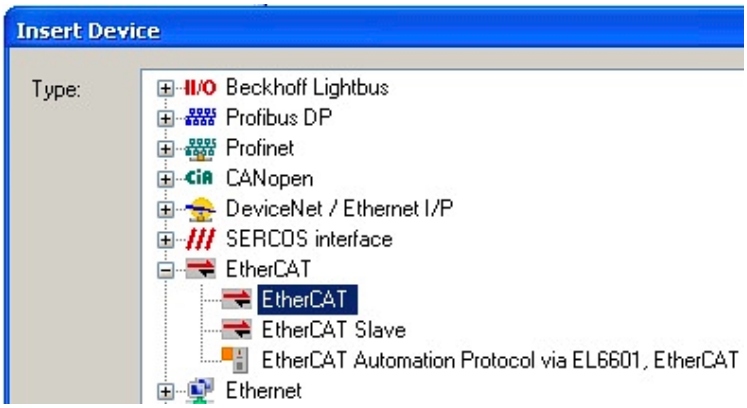


Fig. 114: Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3)

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

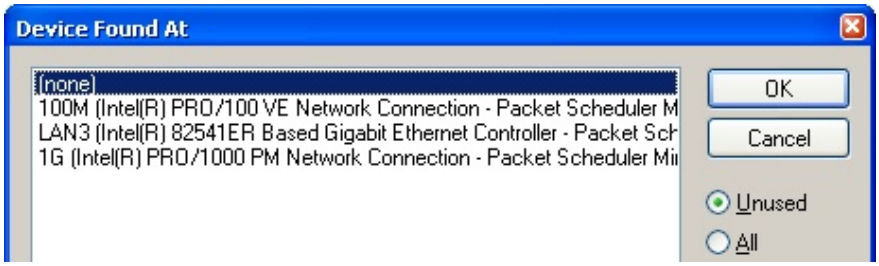


Fig. 115: Selecting the Ethernet port

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

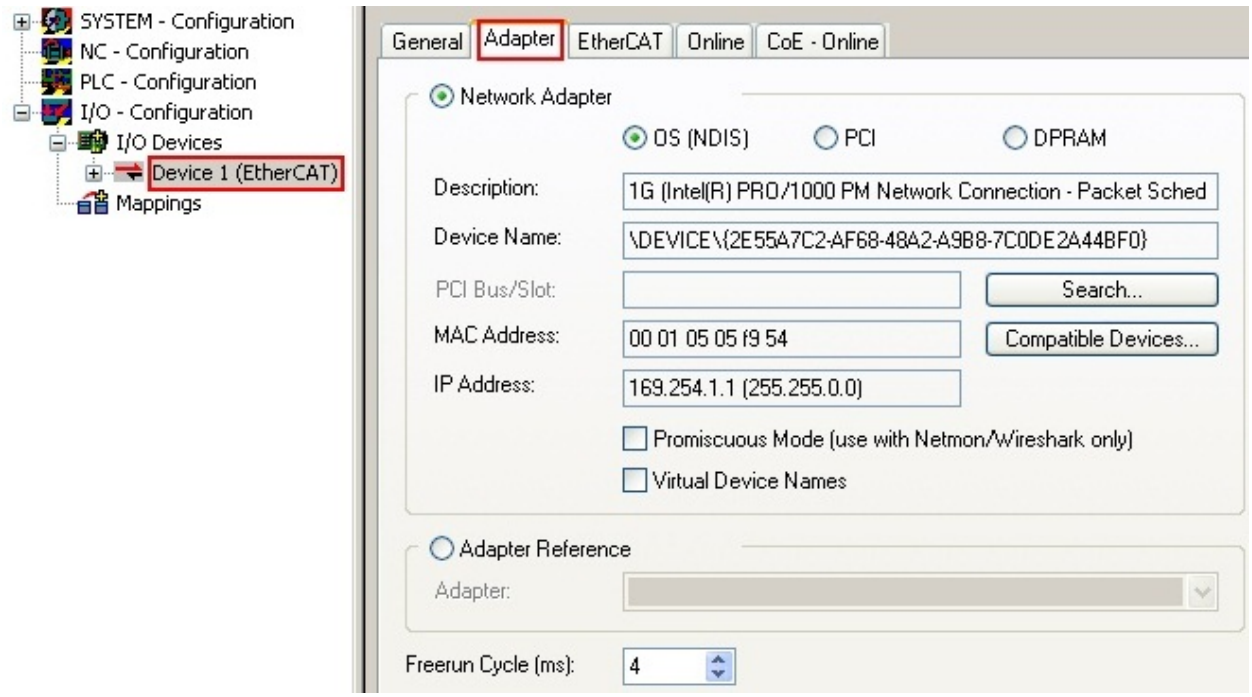
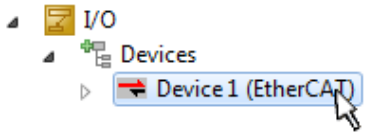


Fig. 116: EtherCAT device properties (TwinCAT 2)

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 [▶ 153].

Defining EtherCAT slaves

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

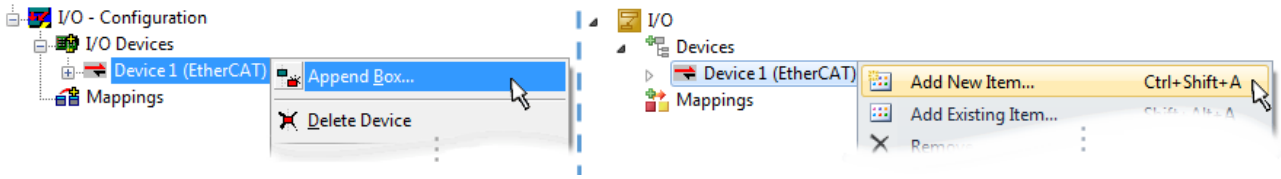


Fig. 117: Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3)

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

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

Overview of physical layer

- “Ethernet”: cable-based 100BASE-TX: couplers, box modules, devices with RJ45/M8/M12 connector
- “E-Bus”: LVDS “terminal bus”, EtherCAT plug-in modules (EJ), EtherCAT terminals (EL/ES), various modular modules

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

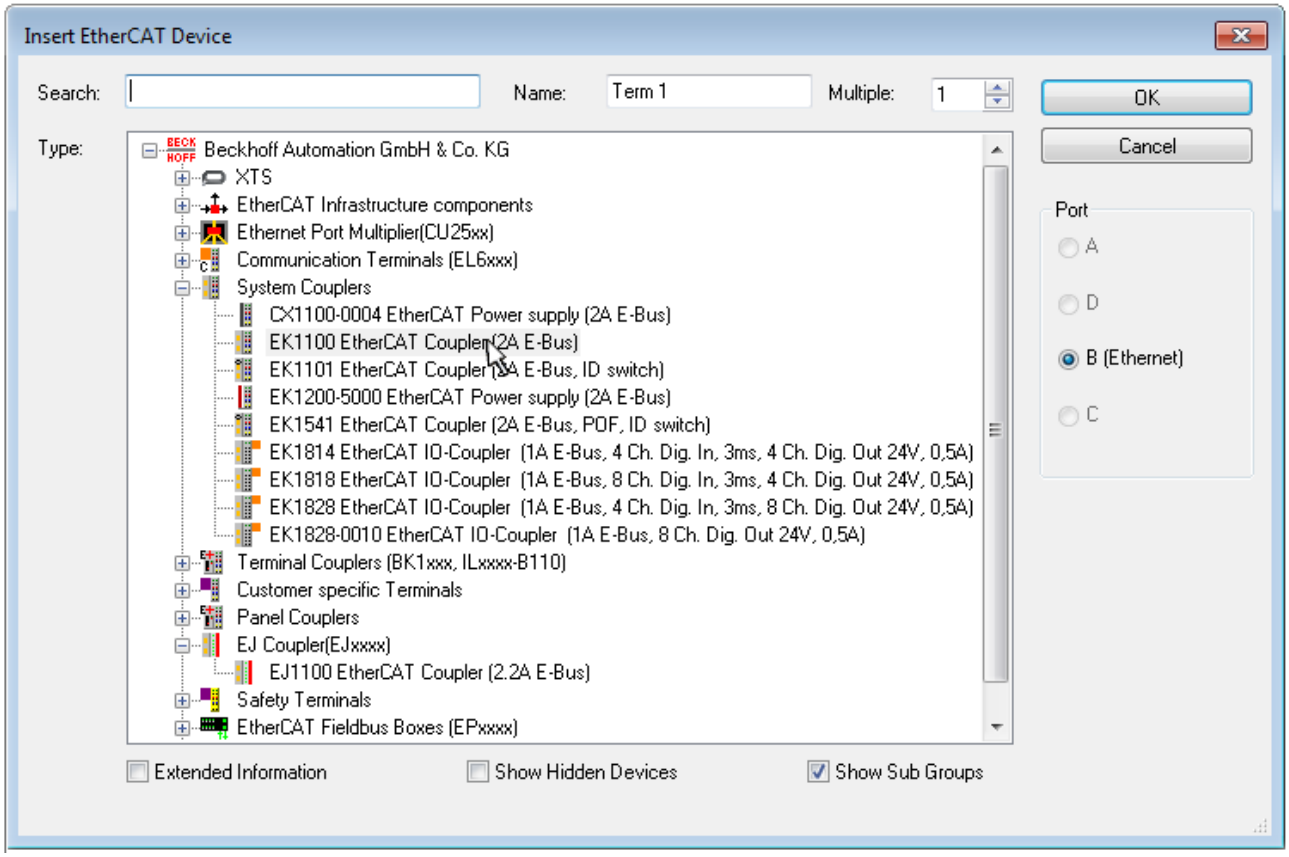


Fig. 118: Selection dialog for new EtherCAT device

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

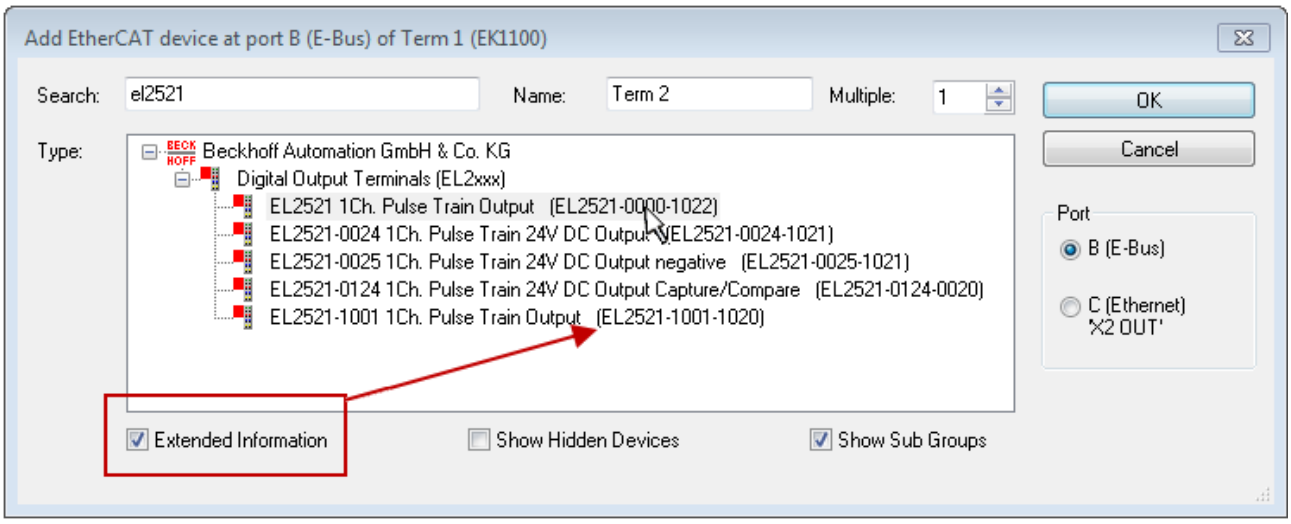


Fig. 119: Display of device revision

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

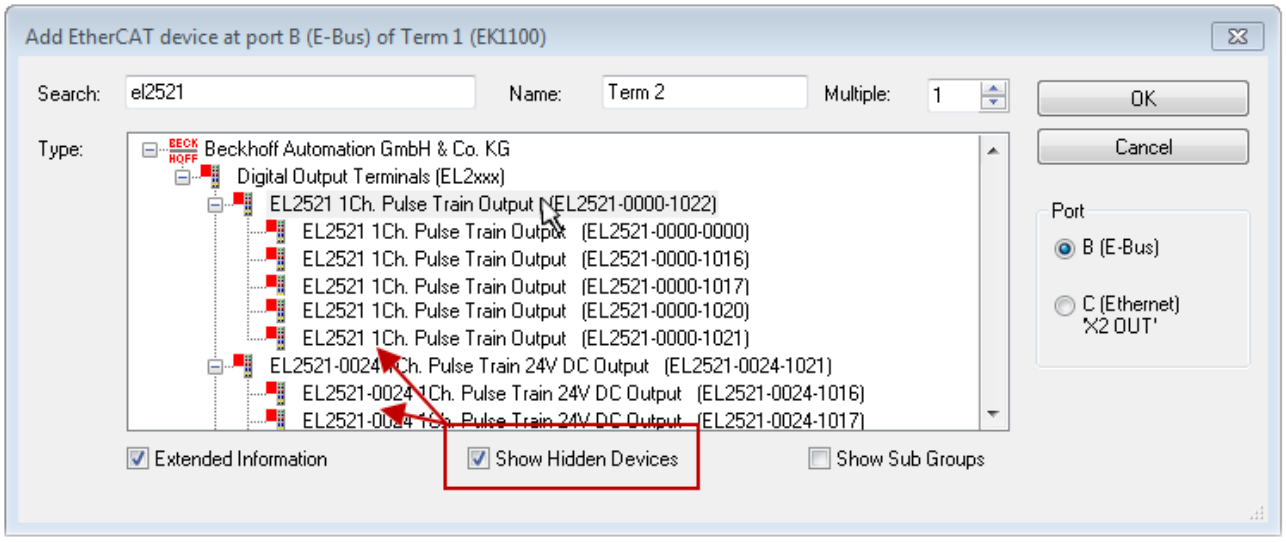


Fig. 120: Display of previous revisions

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.

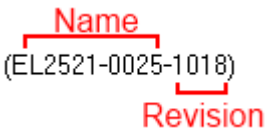


Fig. 121: Name/revision of the terminal

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, ...

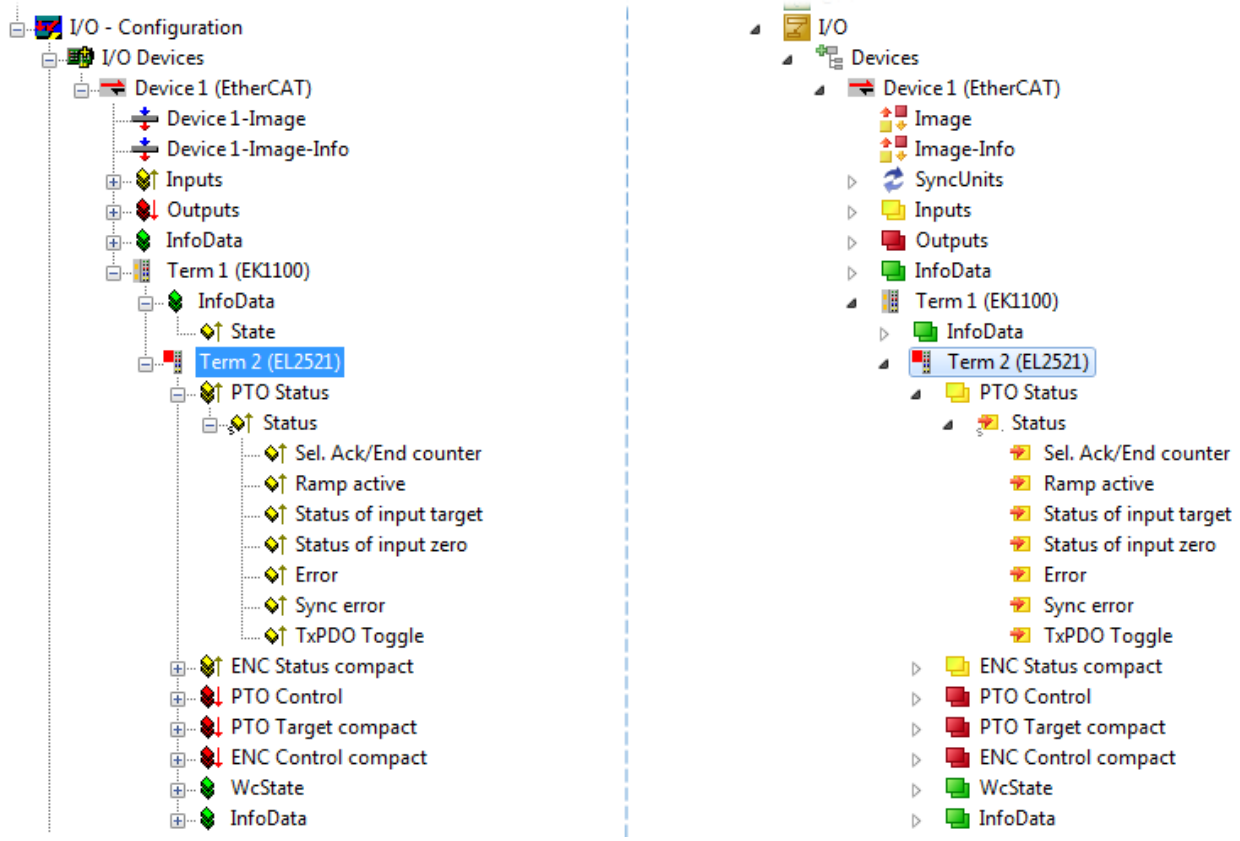


Fig. 122: EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3)

7.3.6 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.

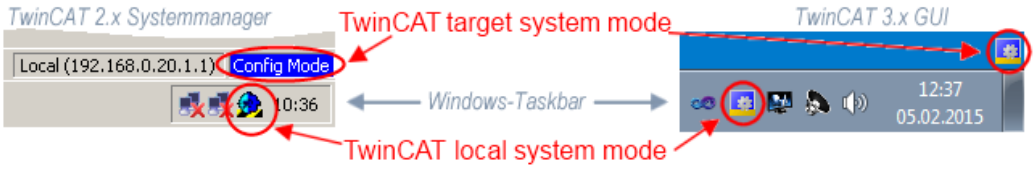


Fig. 123: Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3)

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

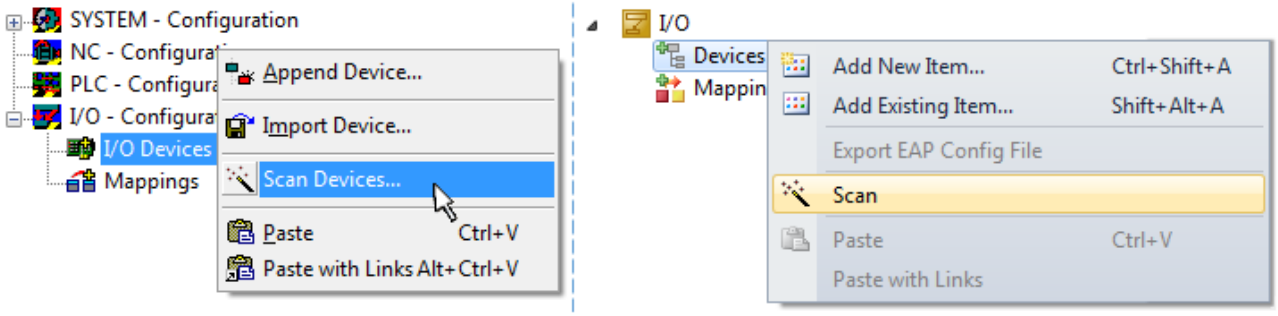


Fig. 124: Scan Devices (left: TwinCAT 2; right: TwinCAT 3)

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

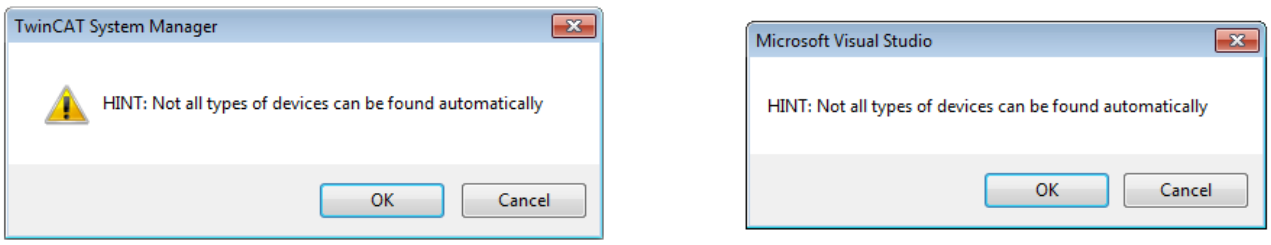


Fig. 125: Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3)

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

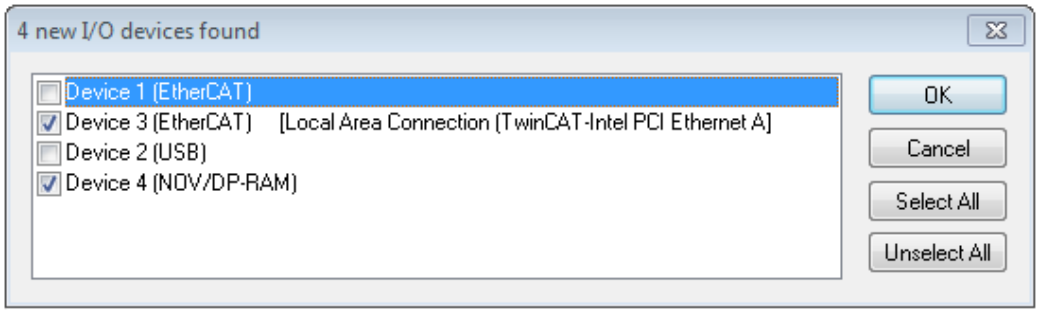


Fig. 126: Detected Ethernet devices

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 [▶ 153].

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.

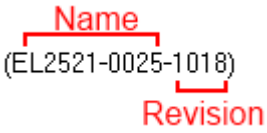


Fig. 127: Example default state

NOTICE

Slave scanning in practice in series machine production

The scanning function should be used with care. It is a practical and fast tool for creating an initial configuration as a basis for commissioning. In series machine production or reproduction of the plant, however, the function should no longer be used for the creation of the configuration, but if necessary for comparison [▶ 176] with the defined initial configuration.Background: since Beckhoff occasionally increases the revision version of the delivered products for product maintenance reasons, a configuration can be created by such a scan which (with an identical machine construction) is identical according to the device list; however, the respective device revision may differ from the initial configuration.

Example:

Company A builds the prototype of a machine B, which is to be produced in series later on. To do this the prototype is built, a scan of the IO devices is performed in TwinCAT and the initial configuration “B.tsm” is created. The EL2521-0025 EtherCAT terminal with the revision 1018 is located somewhere. It is thus built into the TwinCAT configuration in this way:

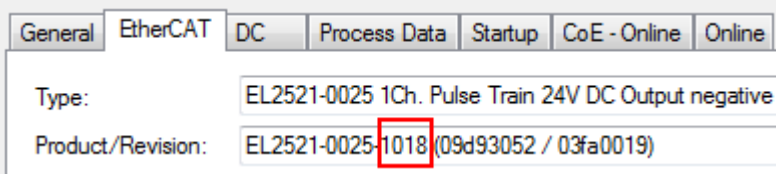


Fig. 128: Installing EthetCAT terminal with revision -1018

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 [▶ 176] against the initial configuration “B.tsm” in order to check the built machine.

However, if the series machine production department now doesn’t use “B.tsm”, but instead carries out a scan to create the productive configuration, the revision **-1019** is automatically detected and built into the configuration:

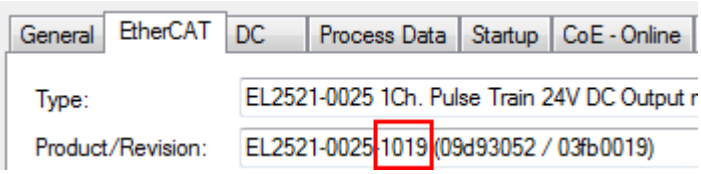


Fig. 129: Detection of EtherCAT terminal with revision -1019

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

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

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

Fig. 130: Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

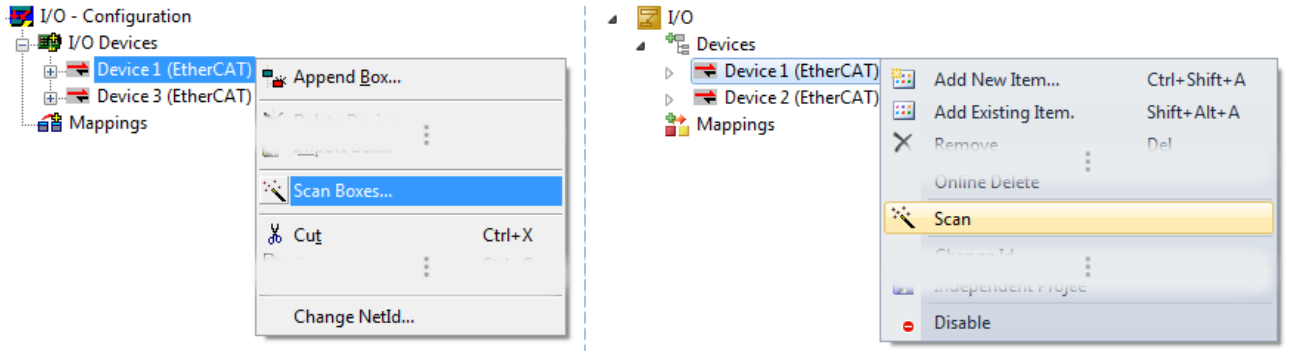


Fig. 131: Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3)

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

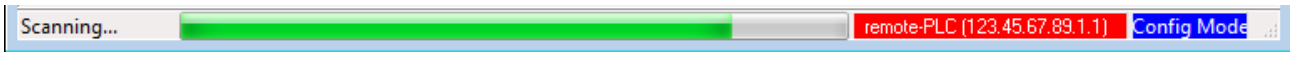


Fig. 132: *Scan progressexemplary by TwinCAT 2*

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

Fig. 133: 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).

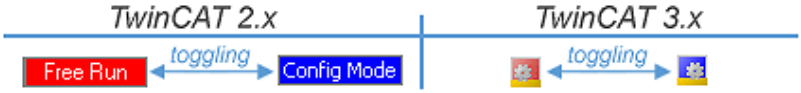


Fig. 134: Displaying of “Free Run” and “Config Mode” toggling right below in the status bar

Fig. 135: TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3)

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

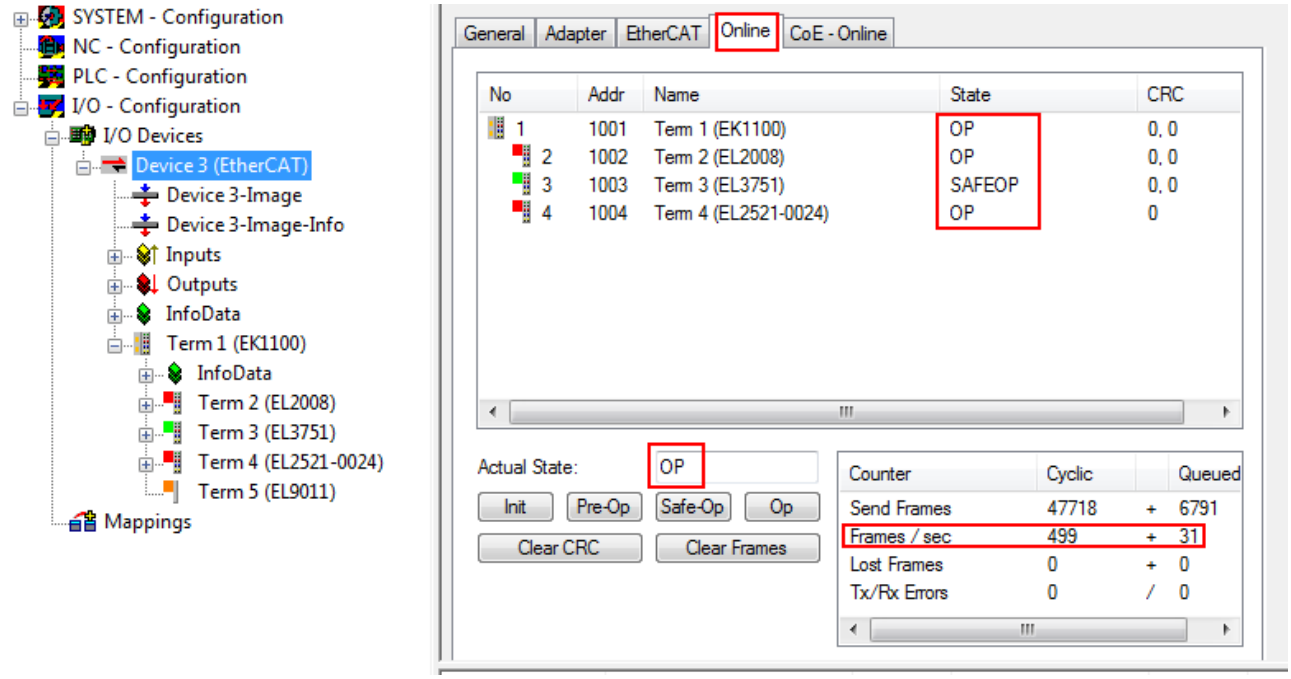


Fig. 136: Online display example

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 [▶ 165].

Troubleshooting

Various effects may occur during scanning.

- An **unknown device** is detected, i.e. an EtherCAT slave for which no ESI XML description is available. In this case the System Manager offers to read any ESI that may be stored in the device. This case is described in the chapter “Notes regarding ESI device description”.
- **Device are not detected properly**
  Possible reasons include:
  - faulty data links, resulting in data loss during the scan
  - slave has invalid device description

  The connections and devices should be checked in a targeted manner, e.g. via the emergency scan. Then re-run the scan.

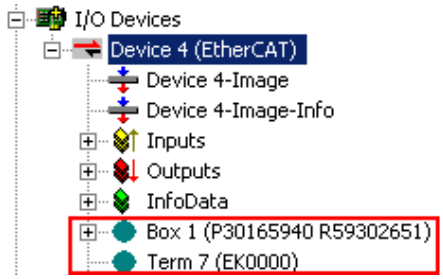


Fig. 137: Faulty identification

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

Scan over existing Configuration

NOTICE

Change of the configuration after comparison

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

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

Fig. 138: Identical configuration (left: TwinCAT 2; right: TwinCAT 3)

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

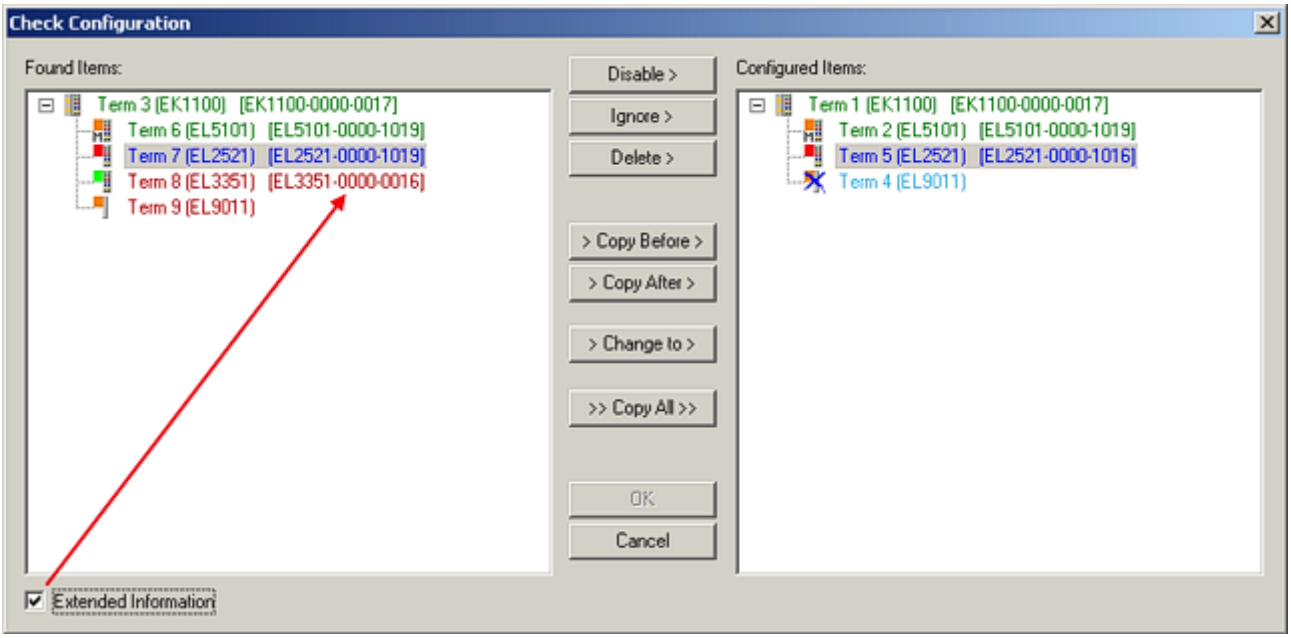


Fig. 139: Correction dialog

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)

Color	Explanation
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.

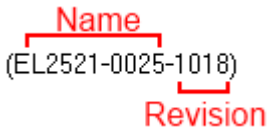


Fig. 140: Name/revision of the terminal

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, ...

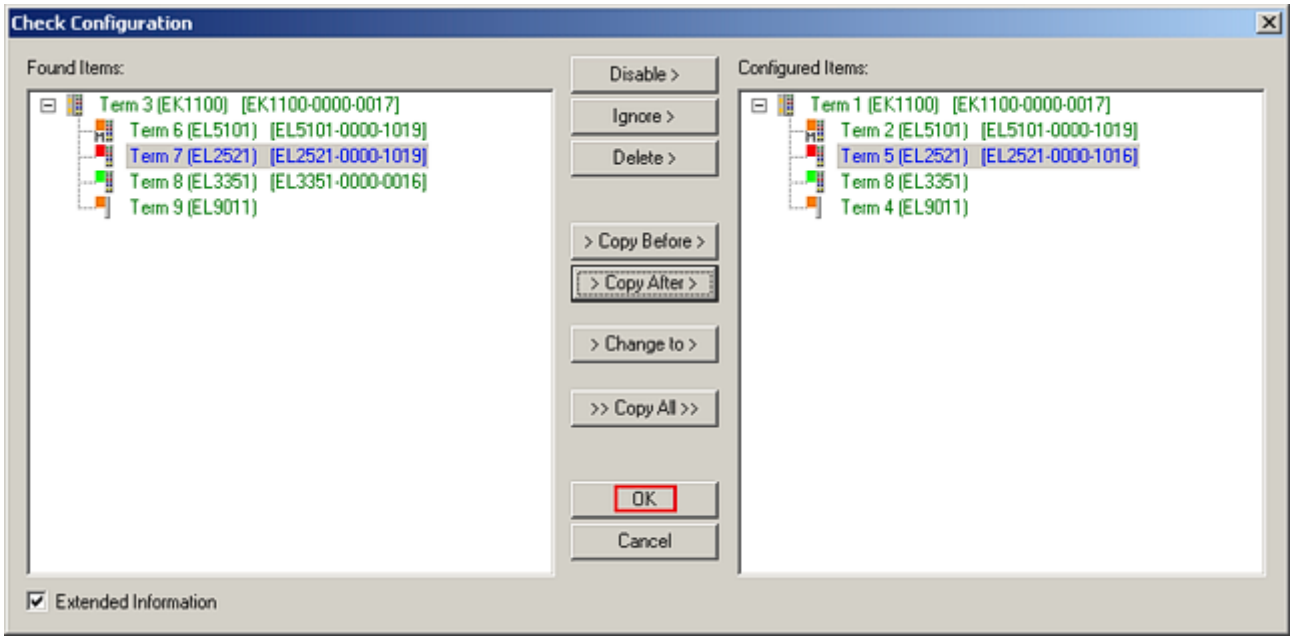


Fig. 141: Correction dialog with modifications

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.

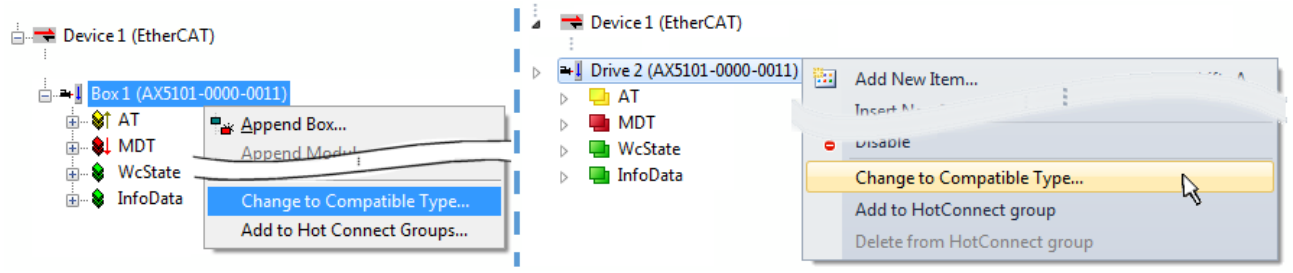


Fig. 142: Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3)

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

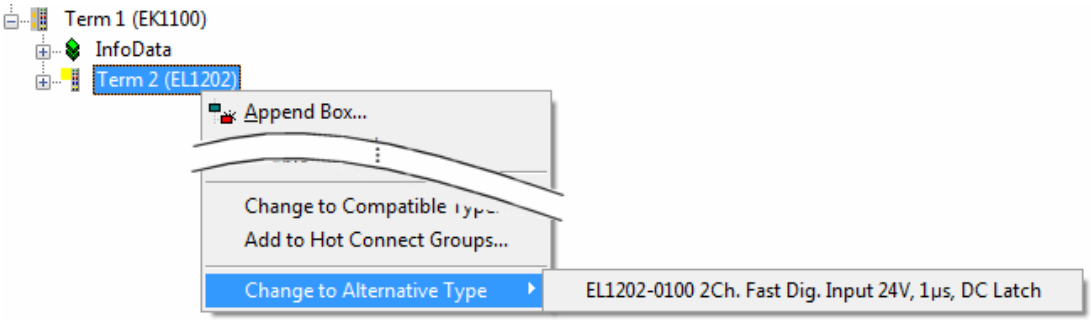


Fig. 143: TwinCAT 2 Dialog Change to Alternative Type

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).

7.3.7 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).

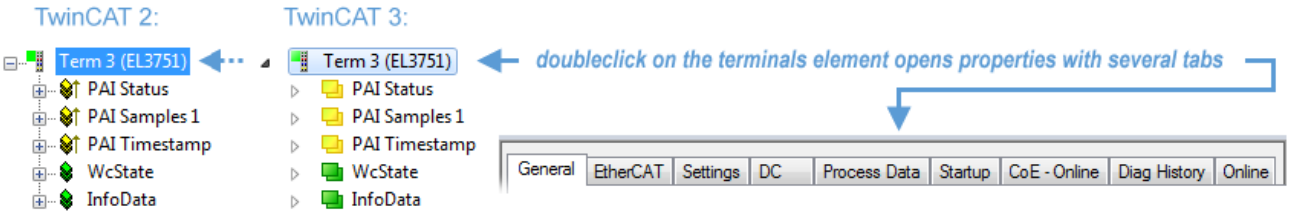


Fig. 144: Branch element as terminal EL3751

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

“General” tab

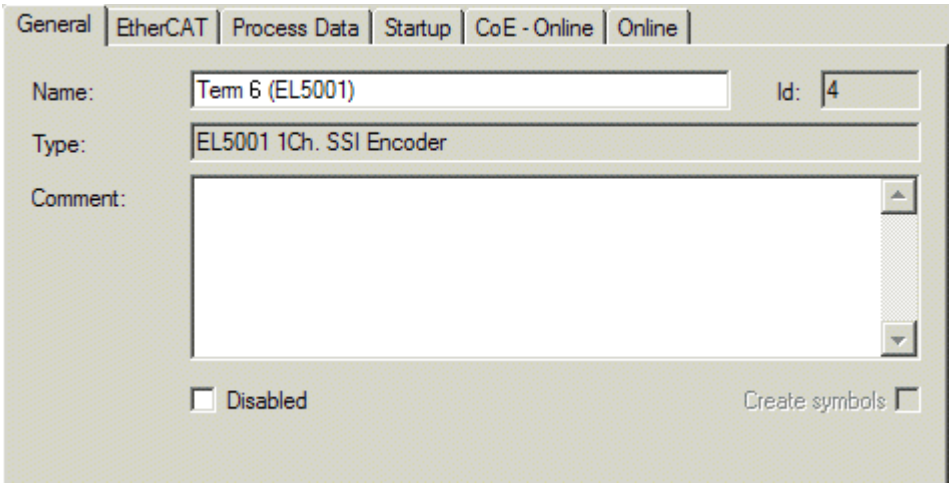


Fig. 145: “General” tab

Name	Name of the EtherCAT device

Id	Number of the EtherCAT device
Type	EtherCAT device type
Comment	Here you can add a comment (e.g. regarding the system).
Disabled	Here you can deactivate the EtherCAT device.
Create symbols	Access to this EtherCAT slave via ADS is only available if this control box is activated.

“EtherCAT” tab

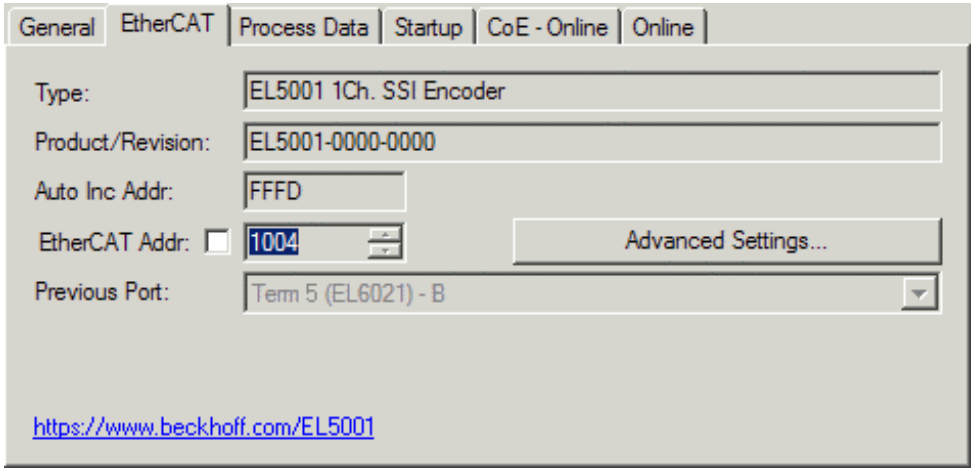


Fig. 146: “EtherCAT” tab

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.

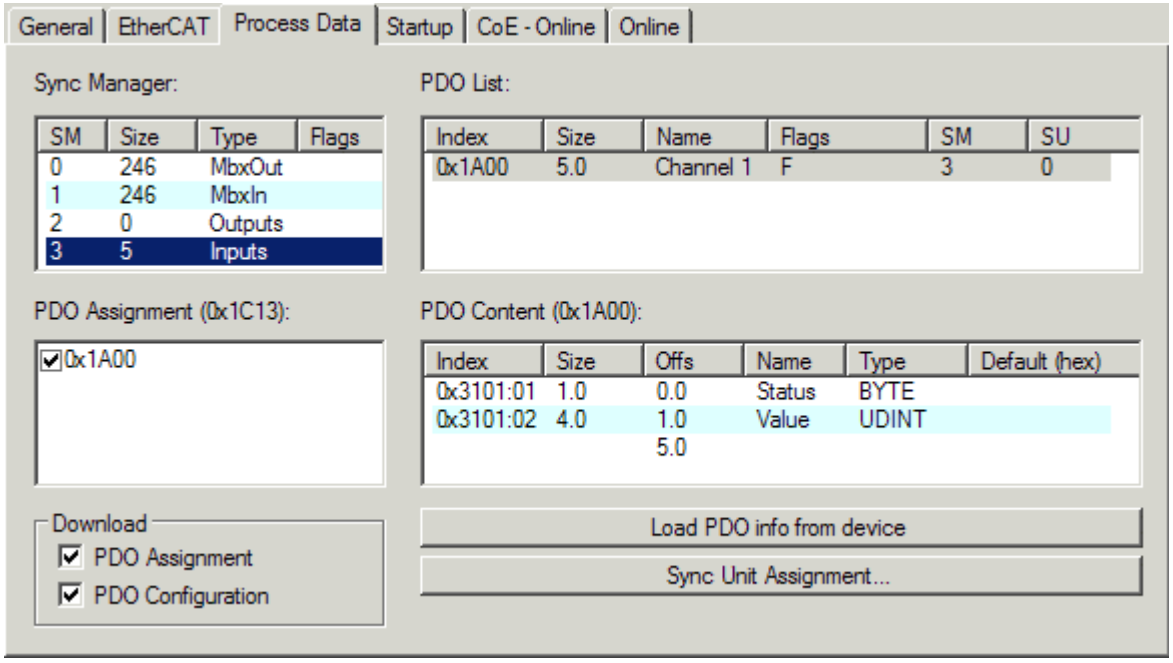


Fig. 147: “Process Data” tab

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

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

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

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

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

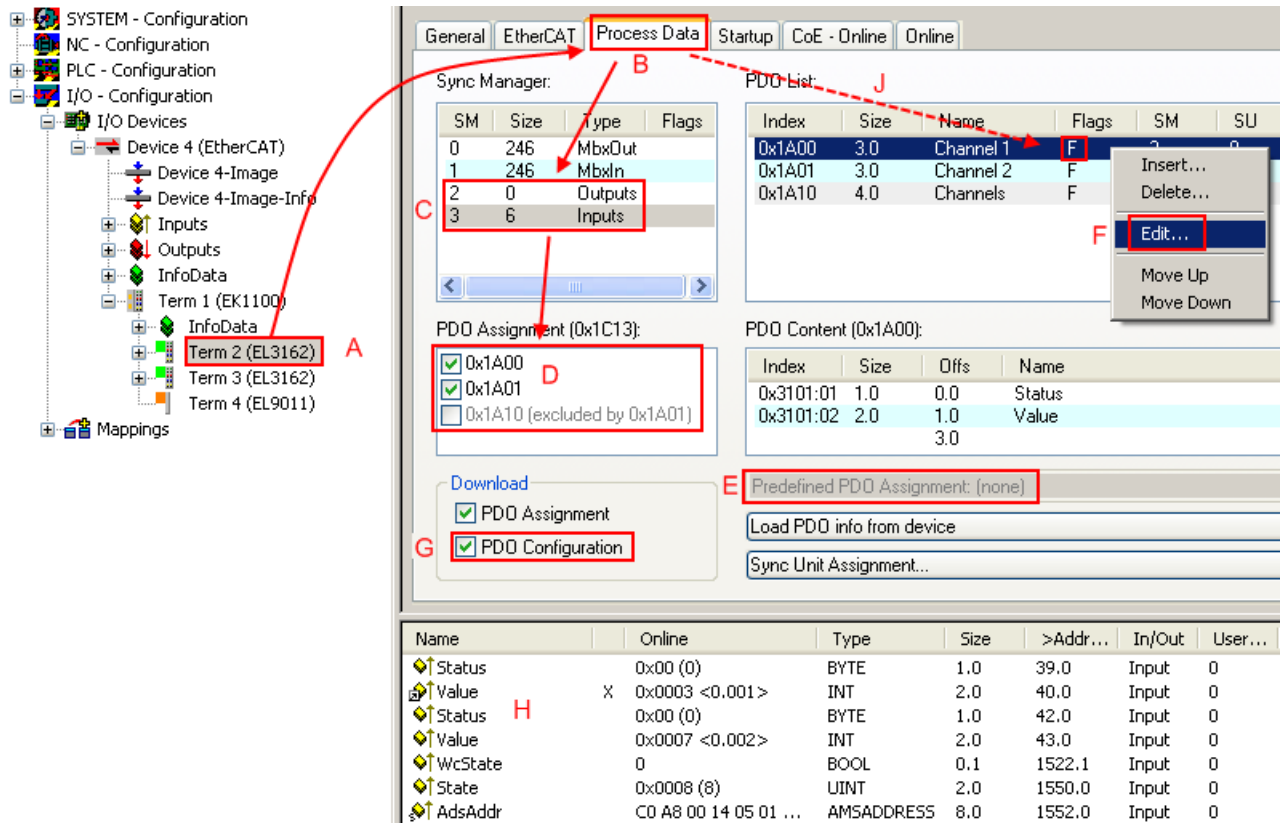


Fig. 148: Configuring the process data

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 [▶ 187] can be found at the end of this section.

“Startup” tab

The *Startup* tab is displayed if the EtherCAT slave has a mailbox and supports the *CANopen over EtherCAT* (CoE) or *Servo drive over EtherCAT* protocol. This tab indicates which download requests are sent to the mailbox during startup. It is also possible to add new mailbox requests to the list display. The download requests are sent to the slave in the same order as they are shown in the list.

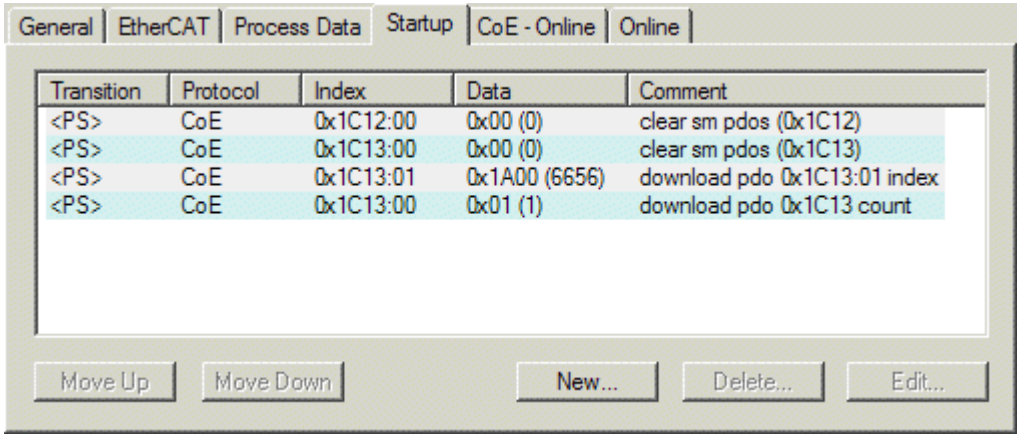


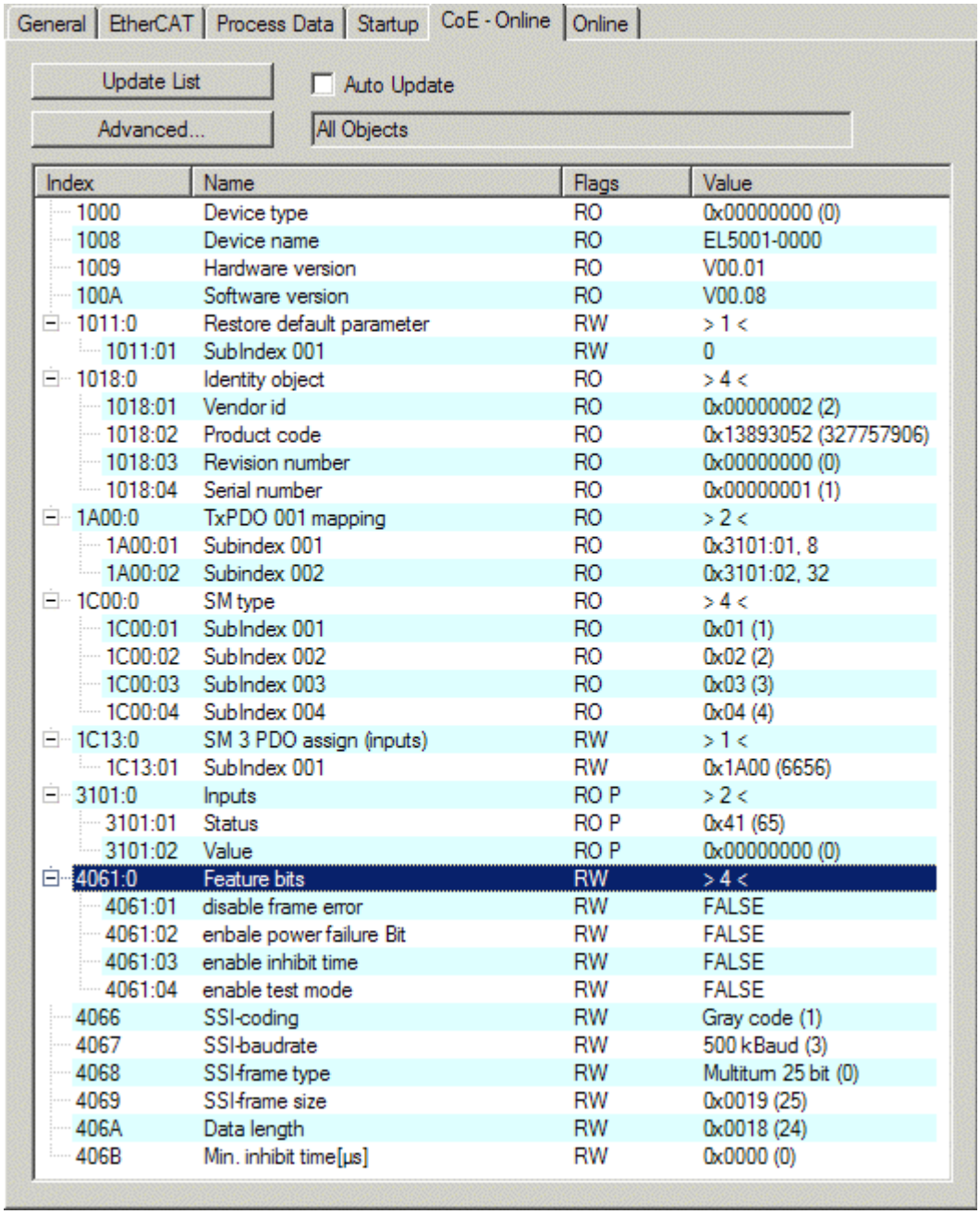
Fig. 149: “Startup” tab

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	Multiturn 25 bit (0)
4069	SSI-frame size	RW	0x0019 (25)
406A	Data length	RW	0x0018 (24)
406B	Min. inhibit time[µs]	RW	0x0000 (0)

Fig. 150: “CoE - Online” tab

Object list display

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

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

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

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

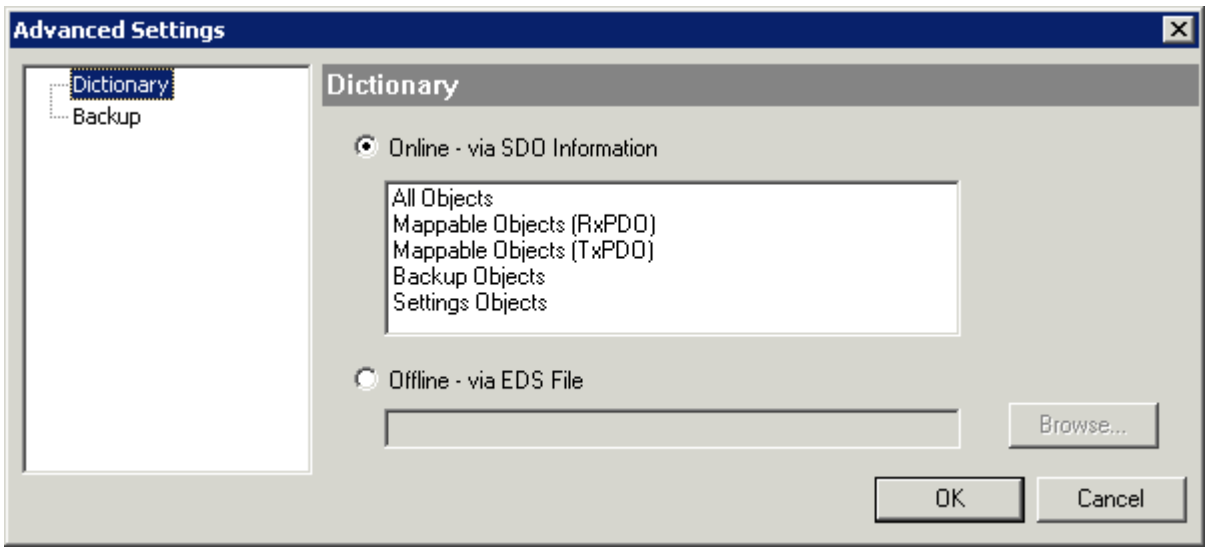


Fig. 151: Dialog “Advanced settings”

Online - via SDO Information	If this option button is selected, the list of the objects included in the object list of the slave is uploaded from the slave via SDO information. The list below can be used to specify which object types are to be uploaded.
Offline - via EDS File	If this option button is selected, the list of the objects included in the object list is read from an EDS file provided by the user.

“Online” tab

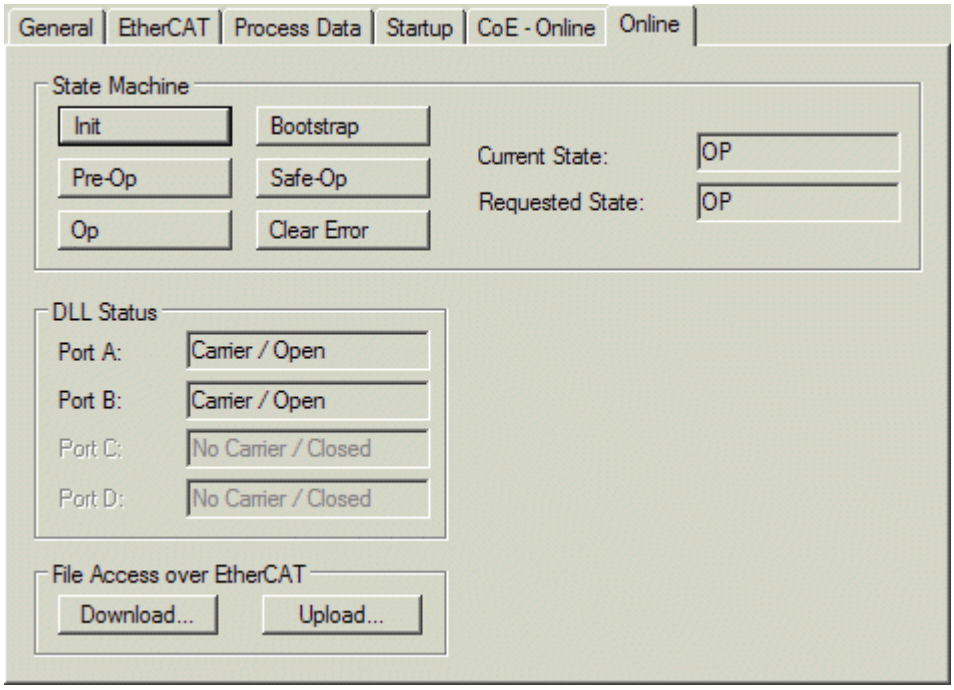


Fig. 152: “Online” tab

State Machine

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

Clear Error	This button attempts to delete the fault display. If an EtherCAT slave fails during change of state it sets an error flag. Example: An EtherCAT slave is in PREOP state (pre-operational). The master now requests the SAFEOP state (safe-operational). If the slave fails during change of state it sets the error flag. The current state is now displayed as ERR PREOP. When the *Clear Error* button is pressed the error flag is cleared, and the current state is displayed as PREOP again.
Current State	Indicates the current state of the EtherCAT device.
Requested State	Indicates the state requested for the EtherCAT device.

DLL Status

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

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

File Access over EtherCAT

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

“DC” tab (Distributed Clocks)

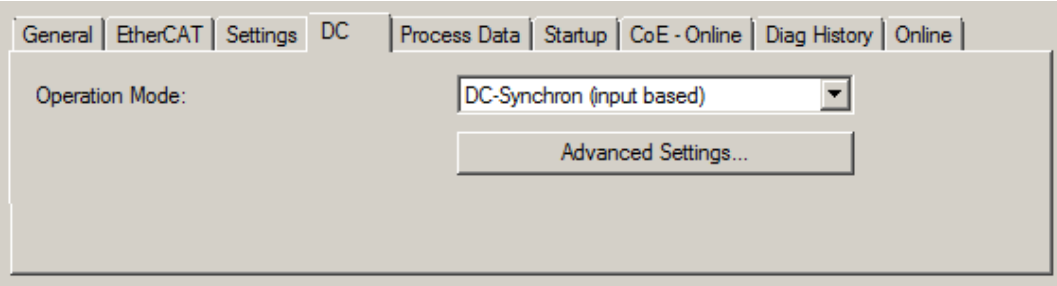


Fig. 153: “DC” tab (Distributed Clocks)

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

7.3.7.1 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 [▶ 185]),

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 [▶ 182] 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.

7.3.8 Import/Export of EtherCAT devices with SCI and XTI

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

7.3.8.1 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.

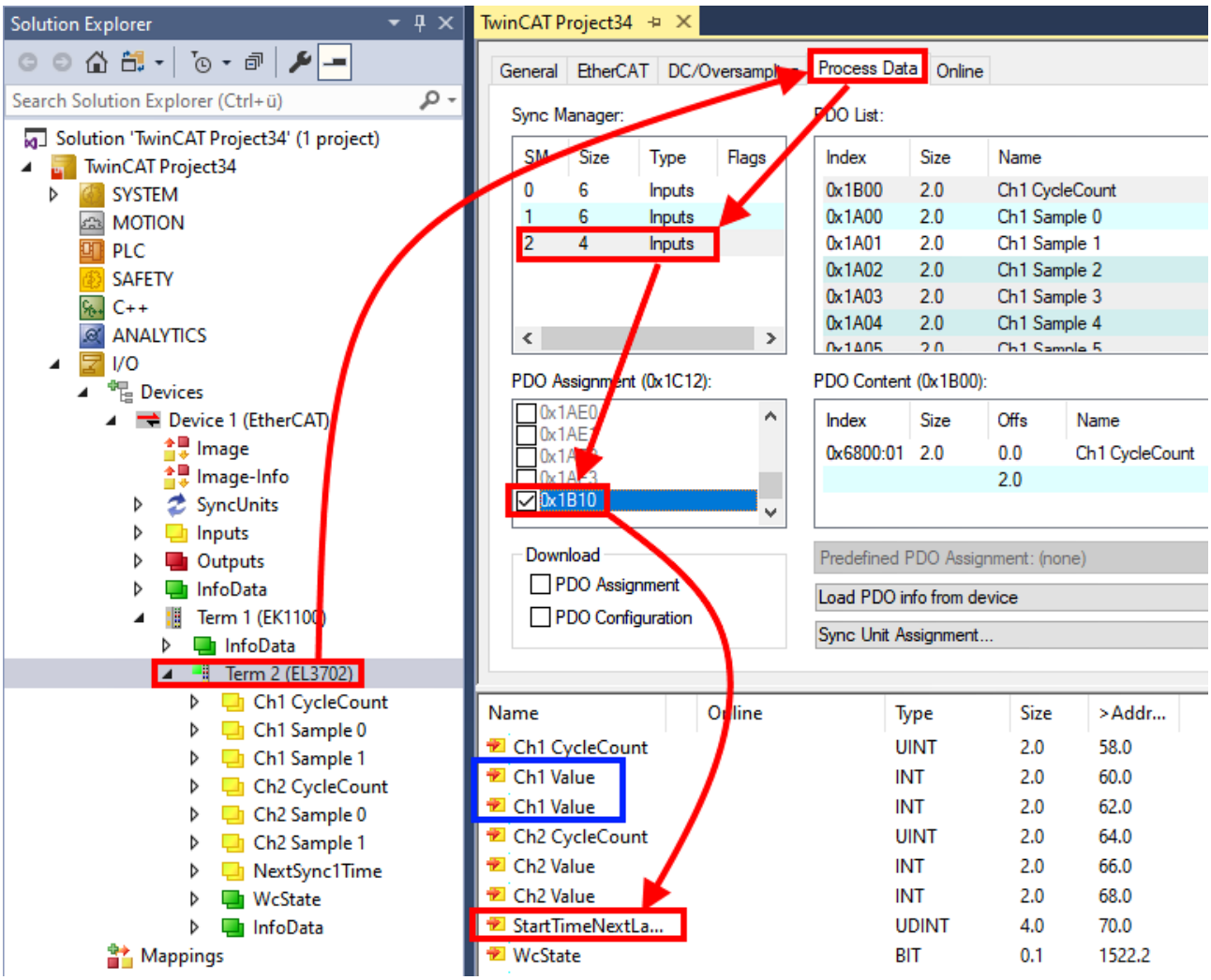
It may become necessary to reuse the modified slave in other projects in this way, without having to make equivalent configuration changes to the slave again. To accomplish this, proceed as follows:

- 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.

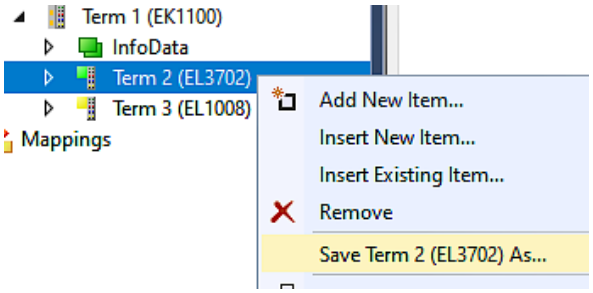
An example is provided below for illustration purposes: an EL3702 terminal with standard setting is switched to 2-fold oversampling (blue) and the optional PDO "StartTimeNextLatch" is added (red):



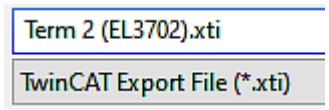
The two methods for exporting and importing the modified terminal referred to above are demonstrated below.

7.3.8.2 Procedure within TwinCAT with xti files

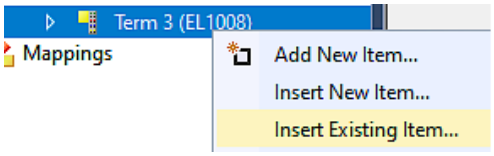
Each IO device can be exported/saved individually:



The xti file can be stored:



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



7.3.8.3 Procedure within and outside TwinCAT with sci file

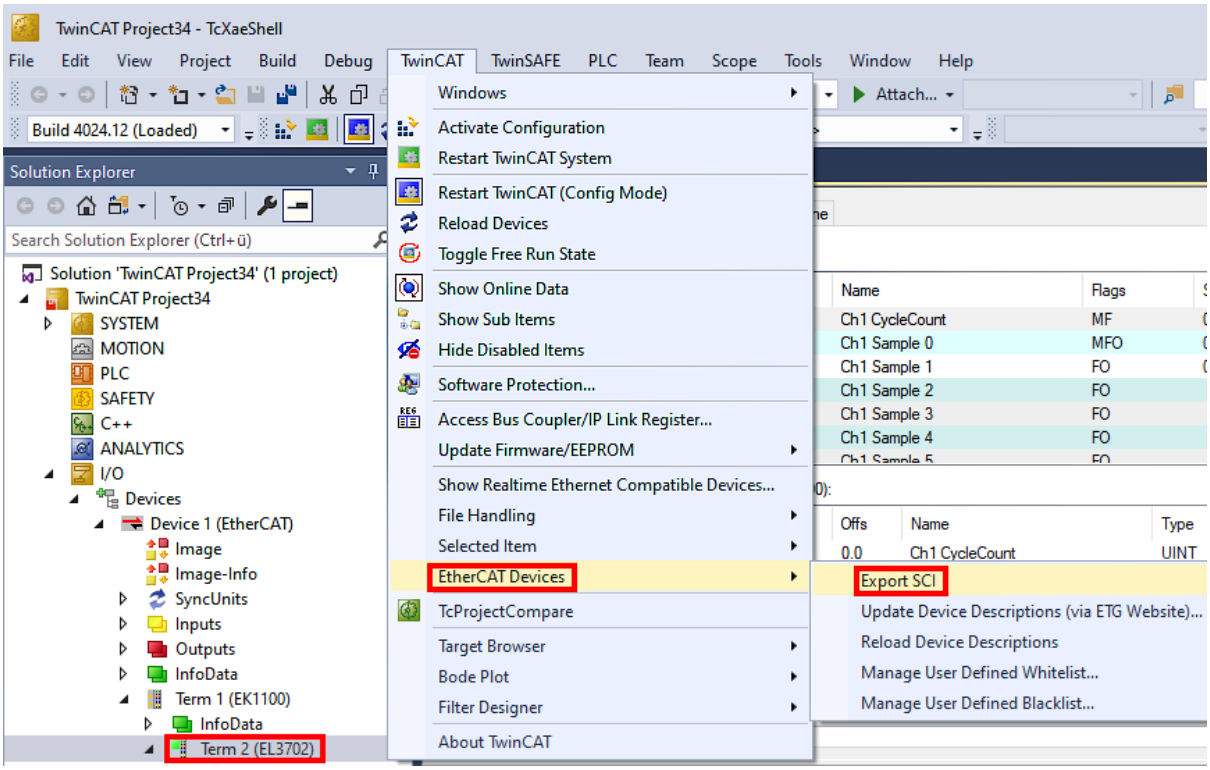
Note regarding availability (2021/01)

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.

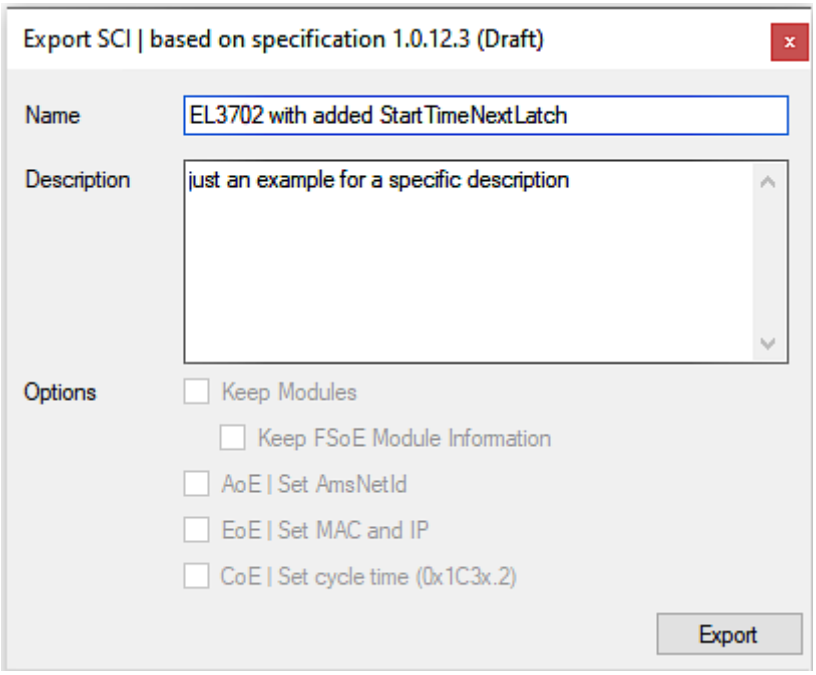
Export:

- 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:

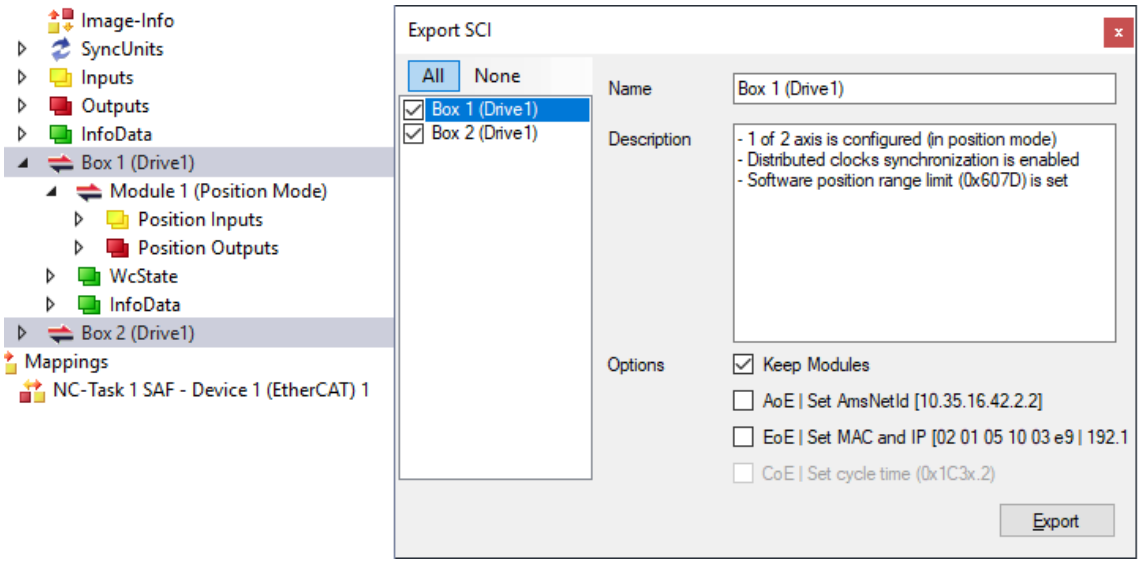
- A description may also be provided:



- Explanation of the dialog box:

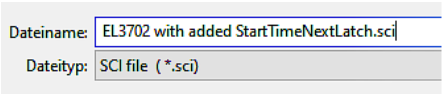
Name		Name of the SCI, assigned by the user.
Description		Description of the slave configuration for the use case, assigned by the user.
Options	Keep modules	If a slave supports modules/slots, the user can decide whether these are to be exported or whether the module and device data are to be combined 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)*:

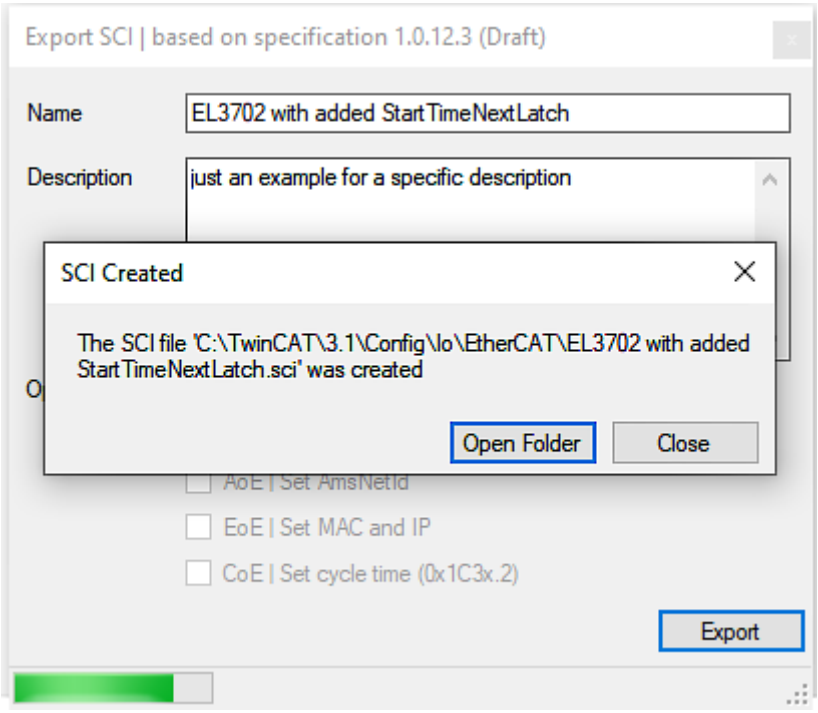


- Selection of the slaves to be exported:
  - All:
    All slaves are selected for export.

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

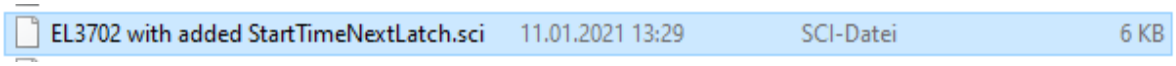


- The export takes place:

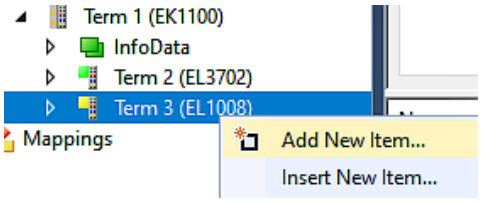


Import

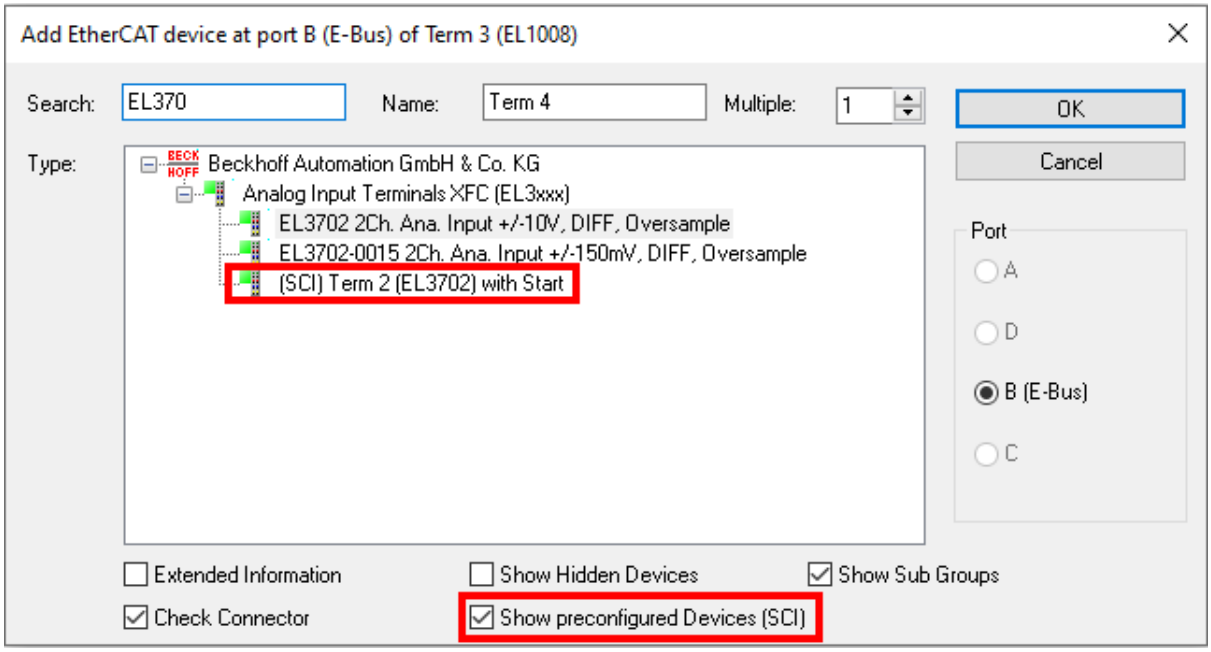
- An sci description can be inserted manually into the TwinCAT configuration like any normal Beckhoff device description.
- The sci file must be located in the TwinCAT ESI path, usually under:
  C:\TwinCAT\3.1\Config\Io\EtherCAT



- Open the selection dialog:

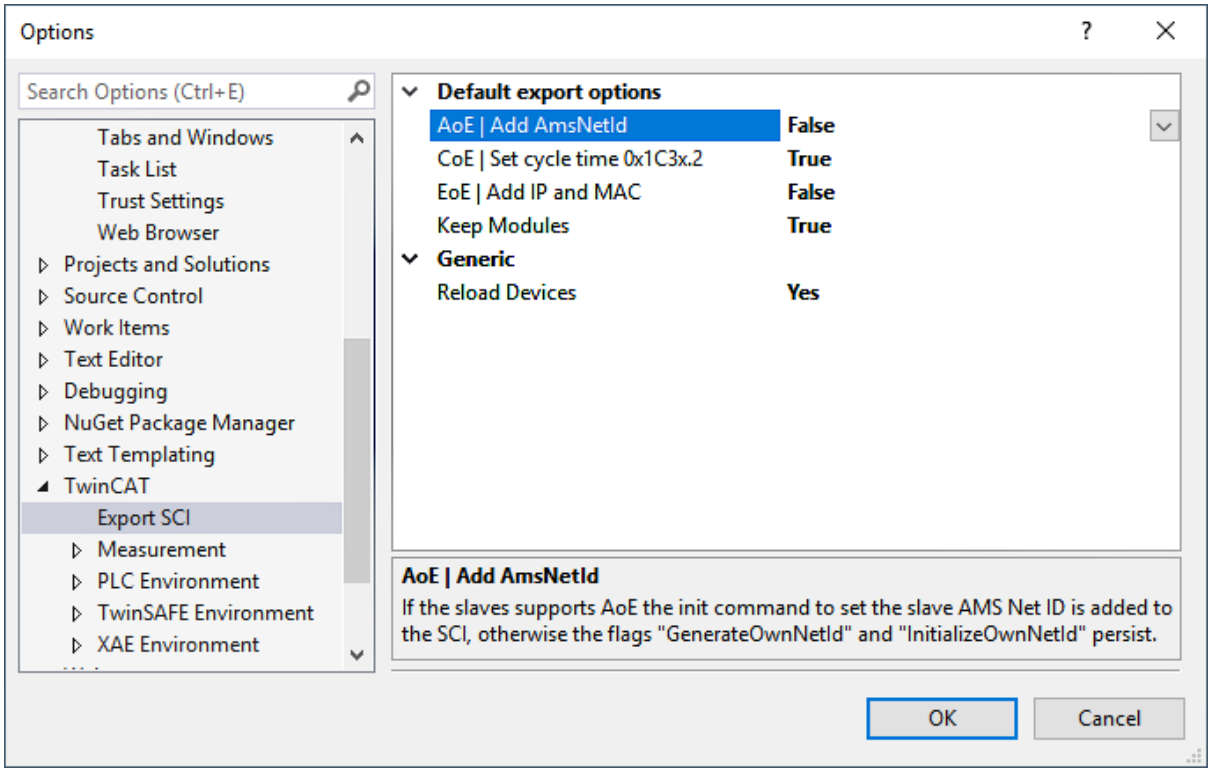


- Display SCI devices and select and insert the desired device:



Additional Notes

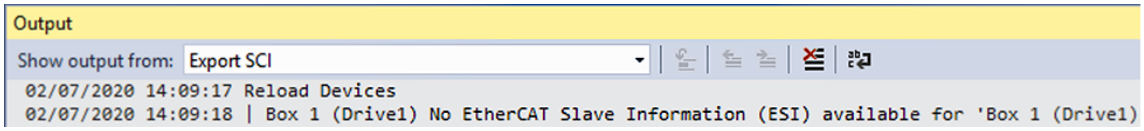
- Settings for the SCI function can be made via the general Options dialog (Tools → Options → TwinCAT → Export SCI):



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:



7.4 EtherCAT basics

Please refer to the EtherCAT System Documentation for the EtherCAT fieldbus basics.

7.5 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.

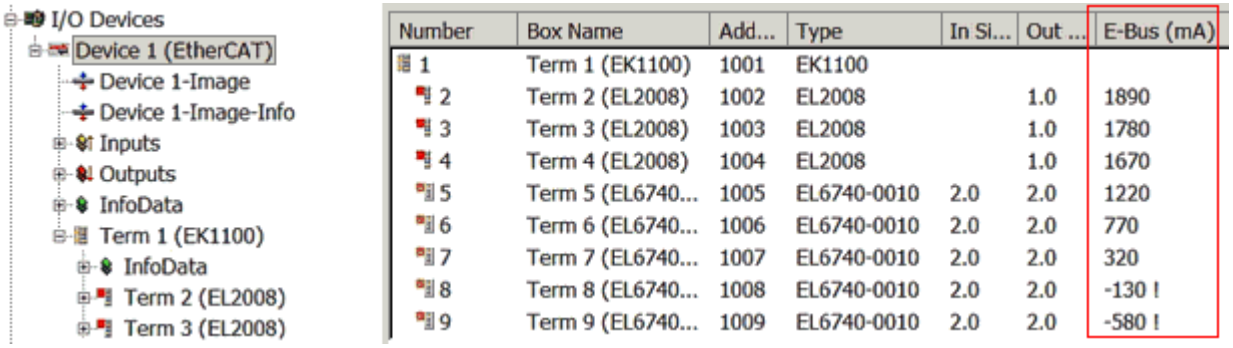


Fig. 154: System manager current calculation

NOTICE

Malfunction possible!

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

7.6 General notes for setting the watchdog

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

The EtherCAT slave controller features two watchdogs:

- Sync Manager (SM) watchdog (default: 100 ms)
- Process Data (PDI) watchdog (default: 100 ms)

Their times are individually parameterized in TwinCAT as follows:

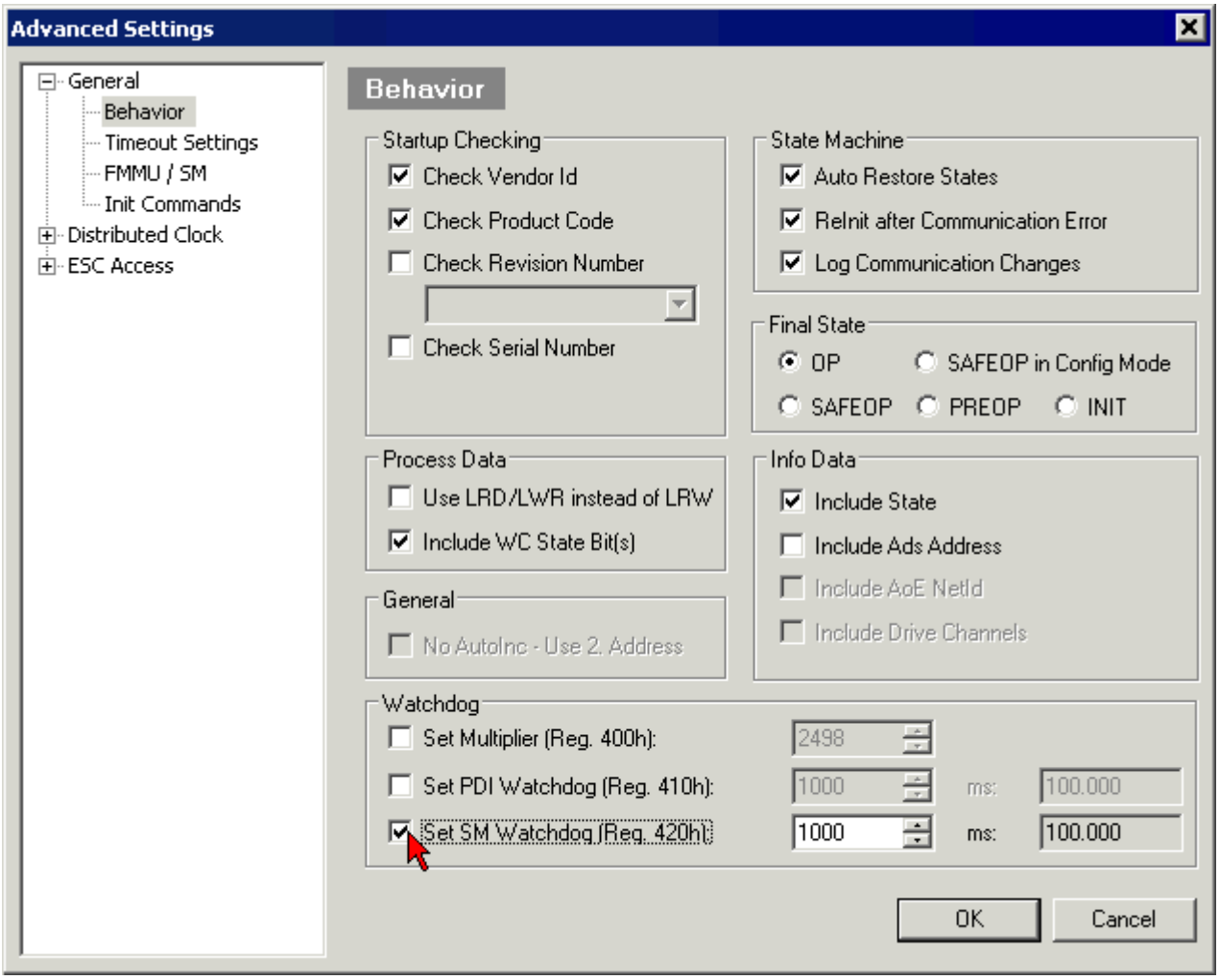


Fig. 155: eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog

Notes:

- the Multiplier Register 400h (hexadecimal, i. e. 0x0400) is valid for both watchdogs.
- each watchdog has its own timer setting 410h or 420h, which together with the Multiplier results in a resulting time.
- important: the Multiplier/Timer setting is only loaded into the slave at EtherCAT startup if the checkbox in front of it is activated.
- if it is not checked, nothing is downloaded and the setting located in the ESC remains unchanged.
- the downloaded values can be seen in the ESC registers 400h, 410h and 420h: ESC Access -> Memory

SM watchdog (SyncManager Watchdog)

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

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

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

PDI watchdog (Process Data Watchdog)

If there is no PDI communication with the ESC for longer than the set and activated Process Data Interface (PDI) watchdog time, this watchdog is triggered.

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

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

Calculation

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

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

The value in “Watchdog multiplier + 2” in the formula above corresponds to the number of 40ns base ticks representing one watchdog tick.

⚠ CAUTION

Undefined state possible!

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

7.7 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
- Operational
- Bootstrap

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

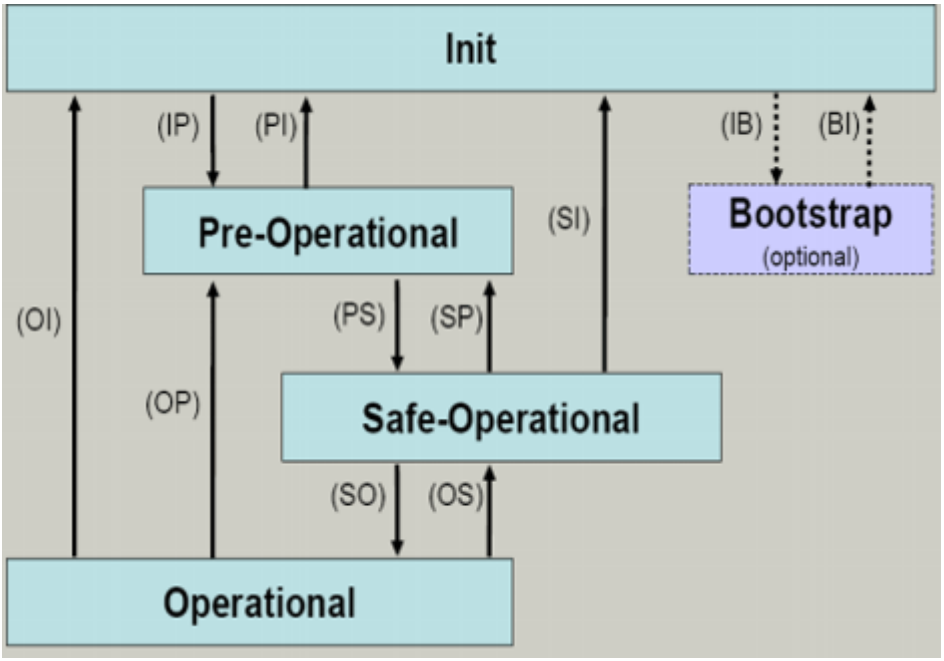


Fig. 156: States of the EtherCAT State Machine

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 Fieldbus Memory Management Unit (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 Dual Port (DP)-RAM areas of the ESC.

In *Safe-Op* state mailbox and process data communication is possible, although the slave keeps its outputs in a safe state, while the input data are updated cyclically.

Outputs in SAFEOP state

The default set watchdog monitoring sets the outputs of the ESC 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 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.

7.8 CoE Interface

General description

The CoE interface (CAN application protocol over EtherCAT interface) 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 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 data 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 (“inputs” from the perspective of the EtherCAT master)

- 0x7000: Output PDOs (“outputs” from the perspective of the EtherCAT master)

Availability

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

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

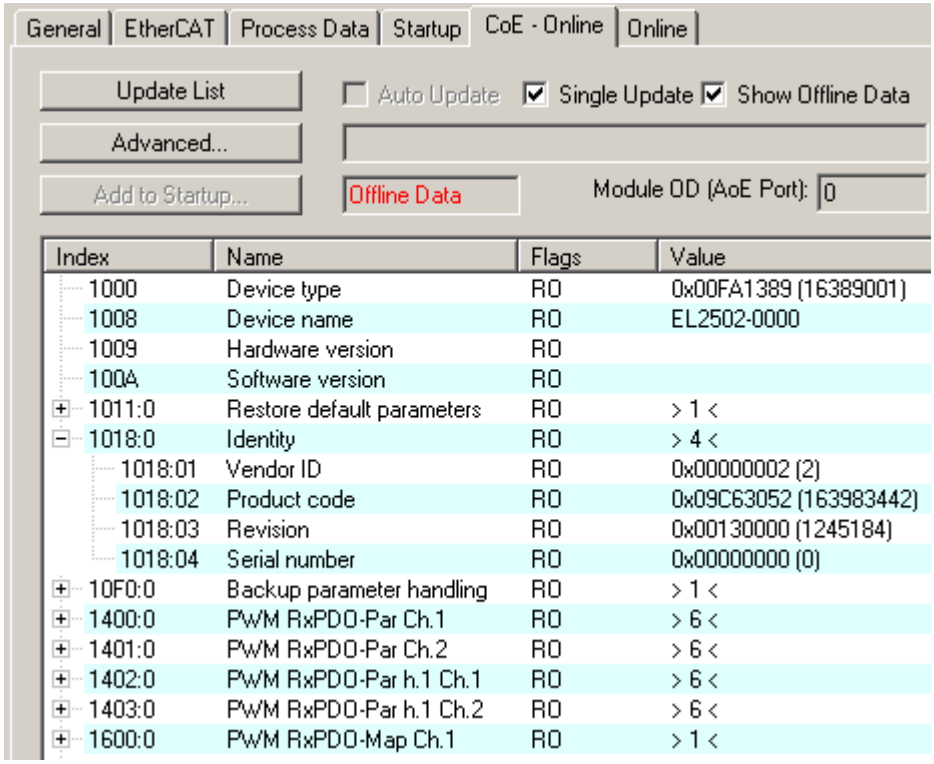


Fig. 157: “CoE Online” tab

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

NOTICE

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

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

- Keep a startup list if components have to be replaced,
- Distinction between online/offline dictionary,
- Existence of current XML description (download from the Beckhoff website),
- "CoE-Reload" for resetting the changes
- Program access during operation via PLC (see TwinCAT 3 | PLC Library: "Tc2_EtherCAT" and Example program R/W CoE)

Data management and function “NoCoeStorage”

Some parameters, particularly the setting parameters of the slave, are configurable and writeable,

- via the System Manager (Fig. “CoE Online” tab) by clicking.
  This is useful for commissioning of the system or slaves. Click on the row of the index to be parameterized and enter a value in the “SetValue” dialog.

- from the control system or 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 index 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.
- If the function is not supported: continuous changing of CoE values is not permissible in view of the lifetime limit.

Startup list

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

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

Recommended approach for manual modification of CoE parameters

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

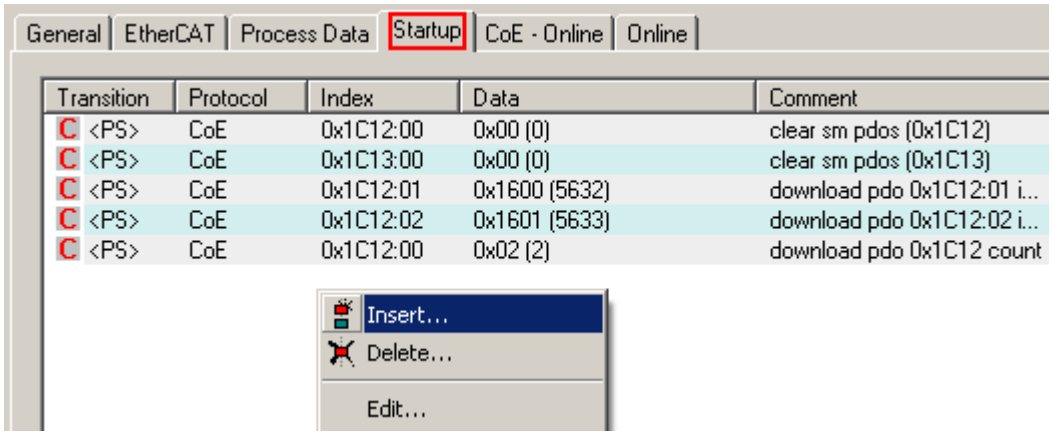


Fig. 158: Startup list in the TwinCAT System Manager

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

Online / offline list

When working with the TwinCAT System Manager, a distinction must be made as to whether the EtherCAT device is currently "available", i.e. switched on and connected via EtherCAT - i.e. **online** - or whether a configuration is created **offline** without slaves being connected.

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 Data** is shown in red.

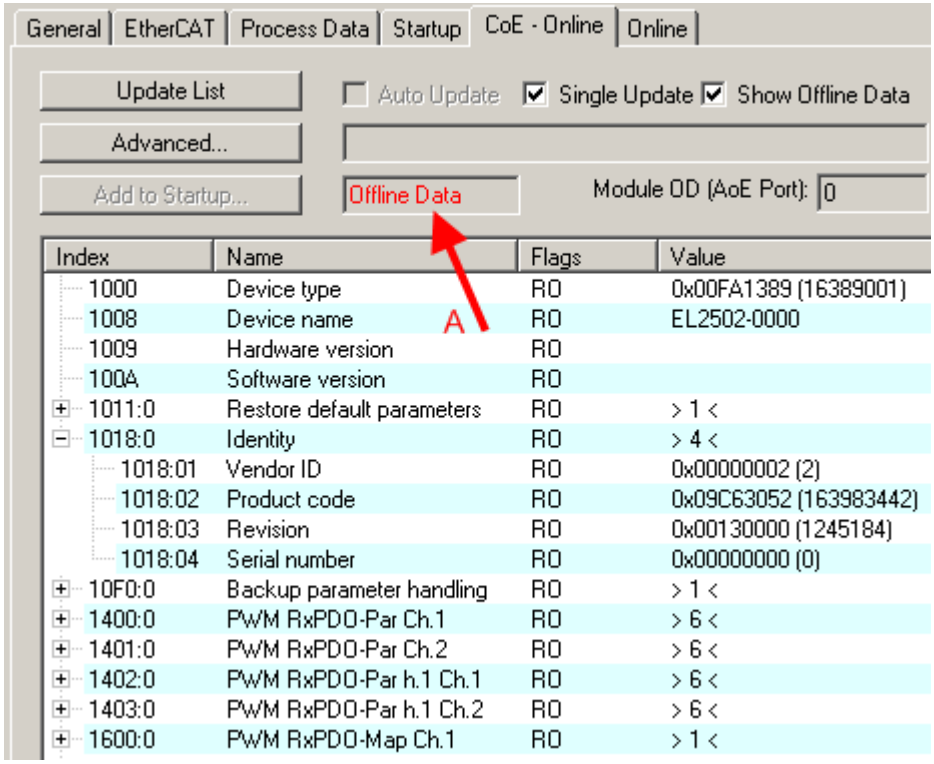


Fig. 159: Offline list

- 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 status of the device is displayed in the CoE.
  - **Online Data** is shown in green.

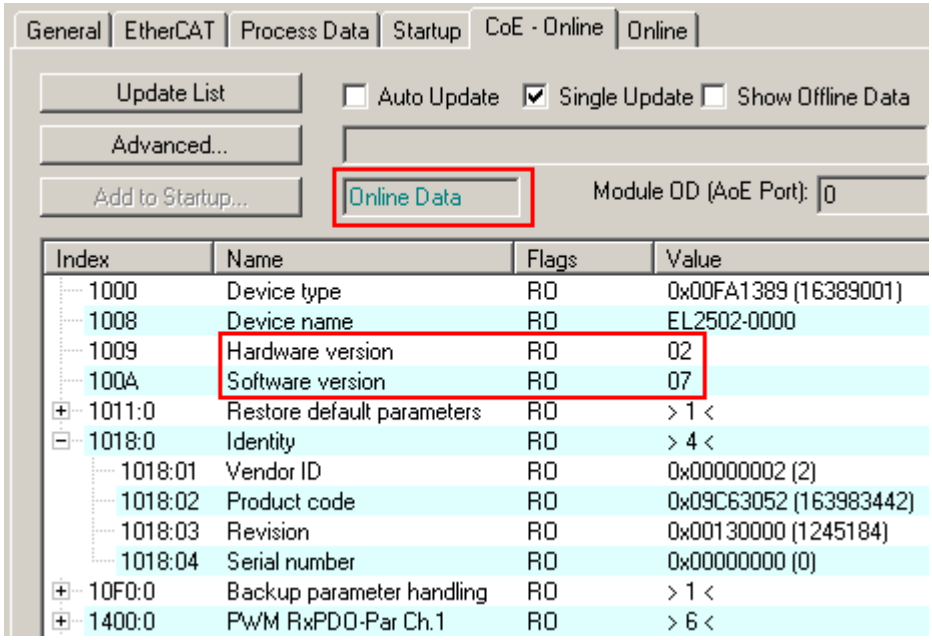


Fig. 160: Online list

Channel-based order

The CoE list is available in EtherCAT devices that usually feature several functionally equivalent channels, for example, a 4-channel analog 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} or 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.

BackUp objects and checksum 0x10F0:01

The following object types, among others, are defined for the CoE parameter/object directory with regard to information retention, but they do not all have to appear in every device at the same time:

- Vendor objects
  - Are stored persistently (fail-safe) in the device.
  - Technically with the property ReadWrite (RW).
  - Can only be modified/deleted if the corresponding vendor password is known.
  - Used for vendor-specific adjustment or identity data.
- BackUp objects
  - These are objects that are stored persistently in the device, even after changes.
  - Technically with the property ReadWrite (RW).
  - They can be modified/deleted at any time via CoE access from the EtherCAT master

  - In particular, they are reset to their initial state, which is stored permanently in the firmware, by selecting "Restore Default Parameters" (see chapter "Restoring the delivery state"). Since this reset to a previous value resembles the restoration of a backup, these are referred to as “BackUp objects”.
  - They are used for regular function parameters of the device that determine the behavior.
- BackUp objects with write protection option.
  - The same applies as for the BackUp objects.
  - In addition, the user can activate write protection for these objects using a code word in xF009, thereby preventing accidental changes. Details can be found in the device documentation of the devices that contain these objects.
- Volatile objects
  - These are objects that are not stored persistently in the device.
  - They are used to display internal information (process data, states, temperatures ...) and are available as ReadOnly (RO) or as function parameters (ReadWrite). However, the latter must be written by the EtherCAT master for each PowerOn if they are to have a value other than the default value.

The device indicates a 16-bit CRC in the 32-bit object 0x10F0:01 CheckSum, subindex 01 of Backup Parameter Handling by means of the current state of the so-called BackUp objects:

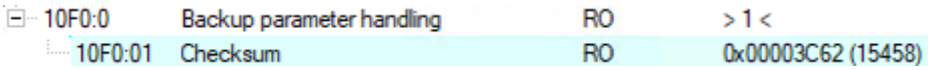


Fig. 161: CoE index 10F0

If a BackUp object is changed, the firmware calculates a new checksum accordingly. This can be used to detect changes to the BackUp objects.

Note: The initial value of the checksum may vary depending on the firmware version, as function extensions can add additional objects to the CRC’s detection range.

7.9 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.

8 Mounting and wiring

8.1 Mounting

8.1.1 Dimensions

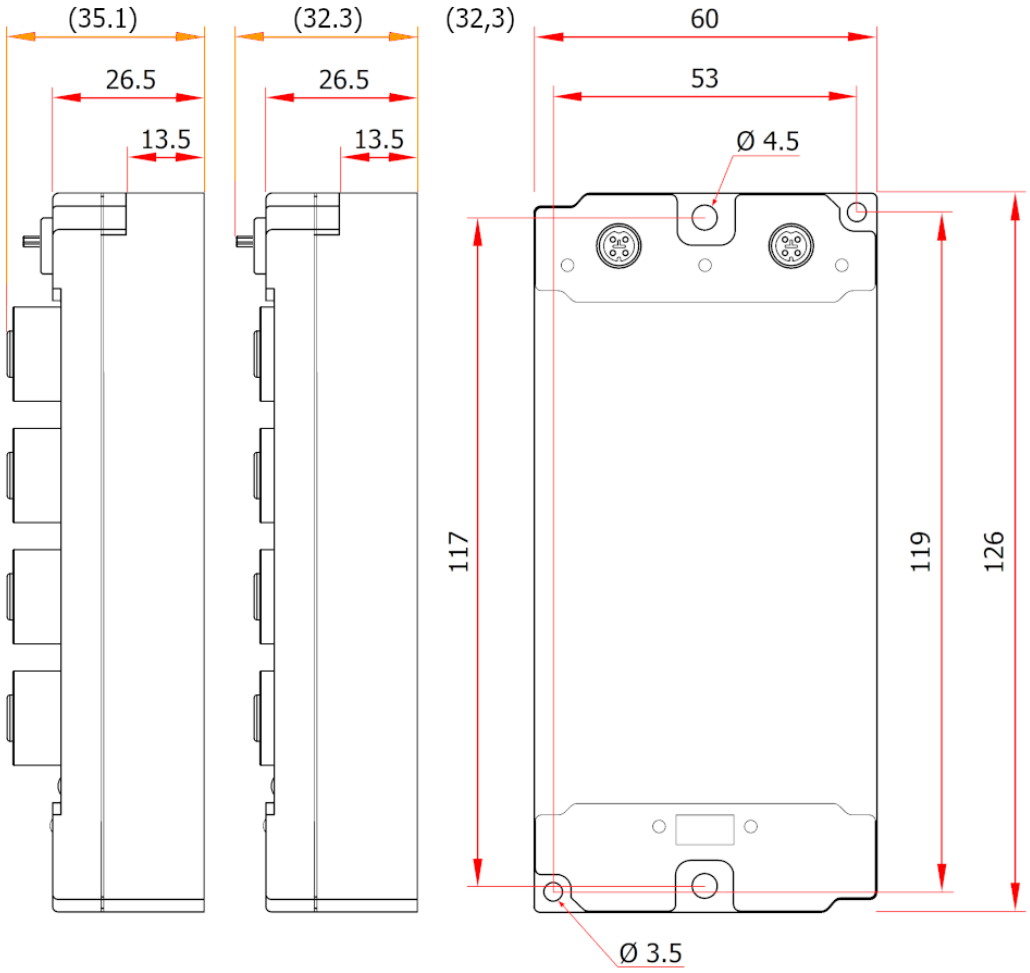


Fig. 162: Dimensions EPP3504-0023, ERP3504-0022 (values in mm)

	EPP3504-0023	ERP3504-0022
Housing material	PA6 (polyamide))	zinc die-cast
Sealing compound	polyurethane	
Mounting	two mounting holes Ø 3.5 mm for M3; two mounting holes Ø 4.5 mm for M4	
Metal parts	brass, nickel-plated	
Contacts	CuZn, gold-plated	
Installation position	variable	
Protection class	IP20	IP67
Dimensions (H x W x D)	approx. 126 x 60 x 26.5 mm (without connectors) approx. 126 x 60 x 32.3 mm (until 2025-10-14 with connectors) approx. 126 x 60 x 35.1 mm (from 2025-10-14 with connectors)	

8.1.2 Fixing

NOTICE

Dirt during assembly

Dirty connectors can lead to malfunctions. Protection class can only be guaranteed if all cables and connectors are connected.

- Protect the plug connectors against dirt during the assembly.

Mount the module with two M4 screws in the centrally located mounting holes.

8.1.3 Functional earth (FE)

The mounting holes [▶ 210] also serve as connections for the functional earth (FE, SGND, see subchapter "Notes on analog measured values"/ "Notices on analog specifications"/ "Explanation of the term GND/ Ground" within chapter "Commissioning" or appendix in the short documentation).

Make sure that the box is grounded to low impedance via the functional earth (FE) connections. You can achieve this, for example, by mounting the box on a grounded machine bed.

Fig. 163: Connection for functional earth (FE) EPP3504-0023

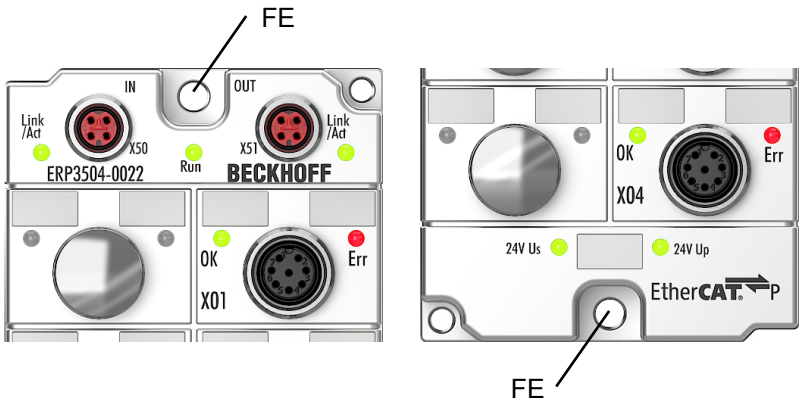


Fig. 164: Connection for functional earth (FE) ERP3504-0022

8.1.4 Tightening torques for plug connectors

Tighten the connectors using a torque wrench. (e.g., Beckhoff ZB8801)

Connector diameter	Tightening torque
M8	0.4 Nm
M12	0.6 Nm

8.2 Notes regarding connectors and wiring

The design of EtherCAT IO modules/terminals/box modules is such that they have two connection sides: one to the fieldbus for communication with the module, which is obligatory, the other to the signal/sensor/actuator to facilitate proper use of the module. The “outer” connection side usually features contacting options for connecting outgoing wires.

Only few I/O devices do not have a second side. Examples include the EL6070 license key terminal and the EL6090 display terminal.

Notes and suggestions for dealing with the connection options are provided below

- **Manufacturer specifications**/notes for connection options must be followed. Any special tools that may have been provided must be used as intended, so that gas-tightness is ensured through the crimping pressure.
- Any detachable connection system is subject to a specified maximum number of **connection cycles**. Each connection/disconnection operation results in wear through friction, mechanical stretching/ relaxation, possibly ingress of contaminants/gases/liquids/condensation, contact discharge, modification of the electrical properties and of the contact point (ohmic contact resistance). In other words, releasing/connecting a contact results in mechanical, chemical and therefore ultimately electrical changes.
  In terms of the application scenario it is therefore important to select suitable connection systems or devices with suitable connection systems:
  - For connections that are more or less permanent, it may make sense to use connectors/contacts with a maximum **number of mating cycles** (as specified by the manufacturer) of 10 to 100 cycles. This may be the case if devices are installed/wired only once, and over the entire lifetime rewiring is only expected to become necessary during maintenance work.
  - For connections that have to be detached on a regular basis, connectors/contacts with a maximum number of mating cycles of 1,000 or higher should be selected. Such connections can typically be found in laboratory environments, where the cabling may be changed several times each day but high-quality contact must nevertheless be ensured over many years.
- When handling and assembling connectors/contacts it is essential to avoid **contact with hand perspiration/liquids**, even for low-tech connections (open stranded wire, spring-loaded technology/ push-in). Acidic/alkaline liquids may have a very aggressive effect on the contact surface and quickly lead to structural changes and oxidation layers. These are very disruptive for analog measurements, particularly since they undermine the reproducibility of measurements and can therefore result (if known) in large systematic measurement uncertainty. It may be possible to rectify the problem by thorough follow-up cleaning.
- The actual/expected **load during operation** must be taken into account when selecting connectors.
- Abnormal vibrations can lead to microfriction/corrosion and change the electrical properties, potentially resulting in complete loss of contact.
- Temperature variations affect the mechanical strength of the connection and the spring forces in metallic components.
- Exposure to gas or liquid can damage the connection, particularly if the gas or liquid penetrates to the actual contact region and is unable to escape from there.
- Of high relevance for analog measurements is the **electrical quality** of the connection, both in the short term during commissioning and over the service life under external influences and perhaps repeated mating cycles. This is expressed in the repeatability of the transition. The influence should be checked against the expected accuracy. Of particular relevance is the (frequency dependent) contact resistance. Effects can be:
  - Increasing the contact resistance results in a voltage drop when power is transmitted, potentially leading to critical self-heating.
  - The internal voltage drop can distort corresponding measurements. In order to avoid negative effects, 4/5/6-wire connections should be used in SG/resistance measurements, since non-live contacts are no longer affected by a distorting voltage drop. The popular 3-wire connection for resistance measurement (PT100, PT1000 etc.) does not provide absolute protection, since the singular line cannot be diagnosed. Current/voltage measurements in industrial environments are less sensitive to contact changes.

  - A defective contact surface can lead to random resistance values, depending on the contact position and temperature. This makes reproducible measurements difficult.

- The **effort for establishing the connection**, including assembling the cables and connectors, generally increases with increasing transmission quality requirements. This applies to the tools, diligence and time required. Examples:
  - Spring-loaded technology/push-in connections (e.g. Beckhoff EL terminals), which are common in automation applications, can be established or released in a few seconds with or without ferrule. A screwdriver or push pin is sufficient. On the other hand, in many cases the (ohmic) repeatability is insufficient for high-precision measurements in the SG/R range.
  - Some 10 minutes and costs of some 10 euros should be assumed for assembly a lab-standard LEMO/ODU connector (Beckhoff ELM3704-0001), depending on the number of poles. The result is a top-quality connection system with a high number of permissible mating cycles.
  - An intermediate solution can be field-configurable M8/M12 connections. For reasons of tightness, they are more elaborate to assemble (soldering or insulation displacement contact, if necessary), although the maximum number of mating cycles is similar to maintenance connectors.
- A pre-assembled connection should be subjected to electrical/mechanical testing before commissioning: visual inspection, pull-out test, crimp height measurement, resistance measurement etc.

8.3 Note - power supply

⚠ WARNING

Power supply from SELV / PELV power supply unit!

SELV / PELV circuits (safety extra-low voltage / protective extra-low voltage) according to IEC 61010-2-201 must be used to supply this device.

Notes:

- SELV / PELV circuits may give rise to further requirements from standards such as IEC 60204-1 et al, for example with regard to cable spacing and insulation.
- A SELV supply provides safe electrical isolation and limitation of the voltage without a connection to the protective conductor, a PELV supply also requires a safe connection to the protective conductor.

8.4 Notes on connection technology EPP3504/ ERP3504

EPP3504-0023

There are 4 strain gauge connectors included with the box.

Connector properties:

Push-in spring connection for direct wiring, plug-in unit detachable for maintenance purposes

Conductor cross-section min.	0.2 mm²
Conductor cross-section max.	1.5 mm²
Conductor cross-section flexible min.	0.2 mm²
Conductor cross-section flexible max.	1.5 mm²
Conductor cross-section ferrule without plastic sleeve min.	0.25 mm²
Conductor cross-section ferrule without plastic sleeve max.	1.5 mm²
Conductor cross-section ferrule with plastic sleeve min.	0.14 mm²
Conductor cross-section ferrule with plastic sleeve max.	0.75 mm²
Conductor cross-section AWG min.	24
Conductor cross-section AWG max.	16

ERP3504-0022

See Accessories [▶ 218] for plugs and cables.

8.5 Connectors

NOTICE

Risk of damage to the device!

Bring the EtherCAT/EtherCAT P system into a safe, powered down state before starting installation, disassembly or wiring of the modules!

Two M8 sockets at the upper end of the modules are provided for supply and downstream connection of EtherCAT P:

- IN: left M8 socket for EtherCAT P supply
- OUT: right M8 socket for downstream connection of EtherCAT P

The metal threads of the M8 EtherCAT P sockets are internally linked to the FE connection via high impedance RC combination. See chapter Functional earth (FE) [▶ 211].

Fig. 165: Connectors for EtherCAT P

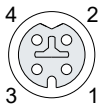


Fig. 166: M8 socket, P-coded

Contact	Signal	Voltage	Core color [1)]
1	Tx +	GND_S	yellow
2	Rx +	GND_P	white
3	Rx -	U_P: peripheral voltage, +24 V_{DC}	blue
4	Tx -	U_S: control voltage, +24 V_{DC}	orange
Housing	Shield	Shield	Shield

[1)] The core colors apply to EtherCAT P cables and ECP cables from Beckhoff.

8.6 Accessories

Plug

Ordering information	Description
ZS2001-0013	Spare plug 6-pole for EPP3504 (10 pieces)
ZS2000-6713	M12 plug field assembly, sensor, IP67, shielded, for ERP3504

Mounting

Ordering information	Description	Link
ZS5300-0011	Mounting rail	Website

Cables

A complete overview of pre-assembled cables for fieldbus components can be found here.

Ordering information	Description	Link
ZK700x-xxxx-xxxx	EtherCAT P cable M8	Website
ZK2000-8100-0xxx	Sensor cable, PUR, 8 x 0.25 mm², drag-chain suitable (for ERP3504)	Website
ZK2000-8181-3xxx	Sensor cable, PVC, 8 x 0.25 mm², shielded, fixed installation (for ERP3504)	Website

Labeling material, protective caps

Ordering information	Description
ZS5000-0010	Protective cap for M8 sockets, IP67 (50 pieces)
ZS5000-0012	Protective cap for M8 sockets, P-coded, IP67 (50 pieces)
ZS5100-0000	Inscription labels, unprinted, 4 strips of 10
ZS5000-xxxx	Printed inscription labels on enquiry

Ordering information	Description	Link
ZS5000-0051	blanking plug, stainless steel, IP69K, for M12 socket, with outer thread (for ERP3504)	Website

Tools

Ordering information	Description
ZB8801-0000	Torque wrench for plugs, 0.4…1.0 Nm
ZB8801-0001	Torque cable key for M8 / wrench size 9 for ZB8801-0000

Further accessories

Further accessories can be found in the price list for fieldbus components from Beckhoff and online at https://www.beckhoff.com.

8.7 Power supply, potential groups

The electronics of a fieldbus-connected I/O device generally consists of two potential groups (exceptions possible, see respective device documentation):

- the communication section, the so-called bus side. This is usually supplied by the control voltage U_S. In EtherCAT Terminals this section is directly connected to the internal 5 V supply from the E-bus and is not directly accessible by the user.
- the signal section, for connecting the input/output signals, the so-called field side.
  This is usually supplied by the peripheral voltage U_P. It consists of 1..n function channels.
  - usually all channels of the device are contiguous in this island, there is no electrical isolation (separate GND) of the channels.
  - in some devices, channels or channel groups may in turn be isolated as sub-islands. The height of the max. permissible electrical isolation is then specified. The device then consists of several potential groups: the bus side and the n channels.
  - depending on the device, the field side can also be supplied
    - indirectly via U_S by transporting the necessary power via the electrical isolation from the bus side to the field side; connection of U_P (or power contacts) is then not required.
    - directly via U_S
  - in special cases (e.g. EL6070 dongle terminal etc.) there is no accessible field side.

Both potential groups are usually electrically isolated. The "load capacity" of the isolation must then be observed in detail, i.e. the voltage difference/potential difference in continuous operation or for a short time between the two areas.

The internal electronics can be supplied via the bus side, field side or both, depending on the device. See the relevant notes about this in the respective device specification.

The plug used can also have an influence on the potential groups; if necessary, its housing is conductively connected to the housing of the box module.

The external system GND (DIN rail, SGND, FE) is always present and represents the reference ground.

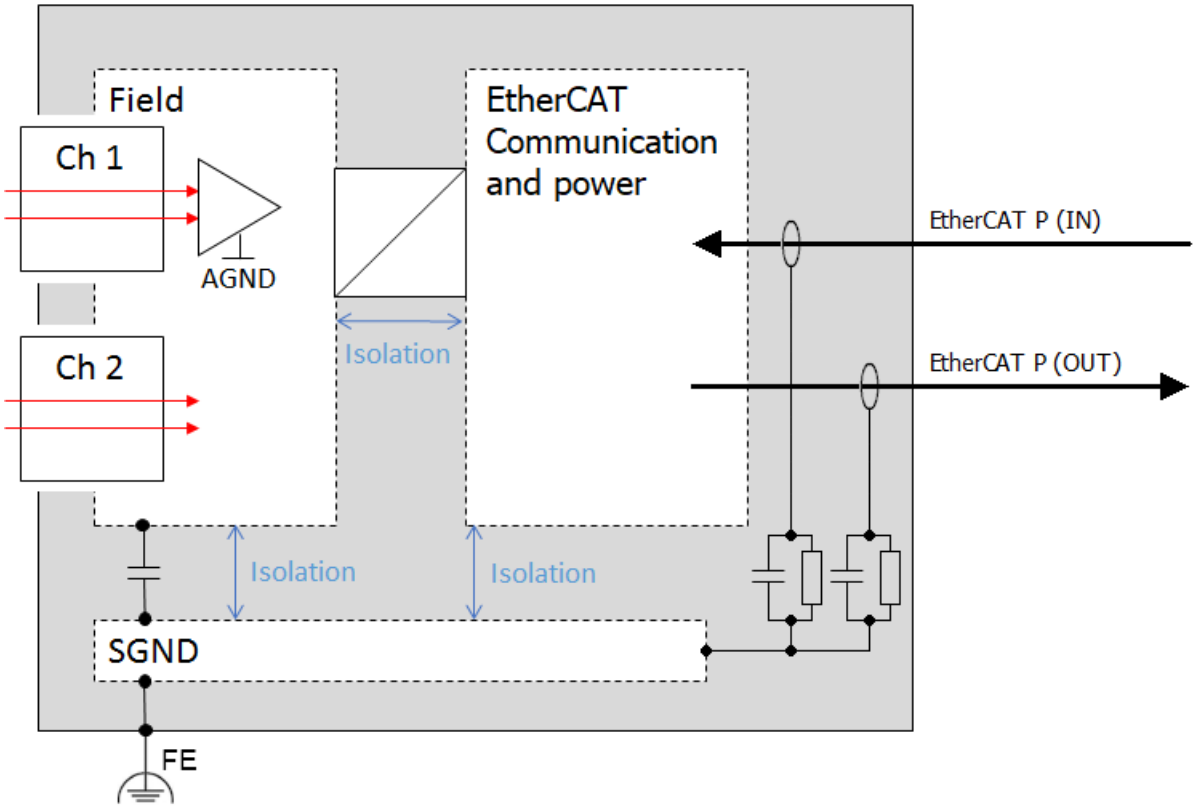
In the following the permissible potential difference is referred to only as "Insulation"; the exact specification (value, type and, if applicable, insulating strength) can be found in the respective specifications of the device.

NOTICE

Isolation between the potential groups in practice

The potential groups are theoretically electrically isolated, i.e. there are only parasitic ohmic connections in the range of MΩ and higher that are unavoidable due to the electronics.
The load capacity of the isolation with regard to voltage level and duration is specified. It results among other things from internal isolation distances and the group-spanning components used, e.g. data transmitters or transformers, and is formulated in view of the underlying standards, which describe application aspects such as aging, contamination or defined overvoltage events.
From this it can be seen that, in practice, potential groups cannot to be operated arbitrarily isolated from the environment. In particular, if EMC disturbances penetrate the potential group, conducted by the external cables or radiated, then this energy seeks its way to SGND and finds it in every case undefined in the group-spanning elements mentioned above. Therefore, practice has shown that potential groups of all kinds should be purposefully and intentionally connected to each other and to SGND with small capacitances in the nF range for interference dissipation, so that the HF interference (and this already starts at 50 Hz) finds a defined path and does not impair the operability.
The ohmic effect of the capacitors in relation to the parasitic ohmic effects is negligible.

The following potential scheme can be specified:



Properties:

- electrical isolation at a specified level
  - between field and E-bus: yes
  - between field and SGND: yes
  - between E-bus and SGND: yes
  - between the channels: no
- Power contacts in use: no
- Connection type: Push-in spring connection

8.8 Status LEDs

8.8.1 RUN, Channel 1 ... Channel 4

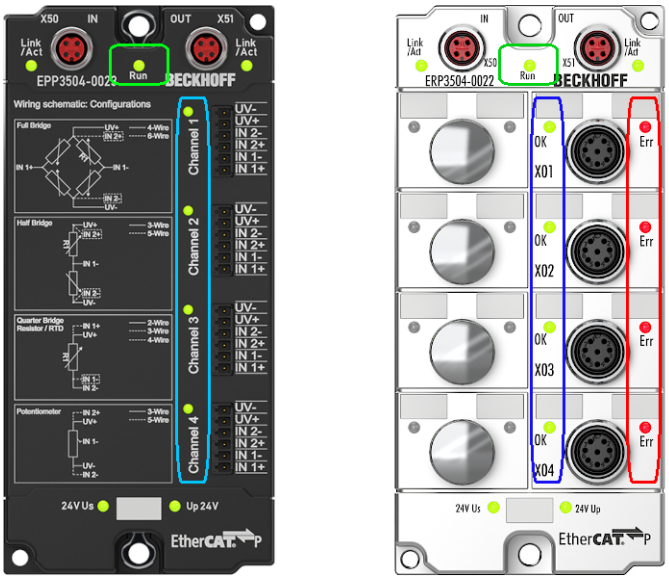


Fig. 167: EPP3504-0023 LEDs RUN, Channel

LED	Color	Meaning	
RUN	green	off	State of the EtherCAT State Machine [▶ 201]: **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** = check the channels of the Sync Manager [▶ 187] and the Distributed Clocks [▶ 208] (if supported)
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible
		flickering	State of the EtherCAT State Machine: **BOOTSTRAP** = function for firmware updates [▶ 237] of the terminal

EPP3504-0023 status LEDs

LED	Color	Meaning
Channel (1…4)	green	*No error*
	red	*Error display, along with error bit in the status, for* • *Measuring range error (not for underrange/overrange!)* • *Measurement type that is not calibrated has been set (see CoE object 0x80nF PAI Vendor Calibration Data [▶ 76])* • *Processor overload (see CoE object 0xF900 PAI Info Data [▶ 85])* • *ADC in "saturation"* • *Analog circuit overloaded "in overload", overvoltage detected at inputs; see chapter "StartUp - What to do if...", as well as notes in chapter "General technical data [▶ 19]"* • *Oversampling error in synchron mode*
	flashing	*Active self-test of the terminal;* *see chapter* "Features"/ "Self-test and self-test report"
	off	*No operation*

ERP3504-0022 status LEDs

LED	Color	Meaning
OK (1…4)	green	*Operation*
	flashing	*Active self-test of the terminal; see chapter* "Features"/ "Self-test and self-test report"
	off	*No operation*
Err (1…4)	red	*Error display, along with error bit in the status, for* • *Measuring range error (not for underrange/overrange!)* • *Measurement type that is not calibrated has been set (see CoE object* 0x80nF PAI Vendor Calibration Data [▶ 76]) • *Processor overload (see CoE object* 0xF900 PAI Info Data [▶ 85]) • *ADC in "saturation"* • *Analog circuit overloaded "in overload", overvoltage detected at inputs; see chapter* "StartUp - What to do if...", *as well as notes in chapter* "General technical data [▶ 19]" • *Oversampling error in synchron mode*
	off	*No error*

8.8.2 Supply voltage

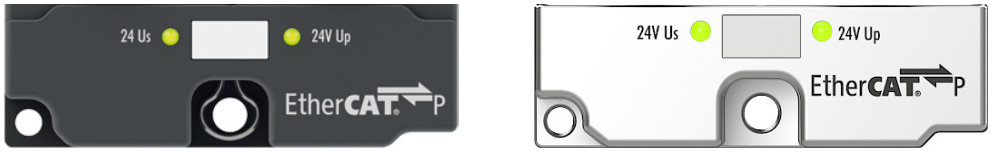


Fig. 168: Status LEDs for the power supply

EtherCAT P Box modules indicate the status of the supply voltages via two status LEDs. The status LEDs are labeled with the designations of the supply voltages: Us and Up.

LED	Indicator	Meaning
U_S (control voltage)	Off	Power supply U_S is not present.
	Lights green	Power supply U_S is present.
U_P (peripheral voltage)	Off	Power supply U_P is not present.
	Lights green	Power supply U_P is present.

8.8.3 EtherCAT

Fig. 169: Status LEDs for ethercat

L/A (Link/Act)

A green LED labeled "L/A" or "Link/Act" is located next to each EtherCAT/EtherCAT P socket. The LED indicates the communication state of the respective socket:

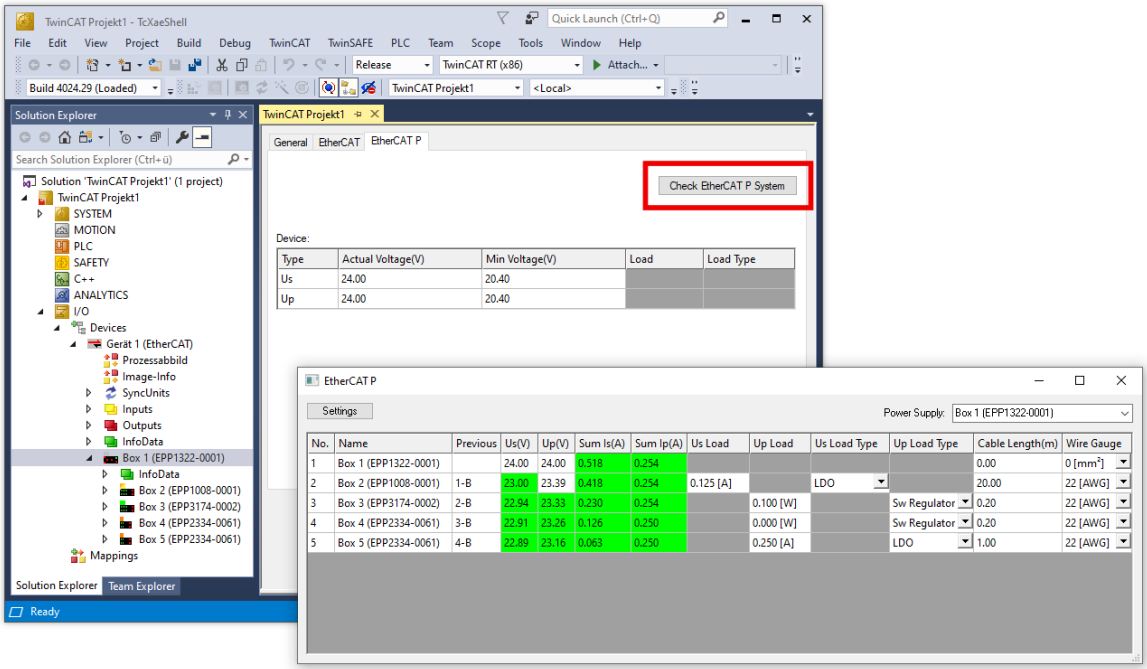
LED	Meaning
off	no connection to the connected EtherCAT device
lit	LINK: connection to the connected EtherCAT device
flashes	ACT: communication with the connected EtherCAT device

8.9 Conductor losses

Take into account the voltage drop on the supply line when planning a system. Avoid the voltage drop being so high that the supply voltage at the box lies below the minimum nominal voltage.
Variations in the voltage of the power supply unit must also be taken into account.

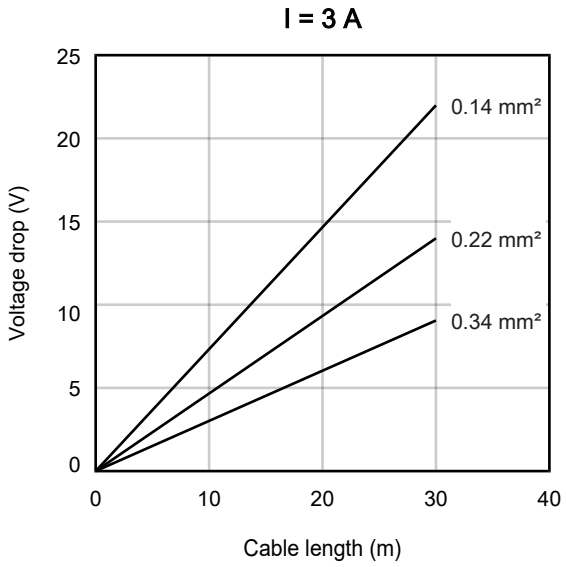
Planning tool for EtherCAT P

You can plan cable lengths, voltages and currents of your EtherCAT P system using TwinCAT 3. The requirement for this is TwinCAT 3 Build 4020 or higher.



Further information can be found in the quick start guide IO configuration in TwinCAT in chapter "Configuration of EtherCAT P via TwinCAT".

Voltage drop on the supply line



8.10 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.

9 Appendix

9.1 Volatility

If there are requirements concerning the volatility of products in your application, for example of the U.S. Department of Defense or similar authorities or security organizations, the following process applies:

The product has both volatile and non-volatile components. Volatile components lose their data immediately after removing power. Non-volatile components keep the data even after loss of power.

If there is customer specific data saved on the product, it cannot be ensured that this data might not be restored through for example forensic measures, even after the data is deleted through the provided tool chain. If this data is confidential, the scrapping of the product after usage is recommended to protect this data.

9.2 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:

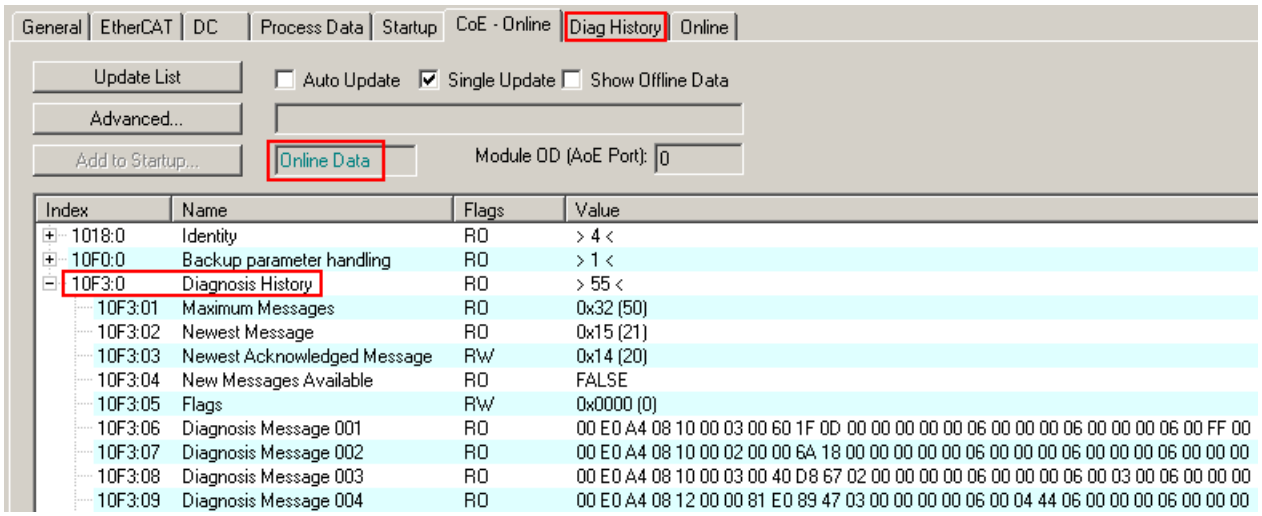


Fig. 170: DiagMessages in the CoE

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.

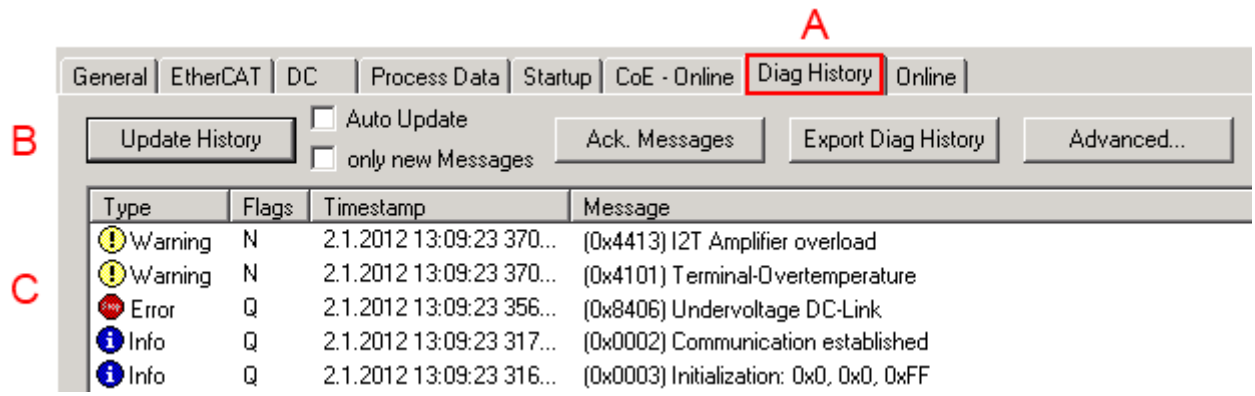


Fig. 171: Implementation of the DiagMessage system in the TwinCAT System Manager

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

From TwinCAT 3.1 build 4022 onwards, DiagMessages sent 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:

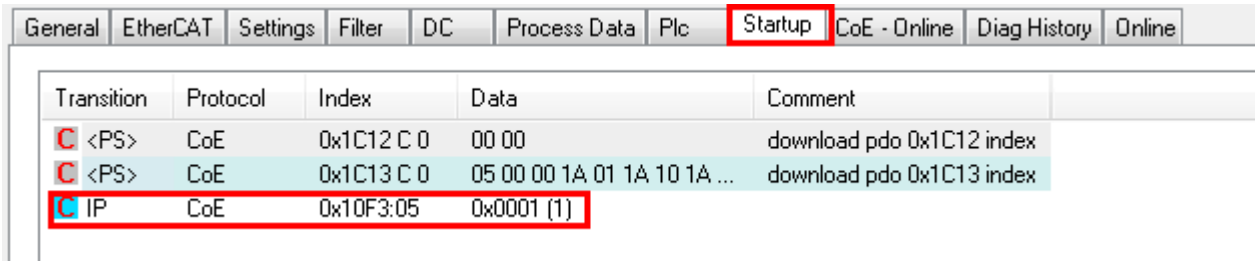


Fig. 172: Startup List

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. The value can then be set back to 1 later from the PLC via CoE access if required.

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

Text ID	Type	Place	Text Message	Additional comment
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))	

Text ID	Type	Place	Text Message	Additional comment
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.
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.

Text ID	Type	Place	Text Message	Additional comment
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	

Text ID	Type	Place	Text Message	Additional comment
0x8289	Error	Communication	Challenge could not be hashed: %X	
0x828A	Error	Communication	Tickstamp Process failed	
0x828B	Error	Communication	PCR Process failed: %X	
0x828C	Error	Communication	Quote Process failed: %X	
0x82FF	Error	Communication	Bootmode not activated	Boot mode not activated
0x8300	Error	Encoder	Set position error: 0x%X, %d	Error while setting the position
0x8301	Error	Encoder	Encoder increments not configured: 0x%X, %d	Encoder increments not configured
0x8302	Error	Encoder	Encoder error	The amplitude of the resolver is too small
0x8303	Error	Encoder	Encoder 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

Text ID	Type	Place	Text Message	Additional comment
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
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

9.3 TcEventLogger and IO

The TwinCAT 3 EventLogger provides an interface for the exchange of messages between TwinCAT components and non-TwinCAT components.

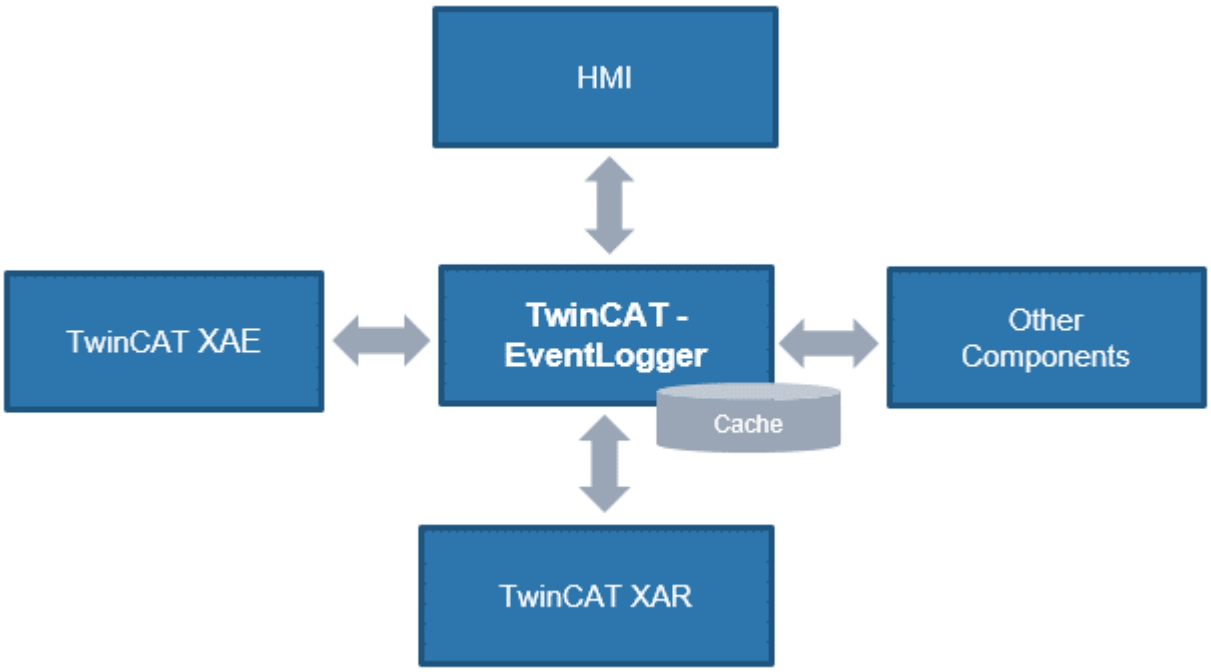


Fig. 173: Schematic representation TCEventLogger

Refer to the explanations in the TwinCAT EventLogger documentation, e.g. in the Beckhoff InfoSys https://infosys.beckhoff.com/ → TwinCAT 3 → TE1000 XAE → Technologies → EventLogger.

The EventLogger saves to a local database under ..\TwinCAT\3.1\Boot\LoggedEvents.db and, unlike the VisualStudio Error Window, is designed for continuous recording.

IO devices can also be a source of messages. If so-called DiagMessages are generated in the IO device, they can be collected by TwinCAT over EtherCAT and displayed in the TcEventLogger with the appropriate device setting. This facilitates the central management of events that hinder operation, as a textual diagnosis no longer needs to be programmed out in the application for each individual IO device. The messages/ events can be displayed directly in the TwinCAT HMI, for example, and thus facilitate the diagnosis.

Notes:

- This feature is supported from TwinCAT 3.1 build 4022.16.
- TwinCAT may be in the RUN or CONFIG mode
- On the manufacturer side, the IO device regarded must (1) generate local DiagMessages and (2) be fundamentally capable of transmitting them as events over EtherCAT. This is not the case with all EtherCAT IO devices/terminals/box modules from Beckhoff.

The messages managed by the EventLogger can be output in or read from

- the HMI → EventGrid
- C#
- the PLC
- TwinCAT Engineering → Logged Events

The use of the EventLogger with EtherCAT IO with TwinCAT 3.1 build 4022.22 during commissioning is explained below.

- The EventLogger window may need to be displayed in the TwinCAT Engineering

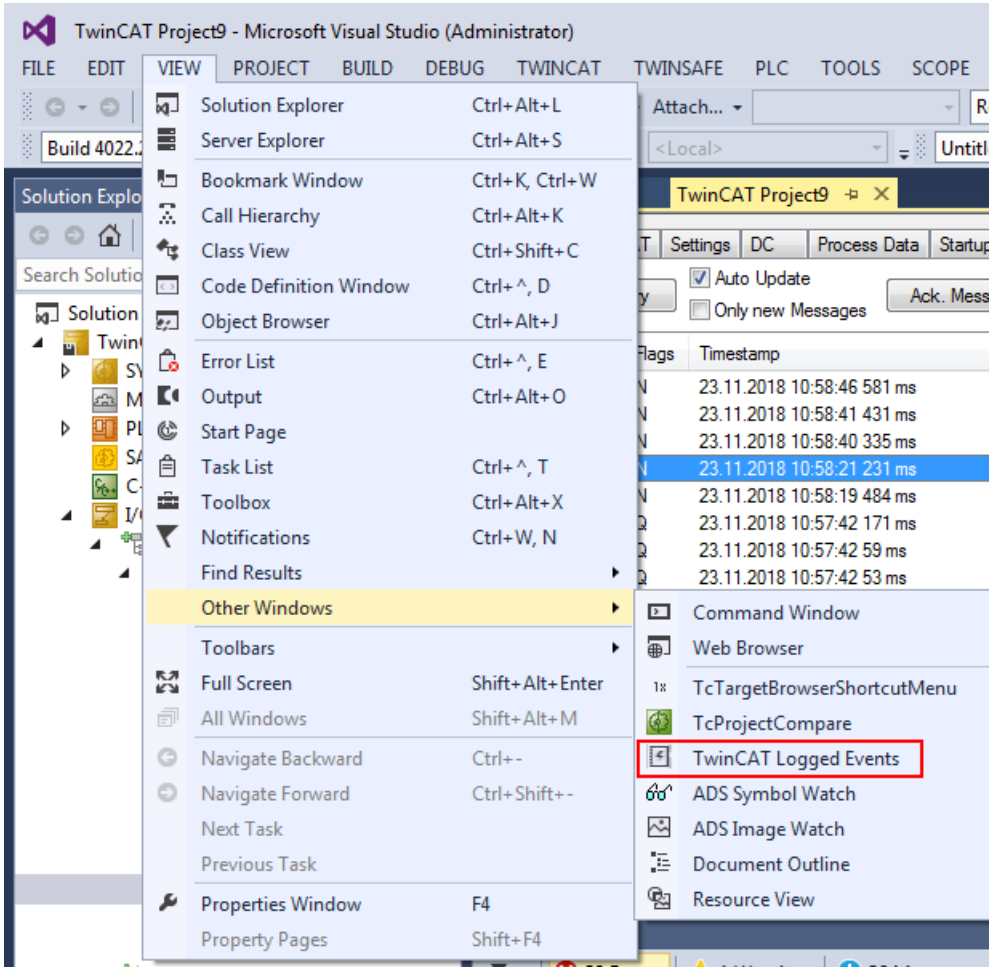


Fig. 174: Display EventLogger window

- Some DiagMessages and the resulting Logged Events are shown below, taking an ELM3602-0002 as an example

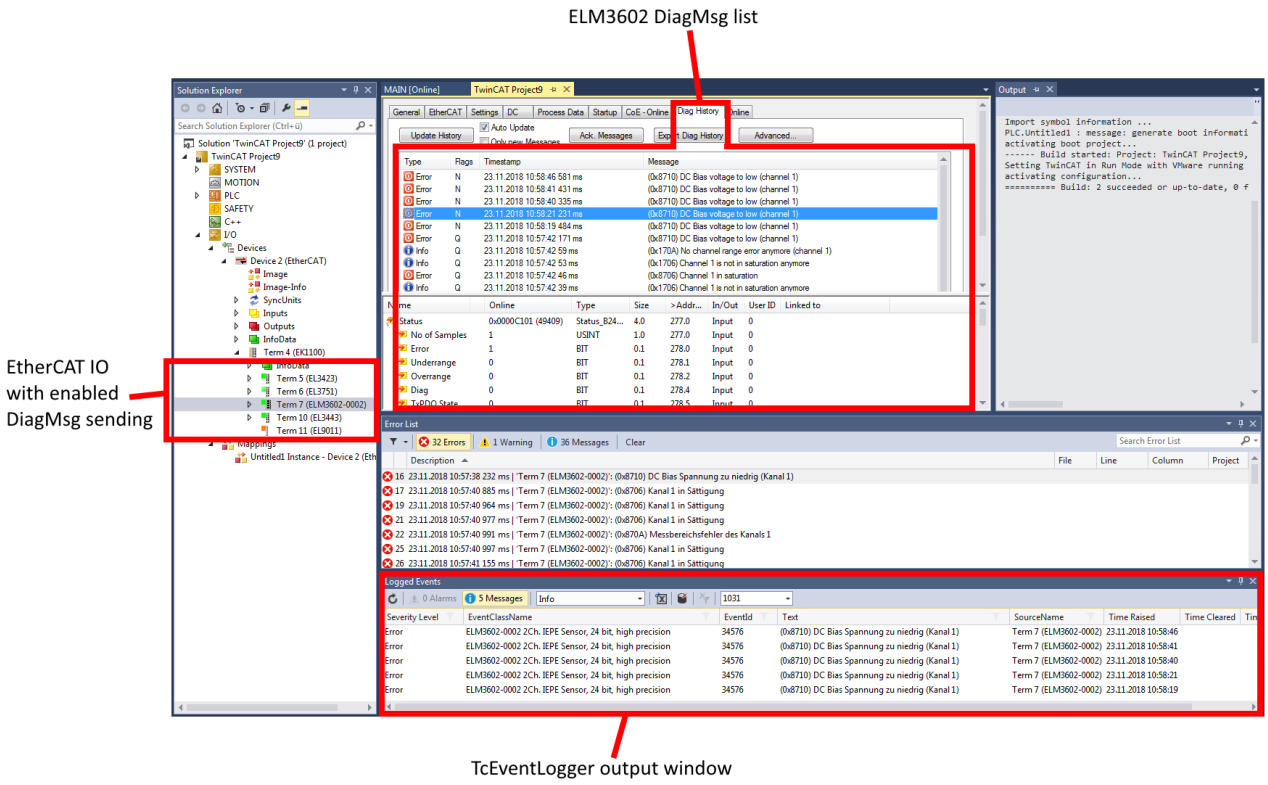


Fig. 175: Display DiagMessages and Logged Events

- Filtering by entries and language is possible in the Logger window.
  German: 1031
  English: 1033

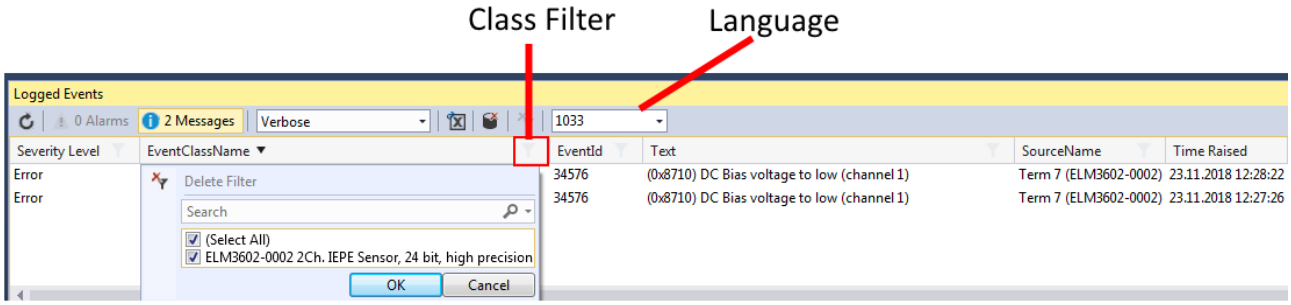


Fig. 176: Setting filter language

- If an EtherCAT slave is enabled by default to transmit DiagMessages as events over EtherCAT, this can be activated/deactivated for each individual slave in the CoE 0x10F3:05. TRUE means that the slave provides events for collection via EtherCAT, while FALSE deactivates the function.

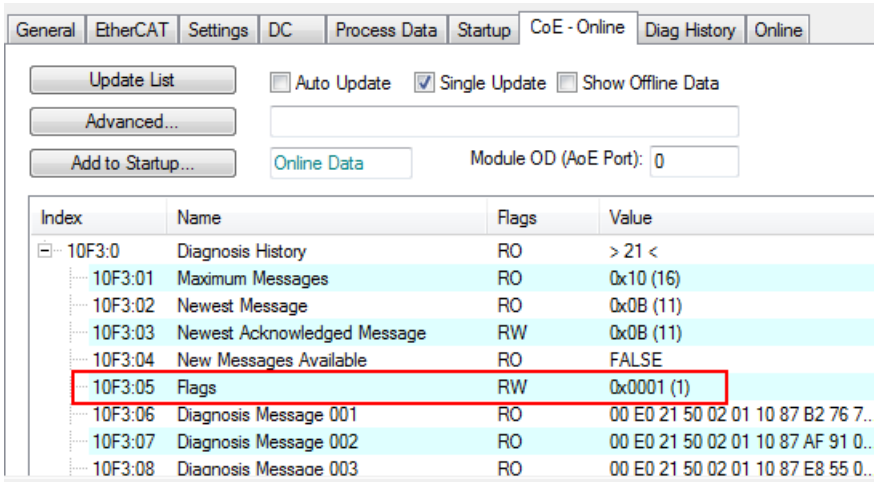


Fig. 177: Activating/deactivating event transmission

- In the respective EtherCAT slave, various "causes" can lead to it transmitting DiagMessages or events. If only some of these are to be generated, you can read in the device documentation whether and how individual causes can be deactivated, e.g. through CoE settings.
- Settings for the TwinCAT EventLogger can be found under Tools/Options

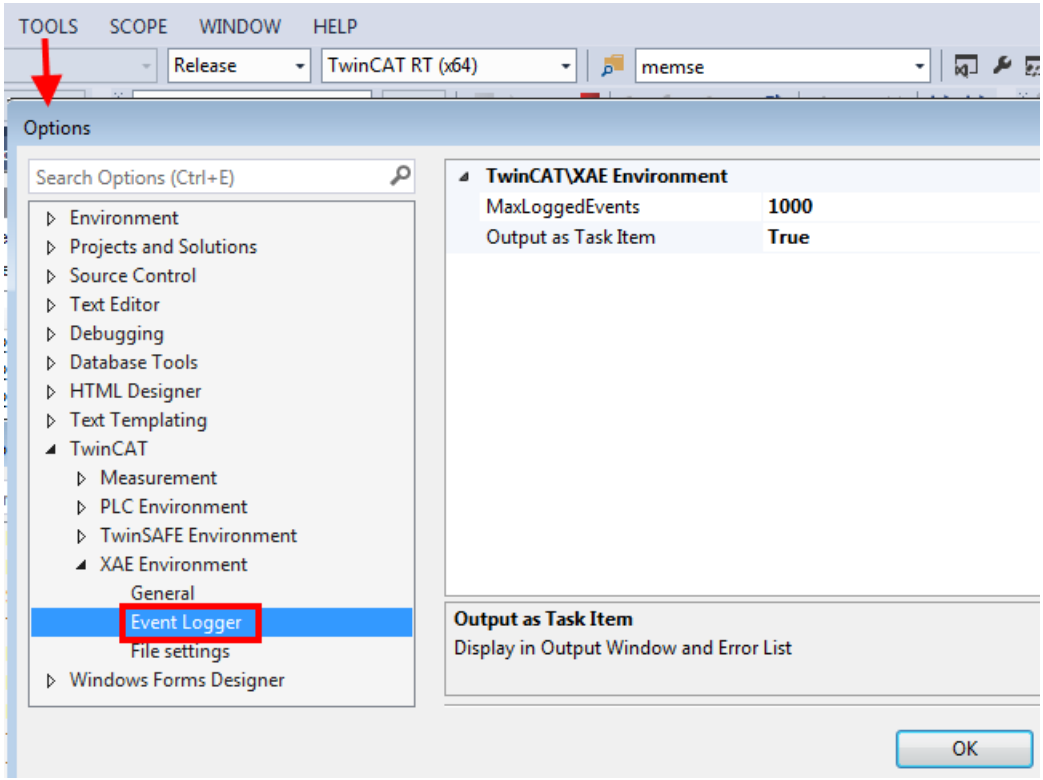


Fig. 178: Settings TwinCAT EventLogger

9.4 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

9.5 Firmware Update

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

NOTICE

Only use TwinCAT 3 software!

A firmware update of Beckhoff IO devices must only be performed with a TwinCAT 3 installation. It is recommended to build as up-to-date as possible, available for free download on the Beckhoff website.

To update the firmware, TwinCAT can be operated in the so-called FreeRun mode, a paid license is not required.

The device to be updated can usually remain in the installation location, but TwinCAT has to be operated in the FreeRun. Please make sure that EtherCAT communication is trouble-free (no LostFrames etc.).

Other EtherCAT master software, such as the EtherCAT Configurator, should not be used, as they may not support the complexities of updating firmware, EEPROM and other device components.

Storage locations

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

- Each EtherCAT slave has a device description, consisting of identity (name, product code), timing specifications, communication settings, etc.
  This device description (ESI; EtherCAT Slave Information) can be downloaded from the Beckhoff website in the download area as a zip file and used in EtherCAT masters for offline configuration, e.g. in TwinCAT.
  Above all, each EtherCAT slave carries its device description (ESI) electronically readable in a local memory chip, the so-called **ESI EEPROM**. When the slave is switched on, this description is loaded locally in the slave and informs it of its communication configuration; on the other hand, the EtherCAT master can identify the slave in this way and, among other things, set up the EtherCAT communication accordingly.

NOTICE

Application-specific writing of the ESI-EEPROM

The ESI is developed by the device manufacturer according to ETG standard and released for the corresponding product.
- Meaning for the ESI file: Modification on the application side (i.e. by the user) is not permitted.
- Meaning for the ESI EEPROM: Even if a writeability is technically given, the ESI parts in the EEPROM and possibly still existing free memory areas must not be changed beyond the normal update process. Especially for cyclic memory processes (operating hours counter etc.), dedicated memory products such as EL6080 or IPC's own NOVRAM must be used.

- Depending on functionality and performance EtherCAT slaves have one or several local controllers for processing I/O data. The corresponding program is the so-called **firmware** in *.efw format.
- In some EtherCAT slaves the EtherCAT communication may also be integrated in these controllers. In this case the controller is usually a so-called **FPGA** chip with *.rbf firmware.

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. ELxxxx-xxxx_REV0016_SW01.efw
- for password=1 to be entered in the download dialog. If password=0 (default setting) only the firmware update is carried out, without an ESI update.
- for the device to support this function. The function usually cannot be retrofitted; it is a component of many new developments from year of manufacture 2016.

Following the update, its success should be verified

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

NOTICE

Risk of damage to the device!

✓ Note the following when downloading new device files

a) Firmware downloads to an EtherCAT device must not be interrupted

b) Flawless EtherCAT communication must be ensured. CRC errors or LostFrames must be avoided.

c) The power supply must adequately dimensioned. The signal level must meet the specification.

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

9.5.1 Device description ESI file/XML

NOTICE

Attention regarding update of the ESI description/EEPROM

Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

The ESI device description is stored locally on the slave and loaded on start-up. Each device description has a unique identifier consisting of slave name (9 characters/digits) and a revision number (4 digits). Each slave configured in the System Manager shows its identifier in the EtherCAT tab:

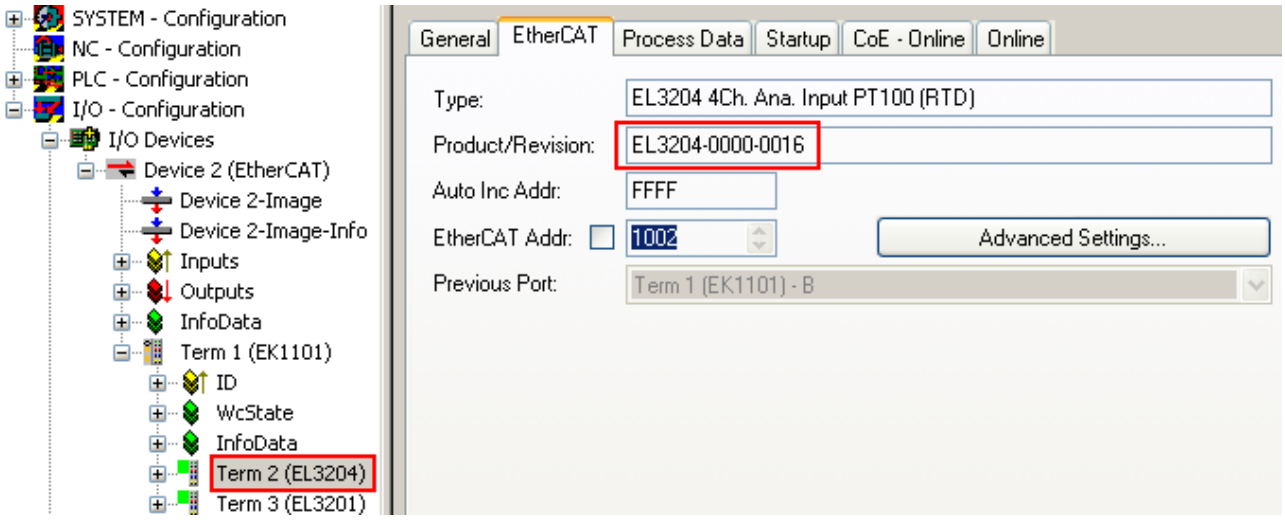


Fig. 179: Device identifier consisting of name EL3204-0000 and revision -0016

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:

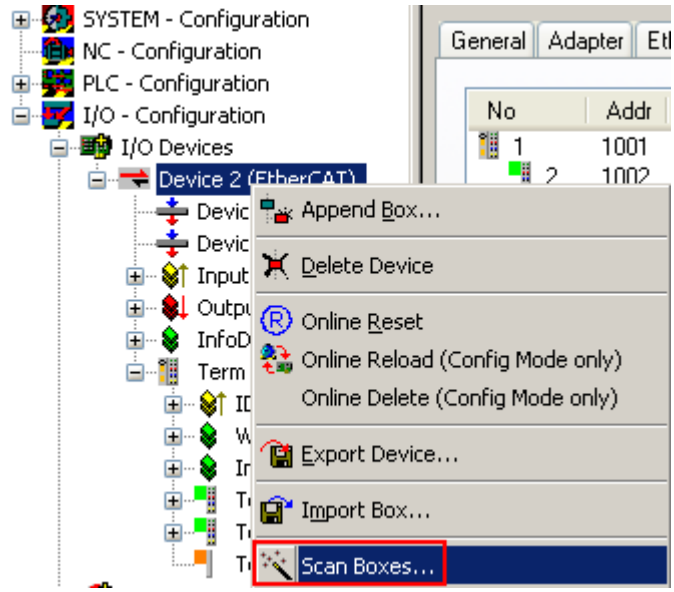


Fig. 180: Scan the subordinate field by right-clicking on the EtherCAT device

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

Fig. 181: Configuration is identical

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

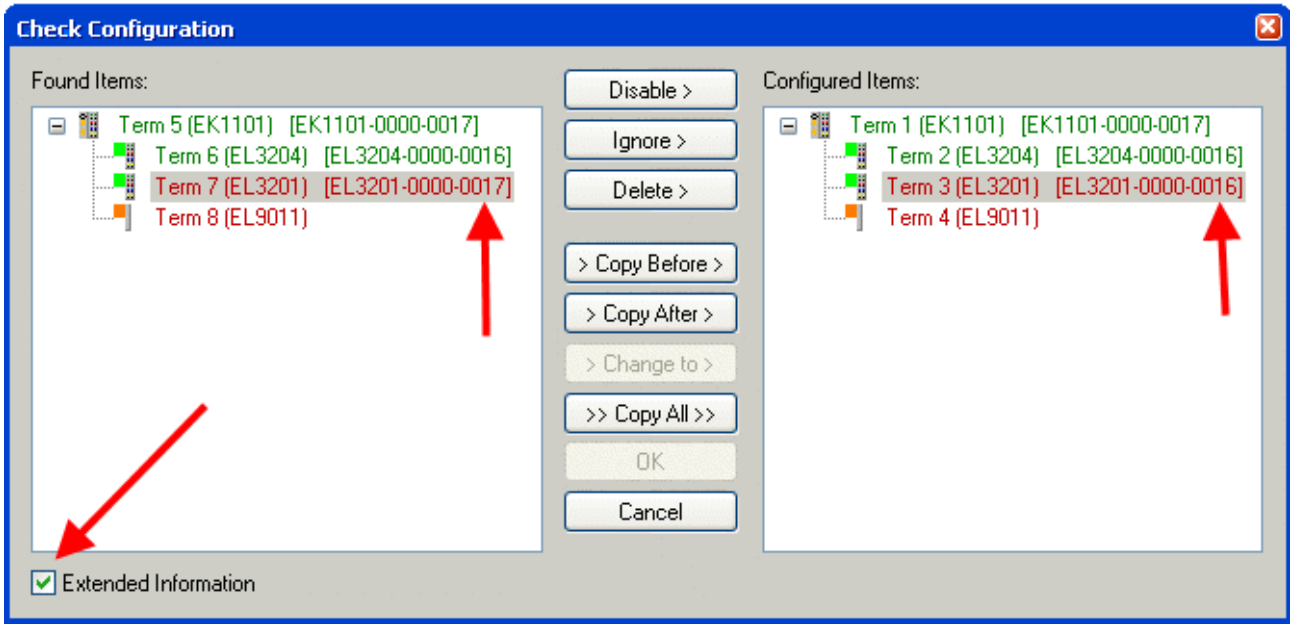


Fig. 182: Change dialog

In this example in Fig. *Change dialog*, an EL3201-0000-**0017** was found, while an EL3201-0000-**0016** was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *EEPROM Update*

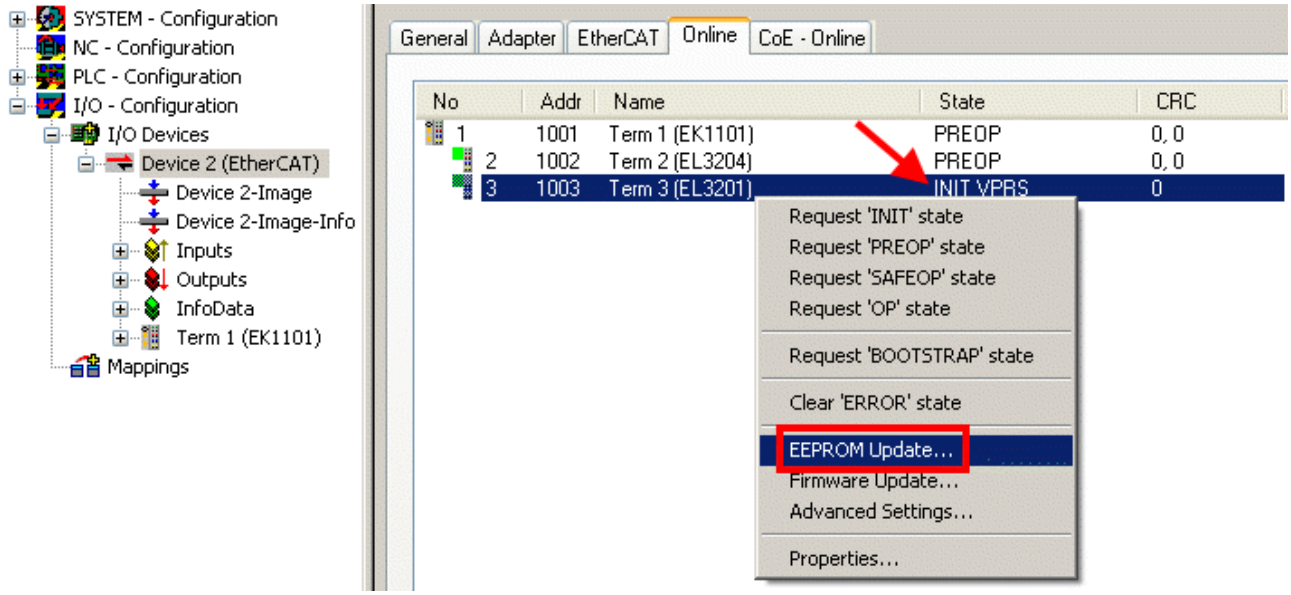


Fig. 183: EEPROM Update

The new ESI description is selected in the following dialog, see Fig. *Selecting the new ESI.* The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.

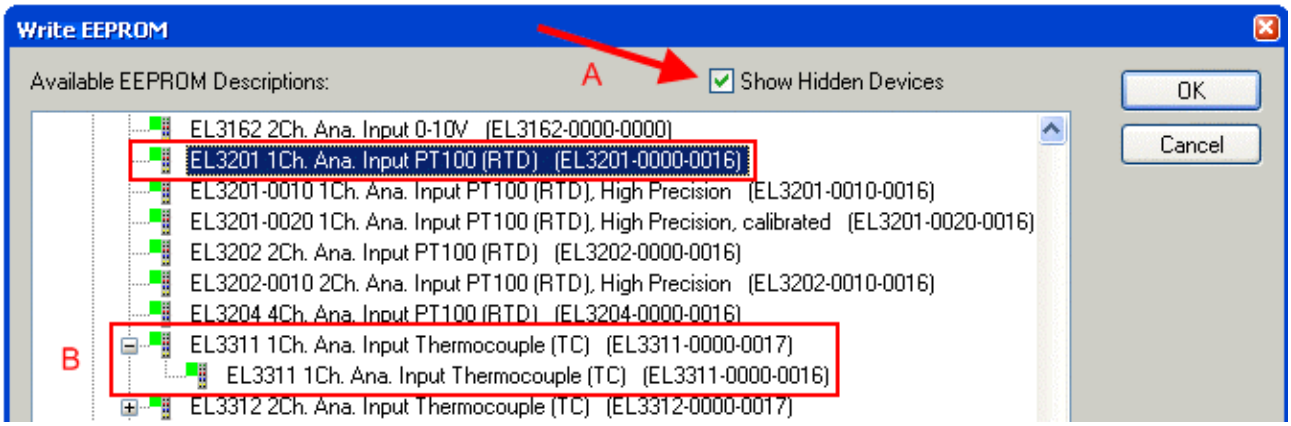


Fig. 184: Selecting the new ESI

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.

9.5.2 Firmware explanation

Determining the firmware version

Determining the version via the TwinCAT 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.

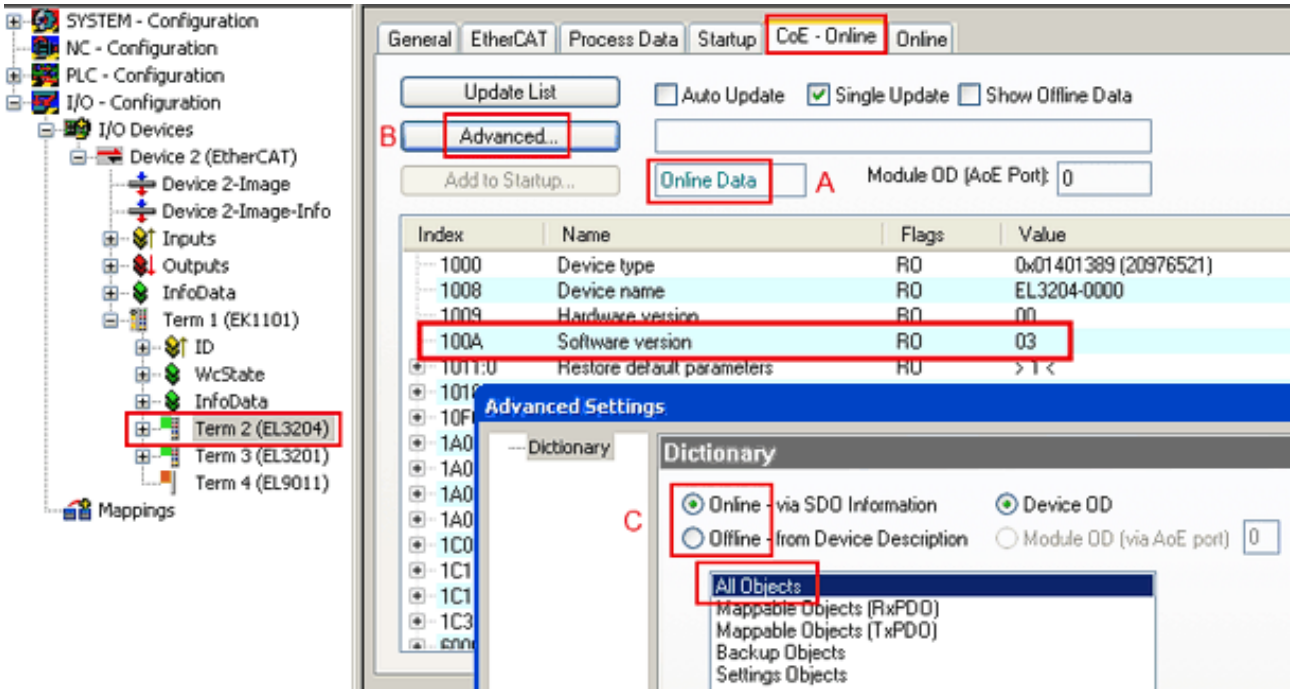


Fig. 185: Display of EL3204 firmware version

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*.

9.5.3 Updating controller firmware *.efw

CoE directory

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

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

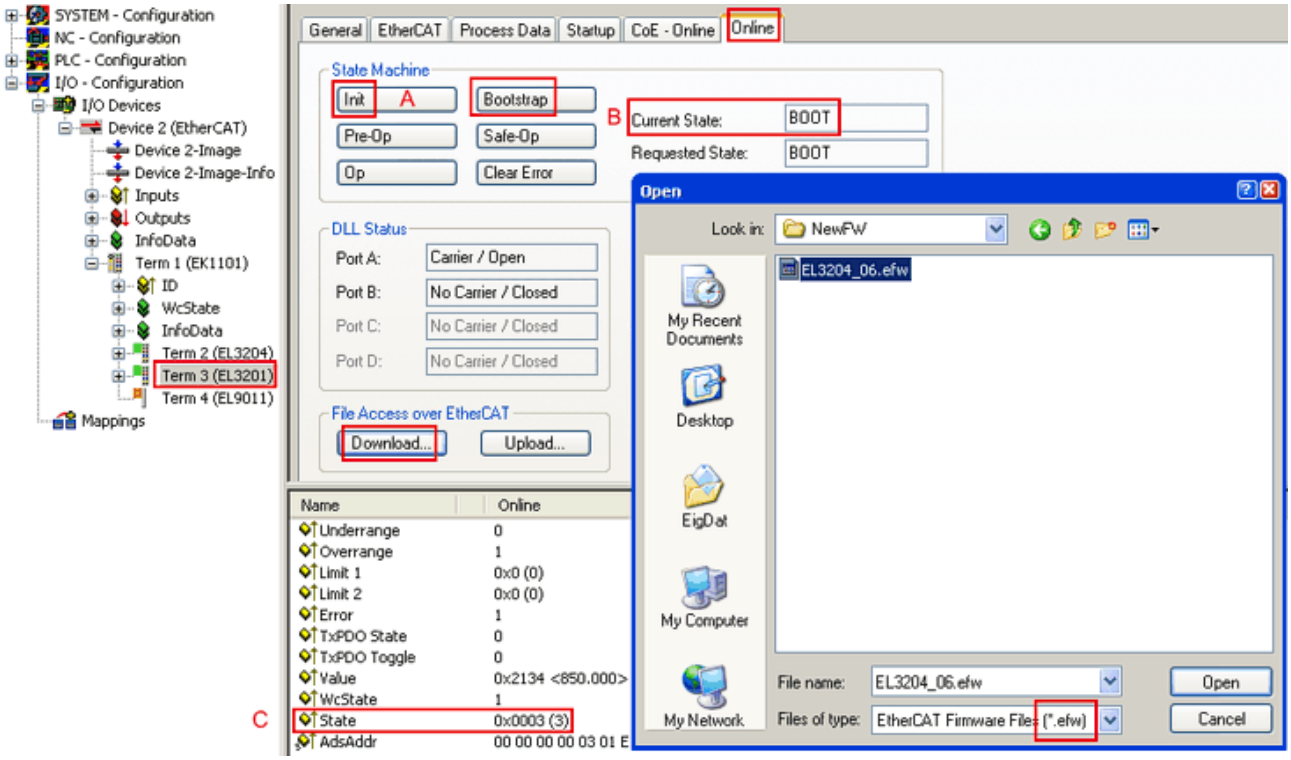
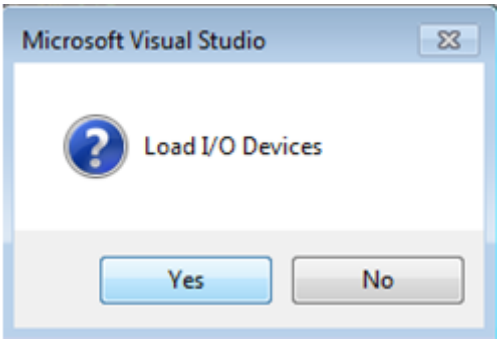
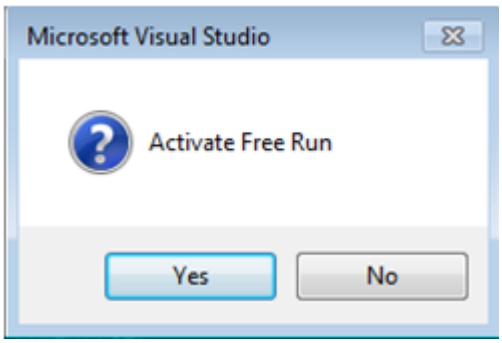


Fig. 186: Firmware Update

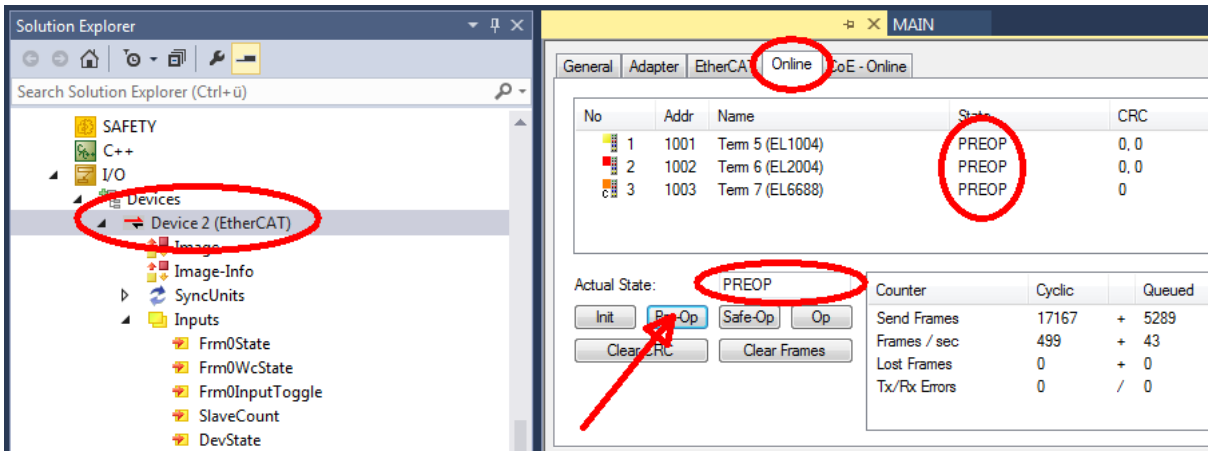
Proceed as follows, unless instructed otherwise by Beckhoff support. Valid for TwinCAT 2 and 3 as EtherCAT master.

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



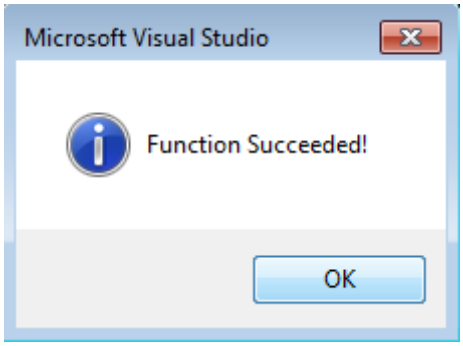


- Switch EtherCAT Master to PreOP



- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP

- Check the current status (B, C)
- Download the new *efw file (wait until it ends). A password will not be necessary usually.



- After the download switch to INIT, then PreOP
- Switch off the slave briefly (don't pull under voltage!)
- Check within CoE 0x100A, if the FW status was correctly overtaken.

9.5.4 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 TwinCAT System Manager

The TwinCAT System Manager indicates the FPGA firmware version. Click on the Ethernet card of your EtherCAT strand (Device 2 in the example) and select the *Online* tab.

The *Reg:0002* column indicates the firmware version of the individual EtherCAT devices in hexadecimal and decimal representation.

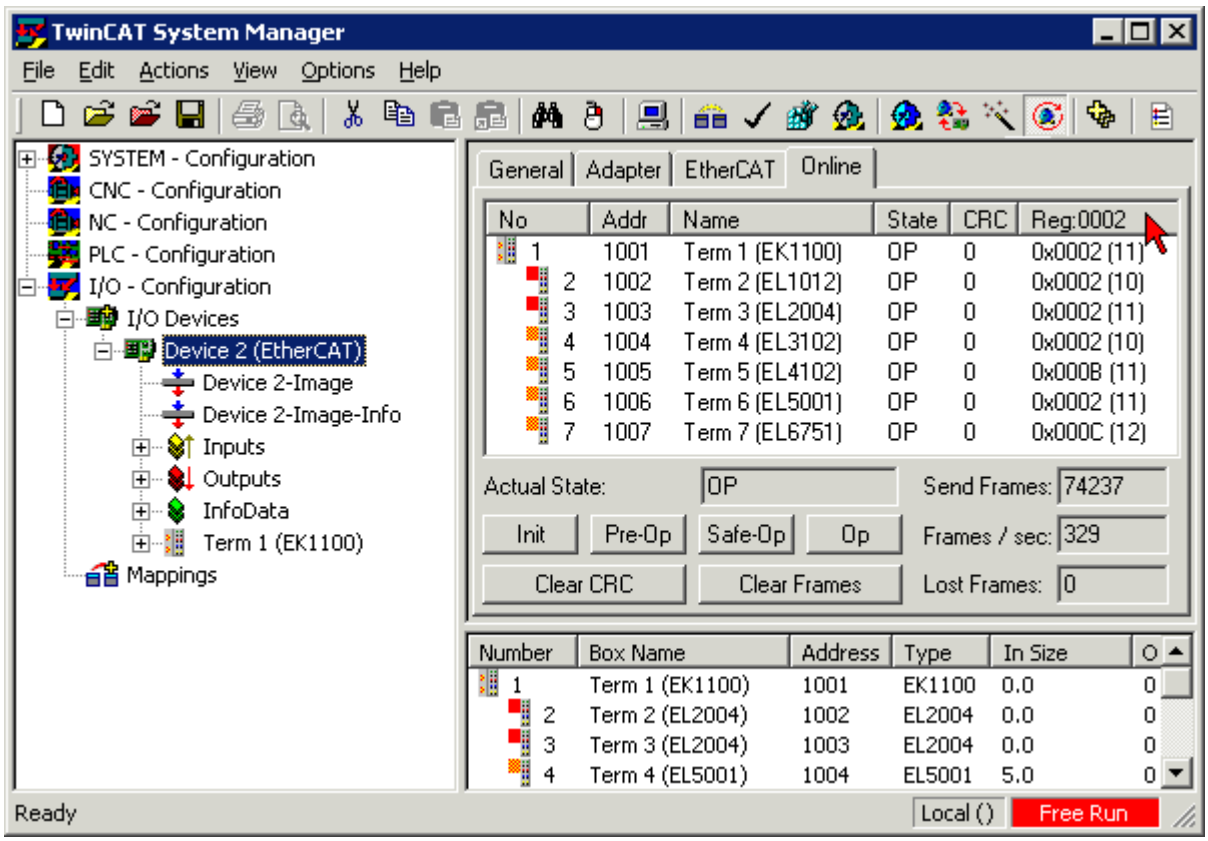


Fig. 187: FPGA firmware version definition

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

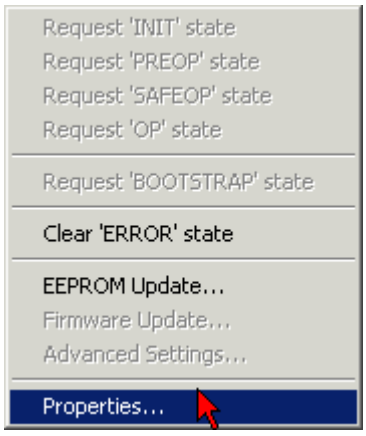


Fig. 188: Context menu *Properties*

The *Advanced Settings* dialog appears where the columns to be displayed can be selected. Under *Diagnosis/***Online View** select the *'0002 ETxxxx Build'* check box in order to activate the FPGA firmware version display.

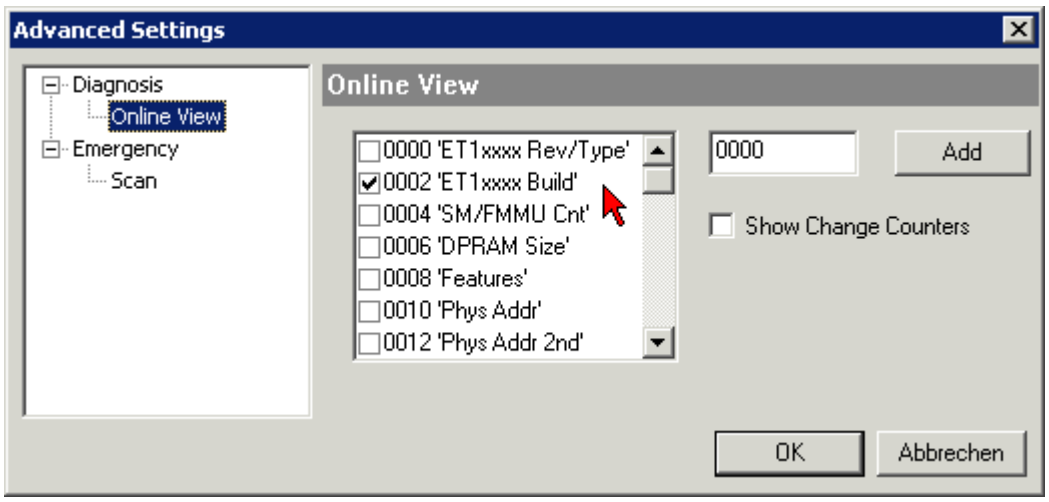


Fig. 189: Dialog *Advanced Settings*

Update

For updating the FPGA firmware

- of an EtherCAT coupler the coupler must have FPGA firmware version 11 or higher;
- of an E-Bus Terminal the terminal must have FPGA firmware version 10 or higher.

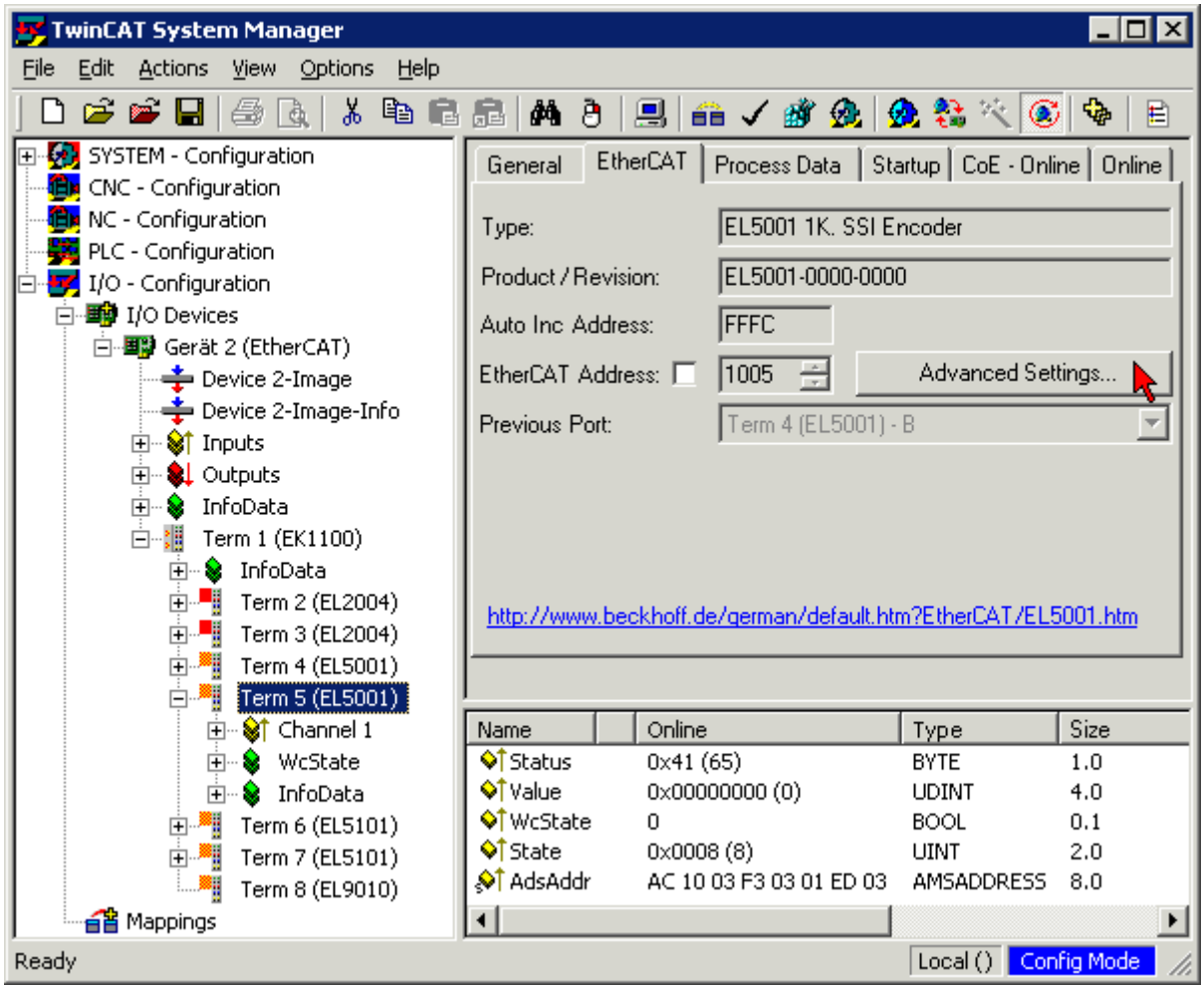
Older firmware versions can only be updated by the manufacturer!

Updating an EtherCAT device

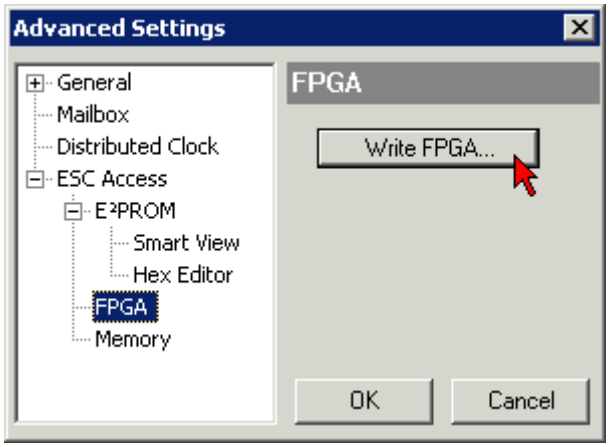
The following sequence order have to be met if no other specifications are given (e.g. by the Beckhoff support):

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

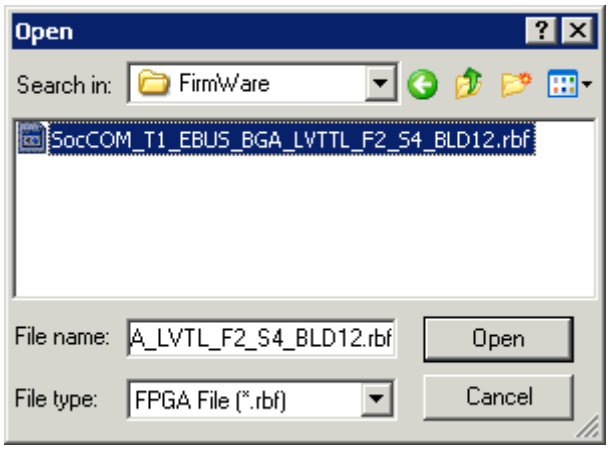
- In the TwinCAT System Manager select the terminal for which the FPGA firmware is to be updated (in the example: Terminal 5: EL5001) and
  click the *Advanced Settings* button in the *EtherCAT* tab:



- The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM*/FPGA click on *Write FPGA* button:



- Select the file (*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device:



- Wait until download ends
- Switch slave current less for a short time (don't pull under voltage!). In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.
- Check the new FPGA status

NOTICE

Risk of damage to the device!

A download of firmware to an EtherCAT device must not be interrupted in any case! If you interrupt this process by switching off power supply or disconnecting the Ethernet link, the EtherCAT device can only be recommissioned by the manufacturer!

9.5.5 Simultaneous updating of several EtherCAT devices

The firmware and ESI descriptions of several devices can be updated simultaneously, provided the devices have the same firmware file/ESI.

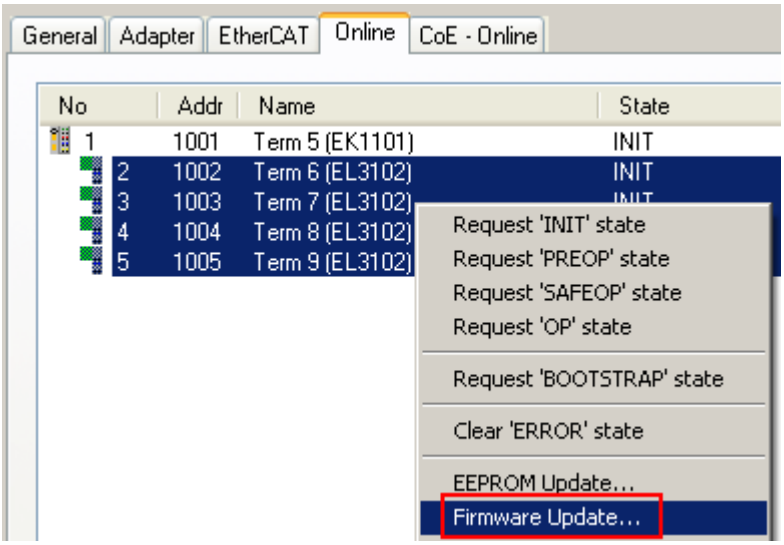


Fig. 190: Multiple selection and firmware update

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

9.6 Firmware compatibility

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

Note

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

NOTICE

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 237]. 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!

EPP3504-0023			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00	04	0016	2019/12
	05	0016	2020/03
	06	0017	2022/06
	07 *)	0018 *)	2026/02 *)

ERP3504-0022			
Hardware (HW)	**Firmware (FW)**	**Revision no.**	**Release date**
00	06	0016	2024/04
	07 *)	0017 *)	2026/02 *)

*) 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.

9.7 Restoring the delivery state

To restore the delivery state for backup objects in ELxxxx terminals / EPxxxx- and EPPxxxx box modules, the CoE object *Restore default parameters, SubIndex 001* can be selected in the TwinCAT System Manager (Config mode).

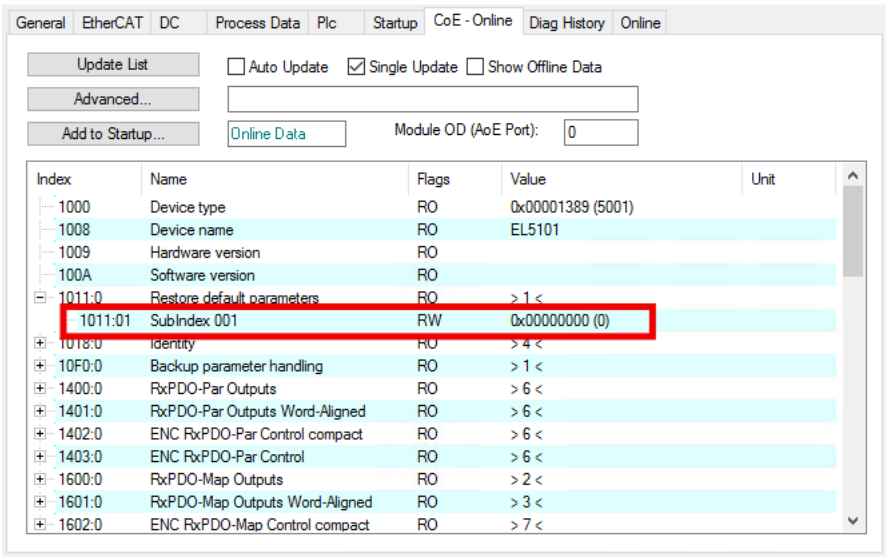


Fig. 191: Selecting the Restore default parameters PDO

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.

All backup objects are reset to the delivery state.

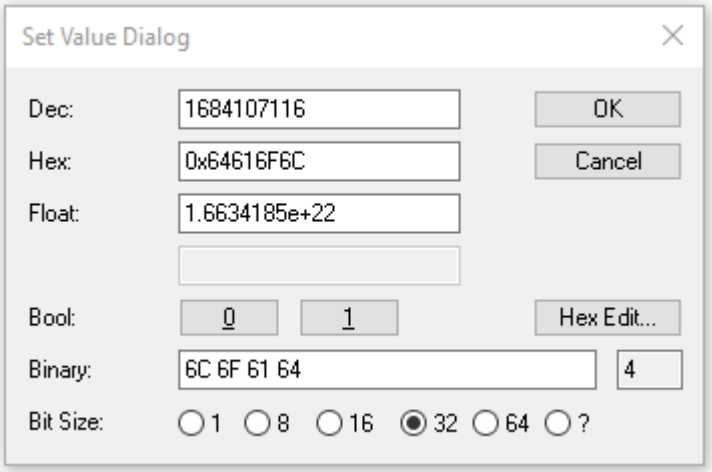


Fig. 192: Entering a restore value in the Set Value dialog

Alternative restore value

In some older terminals / boxes the backup objects can be switched with an alternative restore value:
Decimal value: 1819238756
Hexadecimal value: 0x6C6F6164

An incorrect entry for the restore value has no effect.

9.8 Notes on operation

EMC immunity according to EN 61000-6-2

In order to achieve immunity to electromagnetic interference in accordance with EN 61000-6-2, a digital low-pass filter must be used with which the peak-to-peak (PtP) noise is suppressed to maximally 1/3 of the specified basic accuracy. Such filtering is also useful from the point of view of measurement technology, as the basic accuracy given for the respective terminal is specified "with averaging", i.e. without the influence of white noise. Filtering is not necessary if the unfiltered PtP noise already lies below the 1/3 limit. It may also be useful to do without filtering when fast response times are required. However, it is then possible that the full specified basic accuracy is not achievable in all cases.

Here are two examples of this with fictitious numbers:

- Example 1:
  ELM3004-0000, ±10 V measuring range:
  - Noise without filtering $E_{Noise,PtP}$ < 60 ppm_{FSV}, basic accuracy ±100 ppm_{FSV}.
  - Because 60 < (1/3 · 200), no filtering is required.
- Example 2:
  ELM3004-0000, ±20 V measuring range:
  - Noise without filtering $E_{Noise,PtP}$ < 560 ppm_{FSV}, basic accuracy ±300 ppm_{FSV}.
  - Now 560 > (1/3 · 600), and a low-pass filter is required if the full basic accuracy is to be achieved.

9.9 Further documentation for I/O components with analog in and outputs

NOTICE	
	Further documentation for I/O components with analog in and outputs Also pay attention to the further documentation: **I/O Analog Manual** Notes on I/O components with analog inputs and outputs, which is available in the Beckhoff Information-System and for download on the Beckhoff website www.beckhoff.com on the respective product pages! The content includes the basics of sensor technology and information on analog measured values.

9.10 Support and Service

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

Beckhoff's branch offices and representatives

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

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

You will also find further documentation for Beckhoff components there.

Support

The Beckhoff Support offers you comprehensive technical assistance, helping you not only with the application of individual Beckhoff products, but also with other, wide-ranging services:

- support
- design, programming and commissioning of complex automation systems
- and extensive training program for Beckhoff system components

Hotline:	+49 5246 963 157
e-mail:	support@beckhoff.com
web:	www.beckhoff.com/support

Service

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

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

Hotline:	+49 5246 963 460
e-mail:	service@beckhoff.com
web:	www.beckhoff.com/service

Headquarters Germany

Beckhoff Automation GmbH & Co. KG

Hülshorstweg 20
33415 Verl
Germany

Phone:	+49 5246 963 0
e-mail:	info@beckhoff.com
web:	www.beckhoff.com

9.11 Reshipment and return

This product is individually packed and sealed. Unless otherwise agreed, Beckhoff can only accept returns in unopened original packaging with the seal intact.

List of illustrations

Fig. 1 BIC as data matrix code (DMC, code scheme ECC200) 12
Fig. 2 Example DMC 1P072222SBTNk4p562d71KEL1809 Q1 51S678294 13
Fig. 3 EPP3504-0023 17
Fig. 4 ERP3504-0022 18
Fig. 5 Overview measurement ranges, Bipolar 22
Fig. 6 Overview measurement ranges, Unipolar 22
Fig. 7 Representation ±10 V measurement range 24
Fig. 8 Representation ±80 mV measurement range 26
Fig. 9 Chart: RTD measuring range 29
Fig. 10 Concatenation of the uncertainties in RTD measurement 30
Fig. 11 Diagram for basic accuracy for Pt1000, 3-wire connection 31
Fig. 12 Diagram for basic accuracy for Pt1000, 4-wire connection 31
Fig. 13 Diagram noise ENoise, PtP in dependence on sensor temperature 33
Fig. 14 difference between 500 Ω and 5 kΩ potentiometer 34
Fig. 15 Representation potentiometer measurement range 37
Fig. 16 Connection of the quarter bridge 48
Fig. 17 Relationship between UBridge/UExc and ΔR1 49
Fig. 18 Basic range of a process data value 56
Fig. 19 Example: double click the CoE object PAI-Settings, Index Interface of channel 1 of an ELM3xxx terminal 64
Fig. 20 Function blocks (FBs) for CoE access to the TwinCAT TC2_EtherCAT.lib 65
Fig. 21 TwinCAT FilterControl embedded in the configuration dialog of an EPP35xx/ ERP35xx box module 65
Fig. 22 Filter coefficients Nos. 1 to 12 of channel 1 in the CoE online of an ELM3602 EtherCAT Terminal 66
Fig. 23 StartUp list of an ELM3602 EtherCAT Terminal, already partially modified by direct input 66
Fig. 24 Calling the ADS Monitor Command Test 67
Fig. 25 Activation of "EtherCAT Addr." 67
Fig. 26 Opening the *. tnzip archive 88
Fig. 27 Search of the existing HW configuration for the EtherCAT configuration of the example 88
Fig. 28 Process of interlacing the input data 98
Fig. 29 Configuration and setup for sample program 6: Doubling of the sample rate with 2 x EL3751 99
Fig. 30 Setting the DC shift time for terminal 2 99
Fig. 31 Oversampling 20 ksps with 2 x EL3751 with input signals (below) and result signal (top) 101
Fig. 32 Oversampling 20 ksps with 2 x EL3751 shows input value 1 and input value 2 alternately for a result value 101
Fig. 33 Decimation from 20 µs (left) to 22.675.. µs (right) with ELM3602 105
Fig. 34 Principle of combining two measuring channels with FSV1 and FSV2 108
Fig. 35 Possible structure for the "Measurement range combination" sample program 109
Fig. 36 Combination of two channels of the ELM3602-0002 with ±5 V and ±80 mV measuring range 110
Fig. 37 Combination of two channels of the ELM3602-0002: Supply of a delta voltage in the positive transition range 111
Fig. 38 Function block as an example for analysis of diagnostics information of the box module 112
Fig. 39 Creation of PDO variables (TwinCAT version >= V3.1.4024.0) 113
Fig. 40 Creation of the SmPdoVariables (TwinCAT version >= V3.1.4022.30) 113
Fig. 41 Seek the generated data type of SmPdoVariables 114

Fig. 42 Visualization of the sample implementation: Calibration signature ... 115
Fig. 43 Test data to illustrate the content of patManFactSpecIdCode[0]^ of the FB ... 116
Fig. 44 Selection of the diagnostic information of an EtherCAT Slave ... 119
Fig. 45 Basic EtherCAT Slave Diagnosis in the PLC ... 120
Fig. 46 EL3102, CoE directory ... 122
Fig. 47 Example of commissioning aid for a EL3204 ... 122
Fig. 48 Default behaviour of the System Manager ... 123
Fig. 49 Default target state in the Slave ... 124
Fig. 50 PLC function blocks ... 125
Fig. 51 Illegally exceeding the E-Bus current ... 125
Fig. 52 Warning message for exceeding E-Bus current ... 126
Fig. 53 Relationship between user side (commissioning) and installation ... 127
Fig. 54 Control configuration with Embedded PC, input (EL1004) and output (EL2008) ... 128
Fig. 55 Initial TwinCAT 2 user interface ... 129
Fig. 56 Selection of the target system ... 130
Fig. 57 specify the PLC for access by the TwinCAT System Manager: selection of the target system ... 130
Fig. 58 Select “Scan Devices...” ... 131
Fig. 59 Automatic detection of I/O devices: selection of the devices to be integrated ... 131
Fig. 60 Mapping of the configuration in the TwinCAT 2 System Manager ... 132
Fig. 61 Reading of individual terminals connected to a device ... 132
Fig. 62 TwinCAT PLC Control after startup ... 133
Fig. 63 Example program with variables after a compile process (without variable integration) ... 134
Fig. 64 Appending the TwinCAT PLC Control project ... 134
Fig. 65 PLC project integrated in the PLC configuration of the System Manager ... 135
Fig. 66 Creating the links between PLC variables and process objects ... 135
Fig. 67 Selecting BOOL-type PDO ... 136
Fig. 68 Selecting several PDOs simultaneously: activate “Continuous” and “All types” ... 136
Fig. 69 Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example ... 137
Fig. 70 Choose target system (remote) ... 138
Fig. 71 PLC Control logged in, ready for program startup ... 139
Fig. 72 Initial TwinCAT 3 user interface ... 140
Fig. 73 Create new TwinCAT 3 project ... 140
Fig. 74 New TwinCAT 3 project in the project folder explorer ... 141
Fig. 75 Selection dialog: Choose the target system ... 141
Fig. 76 specify the PLC for access by the TwinCAT System Manager: selection of the target system ... 142
Fig. 77 Select “Scan” ... 142
Fig. 78 Automatic detection of I/O devices: selection of the devices to be integrated ... 143
Fig. 79 Mapping of the configuration in VS shell of the TwinCAT 3 environment ... 143
Fig. 80 Reading of individual terminals connected to a device ... 144
Fig. 81 Adding the programming environment in “PLC” ... 145
Fig. 82 Specifying the name and directory for the PLC programming environment ... 145
Fig. 83 Initial “Main” program for the standard PLC project ... 146
Fig. 84 Example program with variables after a compile process (without variable integration) ... 147
Fig. 85 Start program compilation ... 147
Fig. 86 Creating the links between PLC variables and process objects ... 148
Fig. 87 Selecting BOOL-type PDO ... 148

Fig. 88 Selecting several PDOs simultaneously: activate “Continuous” and “All types” 149
Fig. 89 Application of a “Goto Link Variable”, using “MAIN.bEL1004_Ch4” as an example 149
Fig. 90 Creating a PLC data type 150
Fig. 91 Instance_of_struct 150
Fig. 92 Linking the structure 151
Fig. 93 Reading a variable from the structure of the process data 151
Fig. 94 TwinCAT 3 development environment (VS shell): logged-in, after program startup 152
Fig. 95 System Manager “Options” (TwinCAT 2) 154
Fig. 96 Call up under VS Shell (TwinCAT 3) 154
Fig. 97 TcRteInstall in the TwinCAT directory 154
Fig. 98 Overview of network interfaces 155
Fig. 99 EtherCAT device properties (TwinCAT 2): click on “Compatible Devices…” of tab “Adapter” 155
Fig. 100 Windows properties of the network interface 156
Fig. 101 Exemplary correct driver setting for the Ethernet port 156
Fig. 102 Incorrect driver settings for the Ethernet port 157
Fig. 103 TCP/IP setting for the Ethernet port 158
Fig. 104 Identifier structure 159
Fig. 105 OnlineDescription information window (TwinCAT 2) 160
Fig. 106 Information window OnlineDescription (TwinCAT 3) 160
Fig. 107 File OnlineDescription.xml created by the System Manager 161
Fig. 108 Indication of an online recorded ESI of EL2521 as an example 161
Fig. 109 Information window for faulty ESI file (left: TwinCAT 2; right: TwinCAT 3) 161
Fig. 110 Calling the ESI Updater (TwinCAT 3) 163
Fig. 111 (ESI) Updater 163
Fig. 112 Using the ESI Updater (>= TwinCAT 2.11) 165
Fig. 113 Append EtherCAT device (left: TwinCAT 2; right: TwinCAT 3) 166
Fig. 114 Selecting the EtherCAT connection (TwinCAT 2.11, TwinCAT 3) 166
Fig. 115 Selecting the Ethernet port 166
Fig. 116 EtherCAT device properties (TwinCAT 2) 167
Fig. 117 Appending EtherCAT devices (left: TwinCAT 2; right: TwinCAT 3) 167
Fig. 118 Selection dialog for new EtherCAT device 168
Fig. 119 Display of device revision 168
Fig. 120 Display of previous revisions 169
Fig. 121 Name/revision of the terminal 169
Fig. 122 EtherCAT terminal in the TwinCAT tree (left: TwinCAT 2; right: TwinCAT 3) 170
Fig. 123 Differentiation local/target system (left: TwinCAT 2; right: TwinCAT 3) 171
Fig. 124 Scan Devices (left: TwinCAT 2; right: TwinCAT 3) 171
Fig. 125 Note for automatic device scan (left: TwinCAT 2; right: TwinCAT 3) 171
Fig. 126 Detected Ethernet devices 172
Fig. 127 Example default state 172
Fig. 128 Installing EthetCAT terminal with revision -1018 173
Fig. 129 Detection of EtherCAT terminal with revision -1019 173
Fig. 130 Scan query after automatic creation of an EtherCAT device (left: TwinCAT 2; right: TwinCAT 3) 173
Fig. 131 Manual scanning for devices on a specified EtherCAT device (left: TwinCAT 2; right: TwinCAT 3) 174

Fig. 132 Scan progressexemplary by TwinCAT 2 174
Fig. 133 Config/FreeRun query (left: TwinCAT 2; right: TwinCAT 3) 174
Fig. 134 Displaying of “Free Run” and “Config Mode” toggling right below in the status bar 174
Fig. 135 TwinCAT can also be switched to this state by using a button (left: TwinCAT 2; right: TwinCAT 3) 174
Fig. 136 Online display example 175
Fig. 137 Faulty identification 175
Fig. 138 Identical configuration (left: TwinCAT 2; right: TwinCAT 3) 176
Fig. 139 Correction dialog 176
Fig. 140 Name/revision of the terminal 177
Fig. 141 Correction dialog with modifications 178
Fig. 142 Dialog “Change to Compatible Type…” (left: TwinCAT 2; right: TwinCAT 3) 178
Fig. 143 TwinCAT 2 Dialog Change to Alternative Type 179
Fig. 144 Branch element as terminal EL3751 179
Fig. 145 “General” tab 179
Fig. 146 “EtherCAT” tab 180
Fig. 147 “Process Data” tab 181
Fig. 148 Configuring the process data 182
Fig. 149 “Startup” tab 183
Fig. 150 “CoE - Online” tab 184
Fig. 151 Dialog “Advanced settings” 185
Fig. 152 “Online” tab 185
Fig. 153 “DC” tab (Distributed Clocks) 186
Fig. 154 System manager current calculation 197
Fig. 155 eEtherCAT tab -> Advanced Settings -> Behavior -> Watchdog 199
Fig. 156 States of the EtherCAT State Machine 201
Fig. 157 “CoE Online” tab 203
Fig. 158 Startup list in the TwinCAT System Manager 204
Fig. 159 Offline list 205
Fig. 160 Online list 206
Fig. 161 CoE index 10F0 207
Fig. 162 Dimensions EPP3504-0023, ERP3504-0022 (values in mm) 209
Fig. 163 Connection for functional earth (FE) EPP3504-0023 211
Fig. 164 Connection for functional earth (FE) ERP3504-0022 211
Fig. 165 Connectors for EtherCAT P 217
Fig. 166 M8 socket, P-coded 217
Fig. 167 EPP3504-0023 LEDs RUN, Channel 221
Fig. 168 Status LEDs for the power supply 222
Fig. 169 Status LEDs for ethercat 222
Fig. 170 DiagMessages in the CoE 226
Fig. 171 Implementation of the DiagMessage system in the TwinCAT System Manager 227
Fig. 172 Startup List 227
Fig. 173 Schematic representation TCEventLogger 234
Fig. 174 Display EventLogger window 235
Fig. 175 Display DiagMessages and Logged Events 236
Fig. 176 Setting filter language 236

Fig. 177 Activating/deactivating event transmission 237
Fig. 178 Settings TwinCAT EventLogger 237
Fig. 179 Device identifier consisting of name EL3204-0000 and revision -0016 239
Fig. 180 Scan the subordinate field by right-clicking on the EtherCAT device 240
Fig. 181 Configuration is identical 240
Fig. 182 Change dialog 240
Fig. 183 EEPROM Update 241
Fig. 184 Selecting the new ESI 241
Fig. 185 Display of EL3204 firmware version 242
Fig. 186 Firmware Update 243
Fig. 187 FPGA firmware version definition 245
Fig. 188 Context menu Properties 245
Fig. 189 Dialog Advanced Settings 246
Fig. 190 Multiple selection and firmware update 248
Fig. 191 Selecting the Restore default parameters PDO 250
Fig. 192 Entering a restore value in the Set Value dialog 250

Trademark statements

Beckhoff®, ATRO®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, MX-System®, Safety over EtherCAT®, TC/BSD®, TwinCAT®, TwinCAT/BSD®, TwinSAFE®, XFC®, XPlanar® and XTS® are registered and licensed trademarks of Beckhoff Automation GmbH.

Third-party trademark statements

CANopen and CANopen FD are registered trademarks of CAN in AUTOMATION - International Users and Manufacturers Group e.V.

DeviceNet and EtherNet/IP are trademarks of ODVA, Inc.

DSP System Toolbox, Embedded Coder, MATLAB, MATLAB Coder, MATLAB Compiler, MathWorks, Predictive Maintenance Toolbox, Simscape, Simscape™ Multibody™, Simulink, Simulink Coder, Stateflow and ThingSpeak are registered trademarks of The MathWorks, Inc.

EnDat is a trademark of Dr. Johannes Heidenhain GmbH.

Excel, IntelliSense, Microsoft, Microsoft Azure, Microsoft Edge, PowerShell, Visual Studio, Windows and Xbox are trademarks of the Microsoft group of companies.

MAX®, Stratix®, Cyclone®, Altera®, Agilex™, Arria®, Intel, the Intel logo, Intel Core, Xeon, Intel Atom, Celeron and Pentium are trademarks of Intel Corporation or its subsidiaries.

More Information:

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Germany
Phone: +49 5246 9630
info@beckhoff.com
www.beckhoff.com