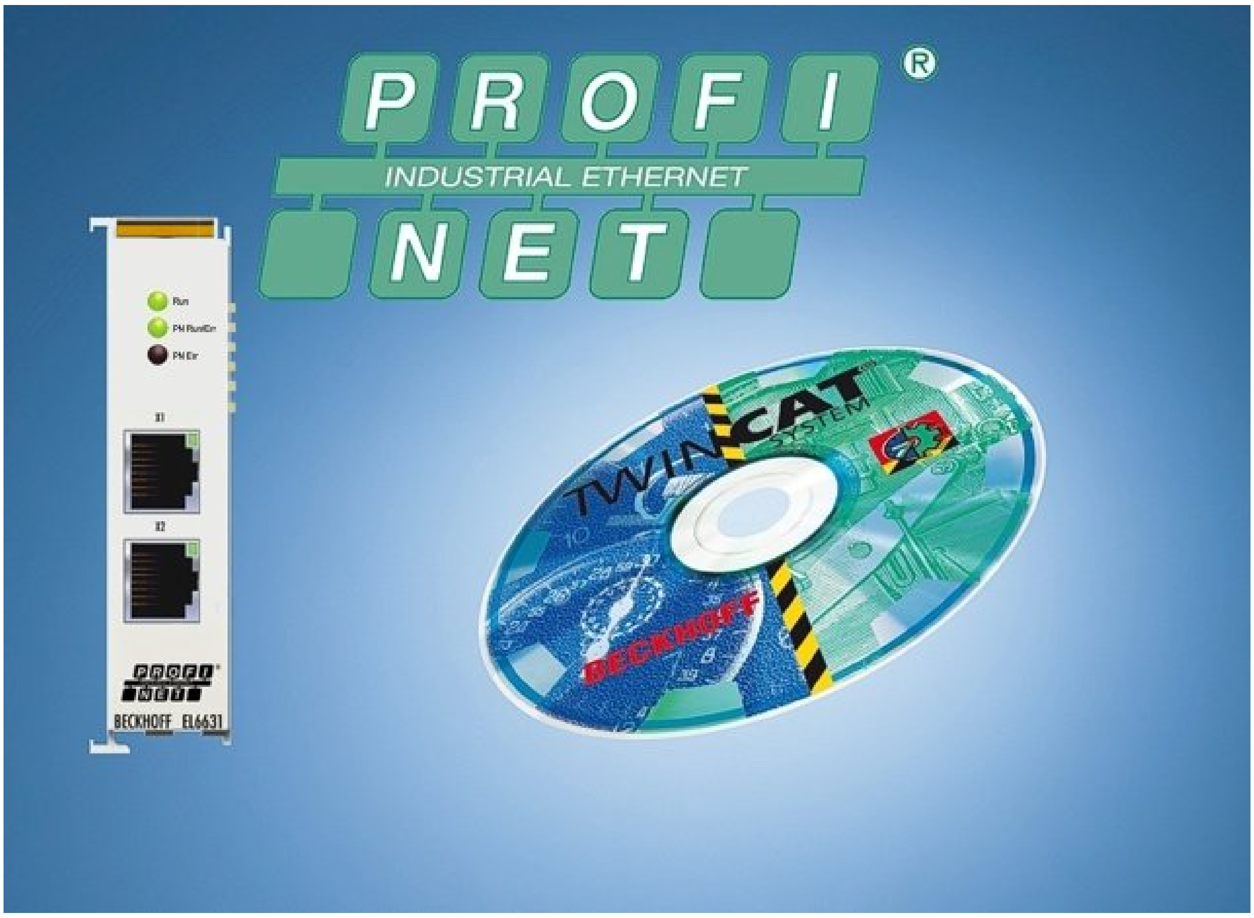


Documentation

EL6631, EL6632

PROFINET Controller Supplement

Version: 2.1
Date: 2017-04-04

BECKHOFF

Table of contents

1 Foreword

1.1 Notes on the documentation

Intended audience

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

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

Disclaimer

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

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

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

Trademarks

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

Patent Pending

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

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

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

1.2 Safety instructions

Safety regulations

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

Exclusion of liability

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

Personnel qualification

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

Description of symbols

In this documentation the following symbols are used with an accompanying safety instruction or note. The safety instructions must be read carefully and followed without fail!

⚠ DANGER	**Serious risk of injury!** Failure to follow the safety instructions associated with this symbol directly endangers the life and health of persons.
⚠ WARNING	**Risk of injury!** Failure to follow the safety instructions associated with this symbol endangers the life and health of persons.
⚠ CAUTION	**Personal injuries!** Failure to follow the safety instructions associated with this symbol can lead to injuries to persons.
❗ Attention	**Damage to the environment or devices** Failure to follow the instructions associated with this symbol can lead to damage to the environment or equipment.
ℹ Note	**Tip or pointer** This symbol indicates information that contributes to better understanding.

1.3 Documentation issue status

Version	Comment
2.1	• Update chapter “Acyclic data” • Update chapter “Technical data” • Update structure • Update revision status
2.0	• Migration • Update structure • Update revision status
1.0	• Addenda and 1st public issue
0.4	• Amendments
0.3	• Revised controller documentation
0.2	• Revised controller documentation
0.1	• First version

1.4 Version identification of EtherCAT devices

Designation

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

- family key
- type
- version
- revision

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

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 website. From 2014/01 the revision is shown on the outside of the IP20 terminals, see Fig. *“EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)”*.

- The type, version and revision are read as decimal numbers, even if they are technically saved in hexadecimal.

Identification number

Beckhoff EtherCAT devices from the different lines have different kinds of identification numbers:

Production lot/batch number/serial number/date code/D number

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

Structure of the serial number: **KK YY FF HH**

KK - week of production (CW, calendar week)
YY - year of production
FF - firmware version
HH - hardware version

Example with
Ser. no.: 12063A02:   12 - production week 12 06 - production year 2006 3A - firmware version 3A 02 - hardware version 02

Exceptions can occur in the **IP67 area**, where the following syntax can be used (see respective device documentation):

Syntax: D ww yy x y z u

D - prefix designation
ww - calendar week
yy - year
x - firmware version of the bus PCB
y - hardware version of the bus PCB
z - firmware version of the I/O PCB
u - hardware version of the I/O PCB

Example: D.22081501 calendar week 22 of the year 2008 firmware version of bus PCB: 1 hardware version of bus PCB: 5 firmware version of I/O PCB: 0 (no firmware necessary for this PCB) hardware version of I/O PCB: 1

Unique serial number/ID, ID number

In addition, in some series each individual module has its own unique serial number.

See also the further documentation in the area

- IP67: EtherCAT Box
- Safety: TwinSafe
- Terminals with factory calibration certificate and other measuring terminals

Examples of markings:

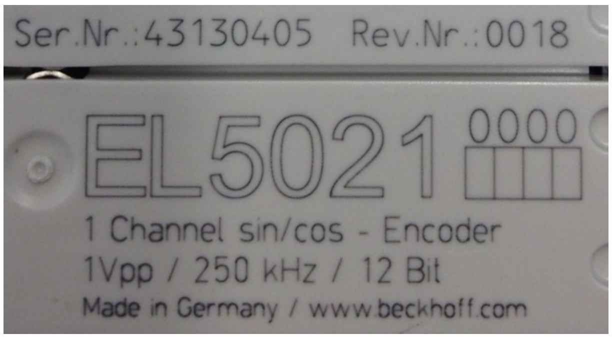

Ser.Nr.:43130405 Rev.Nr.:0018
EL5021 0000
1 Channel sin/cos - Encoder
1Vpp / 250 kHz / 12 Bit
Made in Germany / www.beckhoff.com

Fig. 1: *EL5021 EL terminal, standard IP20 IO device with batch number and revision ID (since 2014/01)*

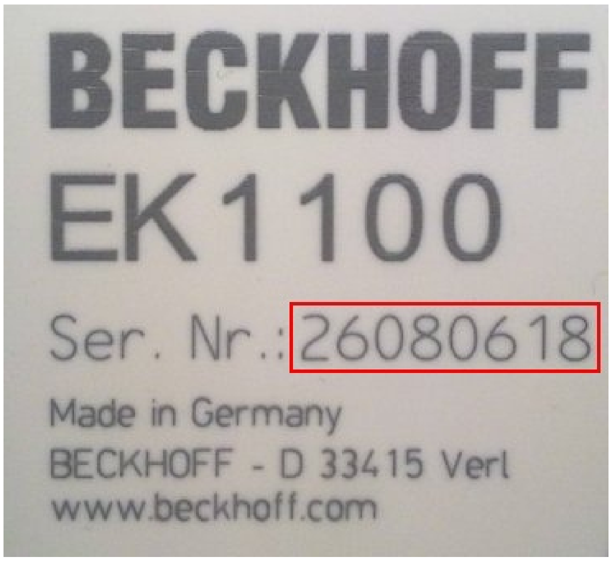


Fig. 2: *EK1100 EtherCAT coupler, standard IP20 IO device with batch number*

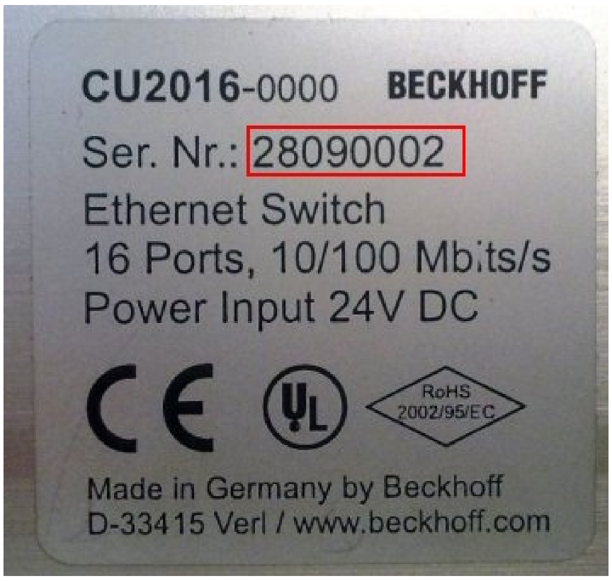


Fig. 3: *CU2016 switch with batch number*

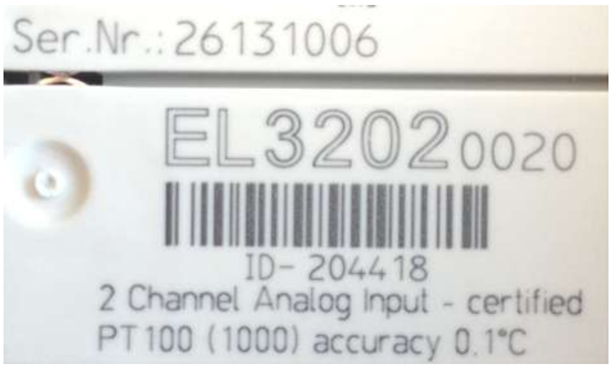

Ser.Nr.: 26131006
EL3202 0020
ID- 204418
2 Channel Analog Input - certified
PT100 (1000) accuracy 0.1°C

Fig. 4: *EL3202-0020 with batch numbers 26131006 and unique ID-number 204418*

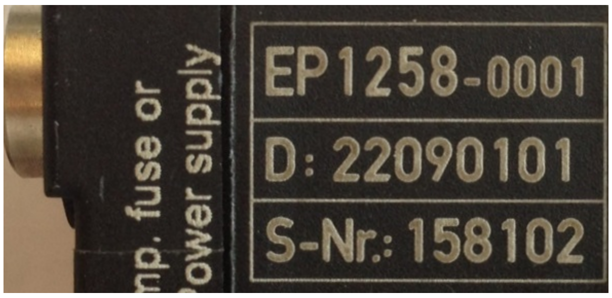


Fig. 5: *EP1258-00001 IP67 EtherCAT Box with batch number 22090101 and unique serial number 158102*

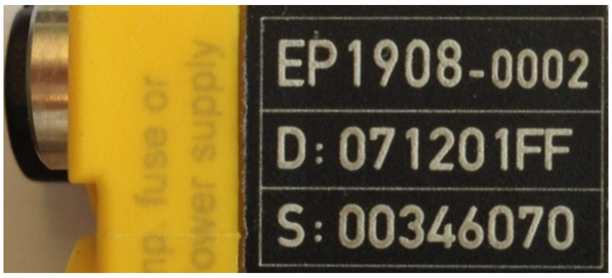


Fig. 6: *EP1908-0002 IP76 EtherCAT Safety Box with batch number 071201FF and unique serial number 00346070*

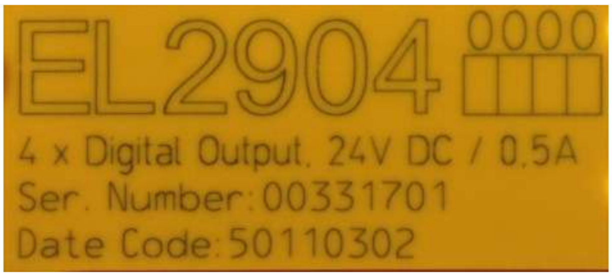


Fig. 7: *EL2904 IP20 safety terminal with batch number/date code 50110302 and unique serial number 00331701*

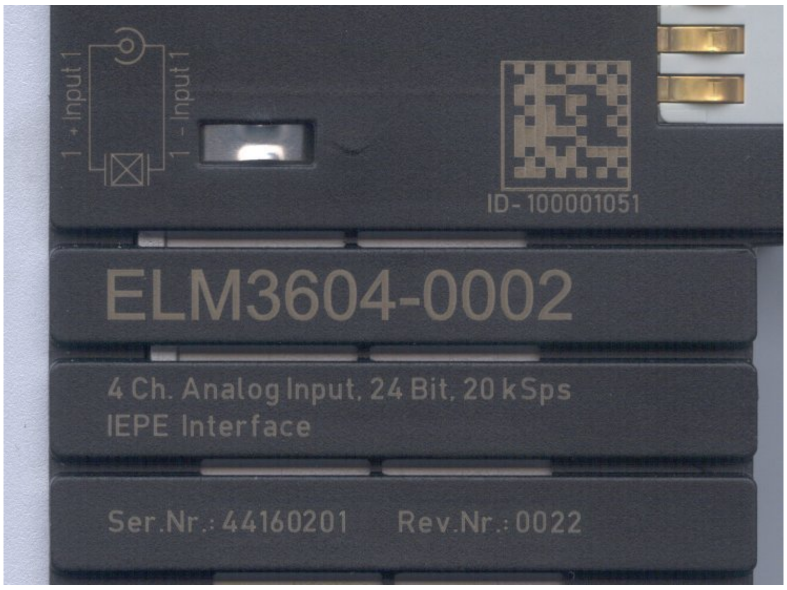


Fig. 8: *ELM3604-0002 terminal with ID number (QR code) 100001051 and unique serial number 44160201*

2 Product overview

2.1 Introduction

The PROFINET IO controller (master) is a TwinCAT supplement that turns every Beckhoff PC-based control system into a PROFINET IO controller. By installing the supplement, a standard Ethernet interface becomes a PROFINET master. This supplement can be used on PCs and Embedded PCs.

PROFINET can be simply coupled with EtherCAT in connection with the EL663x PROFINET terminal for the EtherCAT I/O system, wherein the terminal represents a slave on the EtherCAT side but acts as a controller towards PROFINET. In this way any EtherCAT network can also exchange data with PROFINET IO devices.

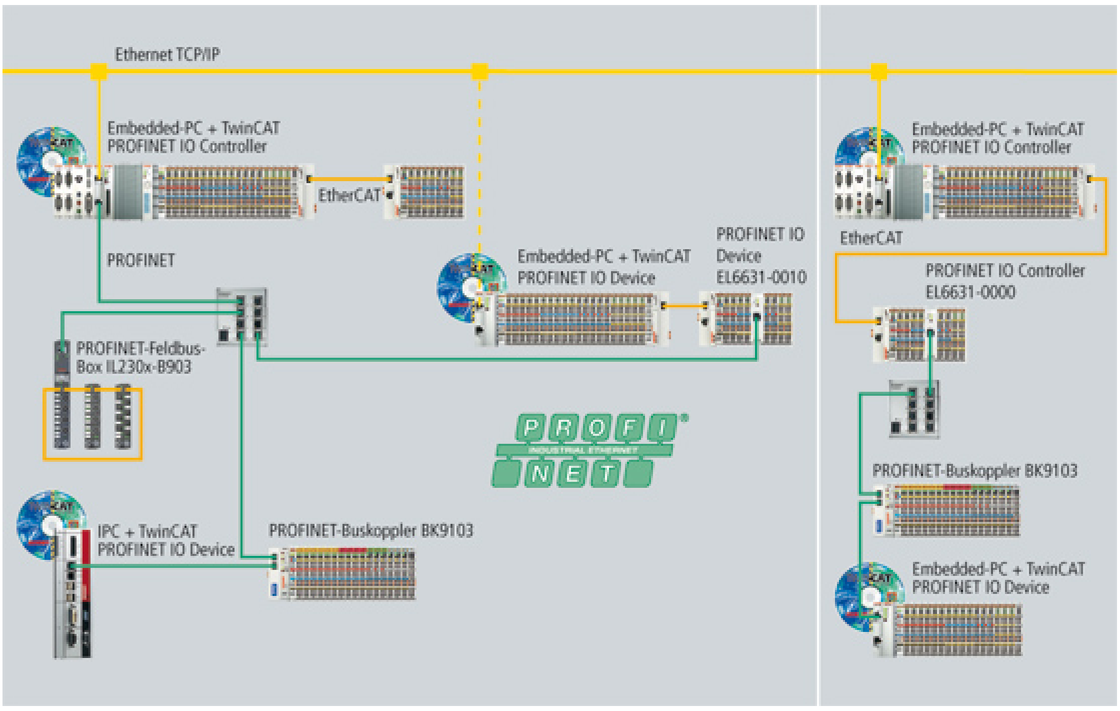

Ethernet TCP/IP
Embedded-PC + TwinCAT PROFINET IO Controller
EtherCAT
PROFINET
Embedded-PC + TwinCAT PROFINET IO Device
PROFINET IO Device EL6631-0010
PROFINET-Feldbus-Box IL230x-B903
PROFINET
IPC + TwinCAT PROFINET IO Device
PROFINET-Buskoppler BK9103
Embedded-PC + TwinCAT PROFINET IO Controller
EtherCAT
PROFINET IO Controller EL6631-0000
PROFINET-Buskoppler BK9103
Embedded-PC + TwinCAT PROFINET IO Device

Fig. 9: PROFINET topology example

There are two kinds of PROFINET: PROFINET I/O and PROFINET CBA. TwinCAT supports PROFINET I/O.

In turn, 4 kinds of cyclic communication are defined within PROFINET I/O: RTClass1 to 3 and RToverUDP. The TwinCAT PROFINET I/O controller and the EL6631 presently support RTClass1, while the EL6632 can communicate over RTClass1 and RTClass3. All cycle times defined for RTClass1 are supported, from 1 ms in power-of-2 steps (1, 2, 4, 8,… ms).

The smallest cycle time presently supported by the EL6632 is 500 µs (for RTClass3).

2.2 Technical data

Technical data	PROFINET via RT Ethernet	PROFINET via EtherCAT (EL6632)	PROFINET via EtherCAT (EL6631)
Ethernet Hardware	Real-Time Ethernet Hardware	EL6632 PROFINET terminal	EL6631 PROFINET terminal
Operating system	Windows XP, XP Embedded (CE under preparation)		
Software	TwinCAT I/O, PLC, NC, NC I, CNC		
Target systems	PC (x86), Windows CE devices	PC (x86), Windows CE devices with EtherCAT interface	PC (x86), Windows CE devices with EtherCAT interface
Cycle time	min. 1 ms	min. 500 us	min. 1 ms
Number of possible IO devices	limited by CPU performance and memory	for IRT up to 5 devices, for RT up to 15 devices	Up to 15 devices

2.3 Activation of the PROFINET Controller Supplement

Requirement:

TwinCAT 2.11 Build 1545
Hardware: Ethernet card with Intel chipset

An Ethernet card with Intel chipset is required in order to use the PROFINET controller Supplement. Install the RT-Ethernet driver for this card (TwinCAT Ethernet driver - Installation). Once the installation of the driver has been completed you can create the PROFINET controller in the system manager and connect it to the hardware (Ethernet card).

The PROFINET controller can be scanned in Config mode (TwinCAT icon is blue) and values can be written in Freerun mode. The appropriate key must be entered in order to operate TwinCAT in Run mode.
In order to obtain the registration key, send the MAC address of the Ethernet card you are using with the order. You will then receive the activation code.

MAC address

You can read the MAC address via the System Manager. To do this, append the PROFINET controller device.

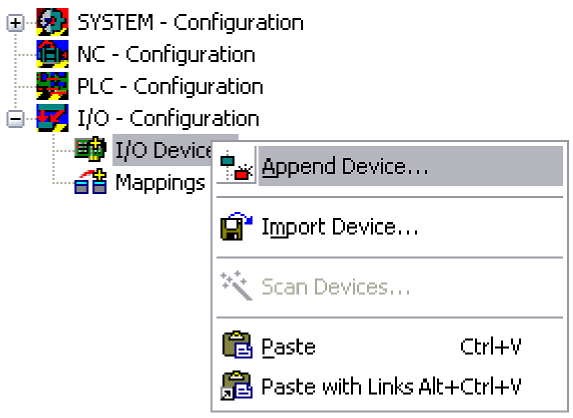
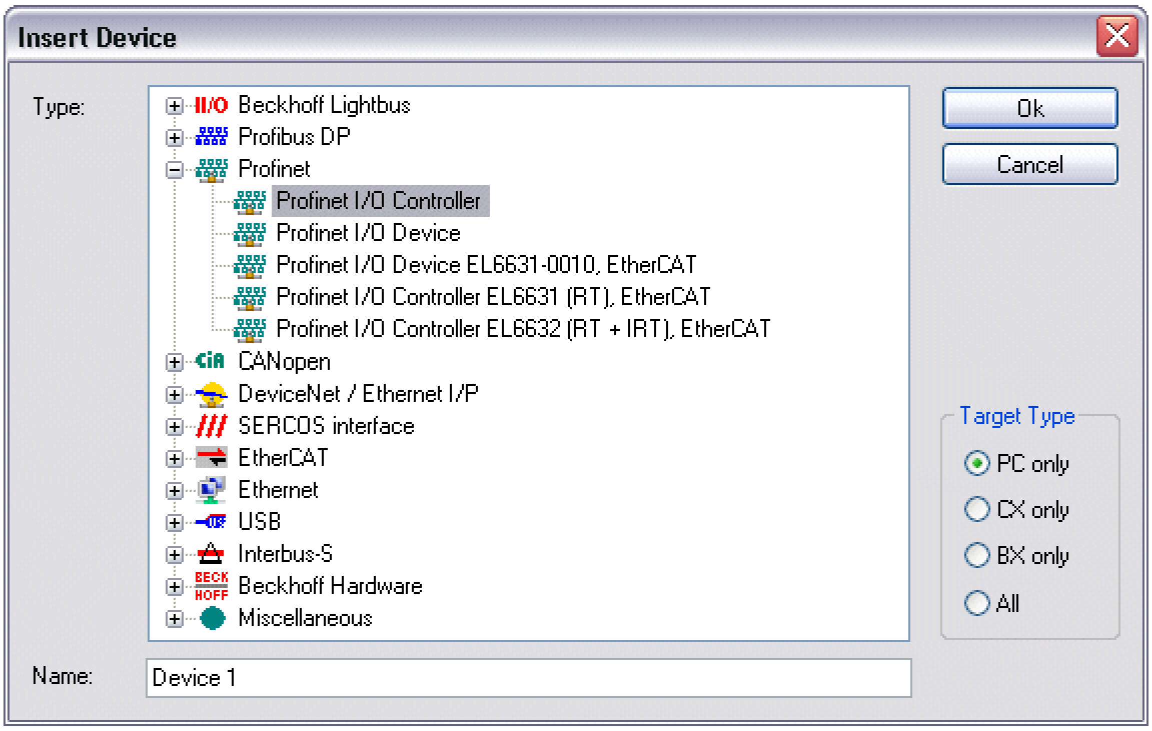


Fig. 10: "Append Device" context menu

Select the PROFINET I/O controller under PROFINET.


Insert Device

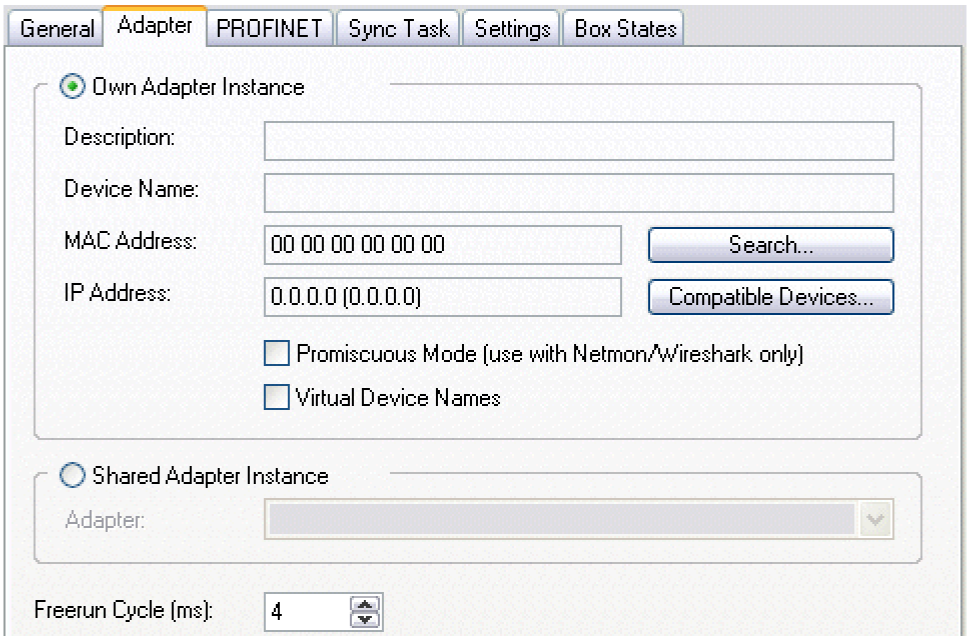
Type:
- Beckhoff Lightbus
- Profibus DP
- Profinet
  - Profinet I/O Controller
  - Profinet I/O Device
  - Profinet I/O Device EL6631-0010, EtherCAT
  - Profinet I/O Controller EL6631 (RT), EtherCAT
  - Profinet I/O Controller EL6632 (RT + IRT), EtherCAT
- CANopen
- DeviceNet / Ethernet I/P
- SERCOS interface
- EtherCAT
- Ethernet
- USB
- Interbus-S
- Beckhoff Hardware
- Miscellaneous

Name: Device 1

Target Type: [✓ PC only] CX only BX only All

Fig. 11: PROFINET Controller/Device selection

Switch to Adapter and press the search button.



[✓ Own Adapter Instance]

Description:

Device Name:

MAC Address: 00 00 00 00 00 00

IP Address: 0.0.0.0 (0.0.0.0)

☐ Promiscuous Mode (use with Netmon/Wireshark only)

☐ Virtual Device Names

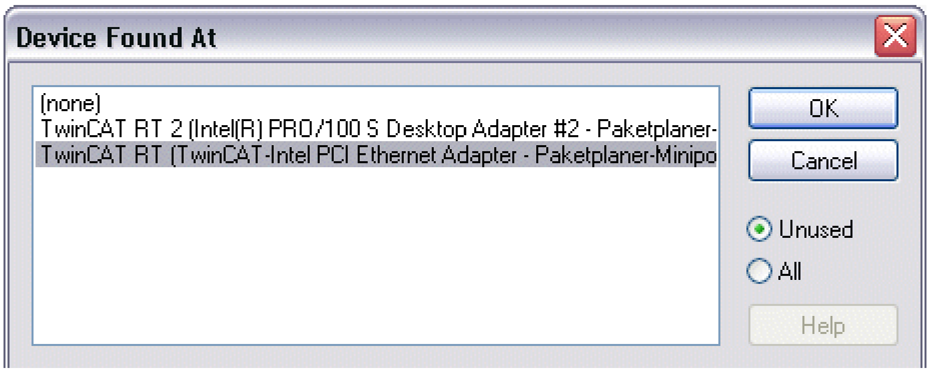
Shared Adapter Instance

Adapter:

Freerun Cycle (ms): 4

Fig. 12: "Adapter" tab, Search button

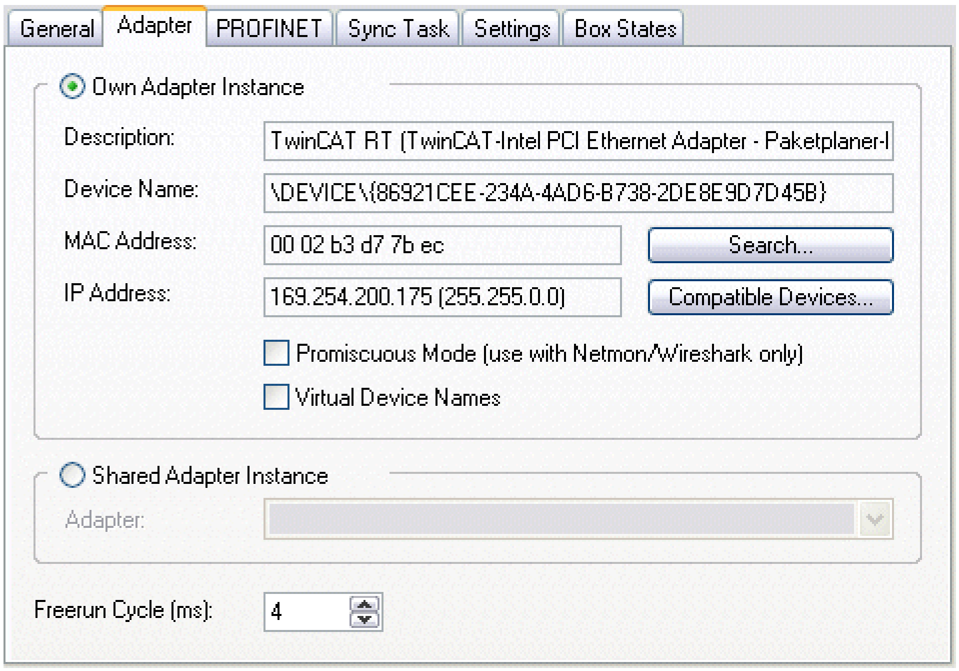
If the real-time driver has been installed correctly, you will be shown the corresponding cards. If no cards are available for selection, this means that the driver has not been correctly installed.
Select the card to which your PROFINET devices are connected.


Device Found At

(none)
TwinCAT RT 2 (Intel(R) PRO/100 S Desktop Adapter #2 - Paketplaner-
TwinCAT RT (TwinCAT-Intel PCI Ethernet Adapter - Paketplaner-Minipo

Fig. 13: TWINCAT network adapter selection

The MAC address of the card will be displayed under Adapter -> MAC address. This is now required for the key.



[✓ Own Adapter Instance]

Description: TwinCAT RT (TwinCAT-Intel PCI Ethernet Adapter - Paketplaner-I

Device Name: \DEVICE\{86921CEE-234A-4AD6-B738-2DE8E9D7D45B}

MAC Address: 00 02 b3 d7 7b ec

IP Address: 169.254.200.175 (255.255.0.0)

Promiscuous Mode (use with Netmon/Wireshark only)

Virtual Device Names

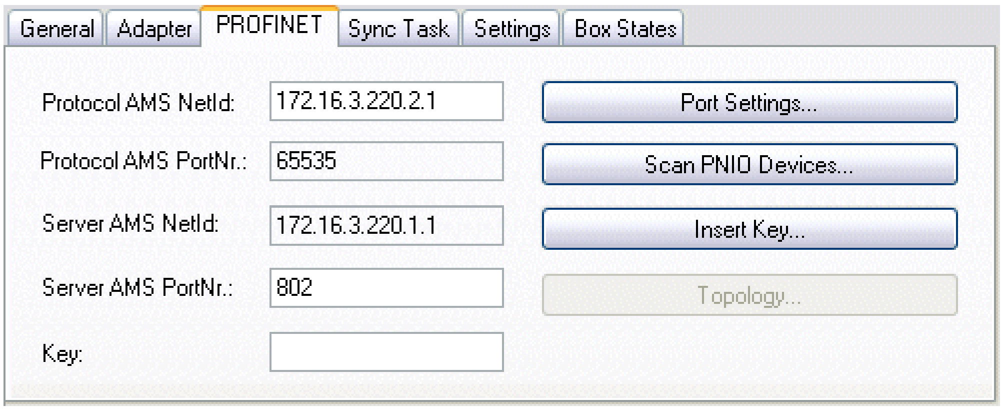
Shared Adapter Instance

Adapter:

Freerun Cycle (ms): 4

Fig. 14: Display of the MAC address

Switch to the PROFINET tab and press ‘Insert key…’. Now enter the key.



Protocol AMS NetId: 172.16.3.220.2.1

Protocol AMS PortNr.: 65535

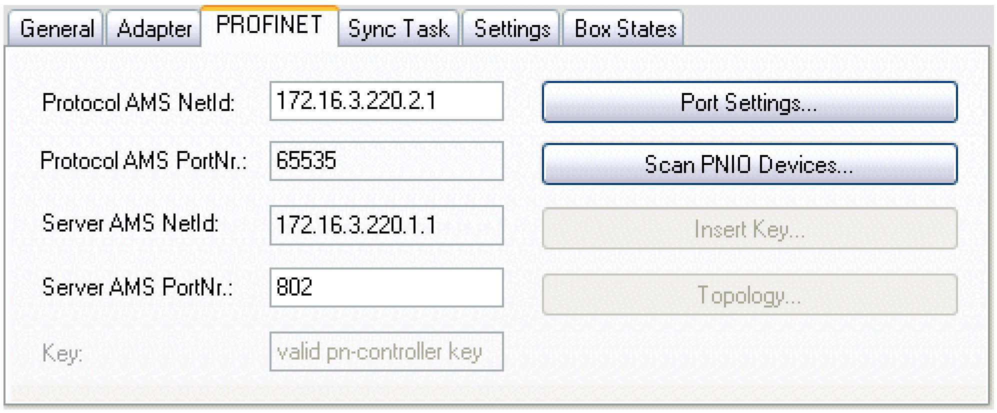
Server AMS NetId: 172.16.3.220.1.1

Server AMS PortNr.: 802

Key:

Fig. 15: Entry of the MAC address as the key via "Insert Key"

If the key is accepted the following appears in the Key field: *valid pn-controller key*.



Protocol AMS NetId: 172.16.3.220.2.1
Protocol AMS PortNr.: 65535
Server AMS NetId: 172.16.3.220.1.1
Server AMS PortNr.: 802
Key: valid pn-controller key

Fig. 16: "PROFINET" tab, confirmation of the key

2.4 EL6631/EL6632

2.4.1 EL6632 PROFINET-IRT-Controller

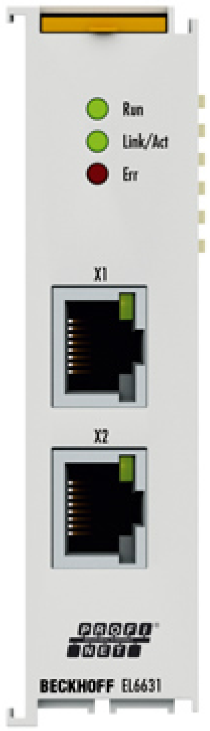


Fig. 17: EL6631

The EL6631 PROFINET IO Controller (Master) terminal supports the complete real-time function (RT) as well as extensive diagnostic possibilities. All services in accordance with Conformance Class B are supported.
Up to 15 PROFINET IO devices can be projected on the EL6631.

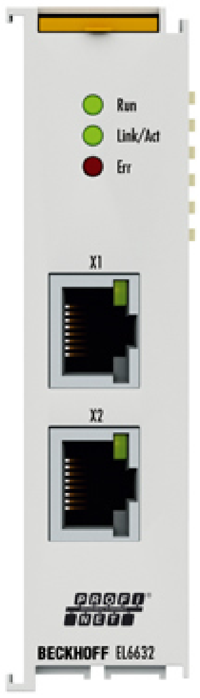


Fig. 18: EL6632

The EL6632 PROFINET-IRT controller terminal offers the complete RT (real time) or IRT (isochronous real-time) functionality and a wide range of diagnostic options. All services in accordance with Conformance Class C are supported.
Depending on the cycle time, up to five PROFINET-IRT or up to 15 PROFINET RT devices can be operated at the EL6632 in a line topology.

Note	**Supplement** The TwinCAT Supplement is not required for the EL6631 and EL6632.

Note	**TwinCAT Version** The released TwinCAT version is TwinCAT 2.11 R3. It must be ensured that the target system also corresponds to the TwinCAT version. Older TwinCAT versions cannot be used!

2.4.2 Technical data - EL6631 PROFINET RT Controller

Technical data	EL6631
Technology	PROFINET IO
Number of ports/channels	2
Ethernet interface	100BASE-TX Ethernet with 2 x RJ 45
Fieldbus	PROFINET RT controller
Cable length	up to 100 m twisted pair
Hardware diagnostics	Status LEDs
Power supply	via the E-bus
Electrical isolation	500 V (E-Bus/Ethernet)
Max. number of RT-capable devices	15 (depending on the cycle time and number of data)
Minimum RT cycle	1 ms
Conformity class	B
Protocol	RT
Drivers	TwinCAT 2.11 R3
Maximum number of process data	1 kbyte input data and 1 byte output data
Configuration	via the EtherCAT master
Current consumption via E-bus	400 mA typ.
Special features	LLDP, SNMP, conformance class B, max. 15 RT users, min. 1 ms RT cycle
Mounting (housing with traction lever unlocking [▸ 21] / housing with front release [▸ 23])	on 35 mm mounting rail conforms to EN 60715
Weight	approx. 75 g
Operating/storage temperature	0…+55 °C/-25…+85 °C
Relative humidity	95 % no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
EMC immunity/emission	conforms to EN 61000-6-2 / EN 61000-6-4
Installation position	Standard installation position (an operating temperature of 45 □ applies to other installation positions), see also notice [▸ 25]
Protection class	IP20
Approval	CE ATEX [▸ 27] cULus [▸ 29]

2.4.3 Technical data - EL6632 PROFINET-IRT-Controller

Technical data	EL6632
Technology	PROFINET IO
Number of ports/channels	2
Ethernet interface	100BASE-TX Ethernet with 2 x RJ 45
Fieldbus	PROFINET RT / IRT controller
Cable length	up to 100 m twisted pair
Hardware diagnostics	Status LEDs
Power supply	via the E-bus
Electrical isolation	500 V (E-Bus/Ethernet)
Max. number of RT-capable devices	5 (depending on the cycle time and number of data)
Max. number of RT-capable devices	15 (depending on the cycle time and number of data)
Minimum IRT cycle	500 µs
Minimum RT cycle	1 ms
Conformity class	C
Protocol	RT or IRT
Drivers	TwinCAT 2.11 R3
Maximum number of process data	1 kbyte input data and 1 byte output data
Configuration	via the EtherCAT master
Current consumption via E-bus	400 mA typ.
Special features	Conformity class C, max. 5 IRT devices in series, max. 15 RT devices, min. 500 µs IRT cycle, min. 1 ms RT cycle
Dimensions (W x H x D)	approx. 26 mm x 100 mm x 52 mm (width aligned: 23 mm)
Mounting (housing with traction lever unlocking [▸ 21] / housing with front release [▸ 23])	on 35 mm mounting rail conforms to EN 60715
Weight	approx. 75 g
Operating/storage temperature	0…+55 °C/-25…+85 °C
Relative humidity	95 % no condensation
Vibration/shock resistance	conforms to EN 60068-2-6/EN 60068-2-27
Installation position	Standard installation position (an operating temperature of 45 □ applies to other installation positions), see also notice [▸ 25]
Protection class	IP20
Approval	CE ATEX [▸ 27] cULus [▸ 29]

3 EL6631/EL6632 - Mounting and wiring

3.1 Instructions for ESD protection

! **Attention**	**Destruction of the devices by electrostatic discharge possible!** The devices contain components at risk from electrostatic discharge caused by improper handling. ✓ Please ensure you are electrostatically discharged and avoid touching the contacts of the device directly. a) Avoid contact with highly insulating materials (synthetic fibers, plastic film etc.). b) Surroundings (working place, packaging and personnel) should by grounded probably, when handling with the devices. c) Each assembly must be terminated at the right hand end with an EL9011 bus end cap, to ensure the protection class and ESD protection.

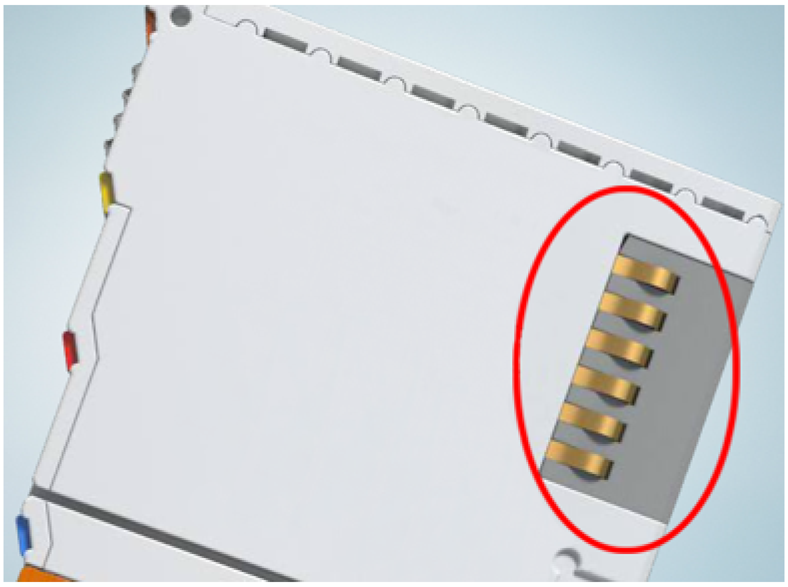


Fig. 19: Spring contacts of the Beckhoff I/O components

3.2 Recommended mounting rails

Terminal Modules und EtherCAT Modules of KMxxxx and EMxxxx series, same as the terminals of the EL66xx and EL67xx series can be snapped onto the following recommended mounting rails:

- DIN Rail TH 35-7.5 with 1 mm material thickness (according to EN 60715)
- DIN Rail TH 35-15 with 1,5 mm material thickness

Note	**Pay attention to the material thickness of the DIN Rail** Terminal Modules und EtherCAT Modules of KMxxxx and EMxxxx series, same as the terminals of the EL66xx and EL67xx series does not fit to the DIN Rail TH 35-15 with 2,2 to 2,5 mm material thickness (according to EN 60715)!

3.3 Mounting and demounting - terminals with traction lever unlocking

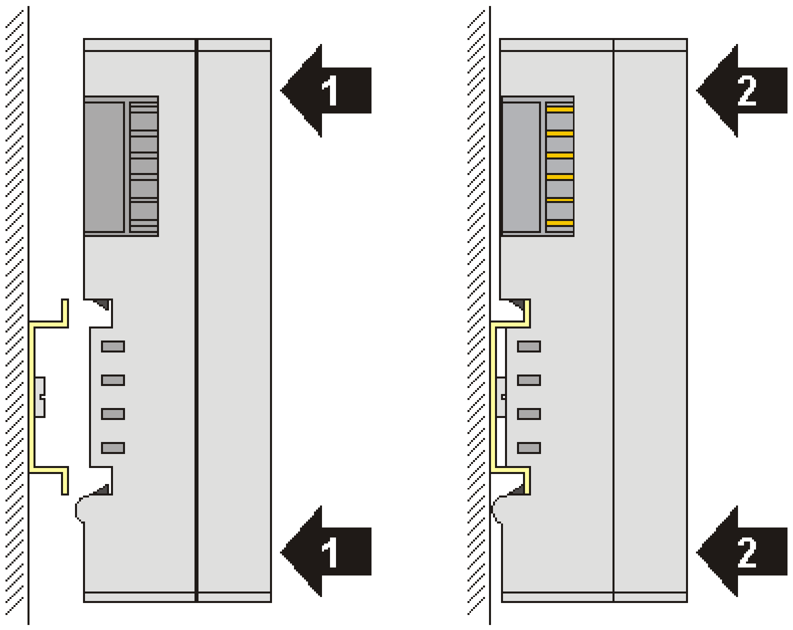
The terminal modules are fastened to the assembly surface with the aid of a 35 mm mounting rail (e.g. mounting rail TH 35-15).

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

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

Mounting

- Fit the mounting rail to the planned assembly location.

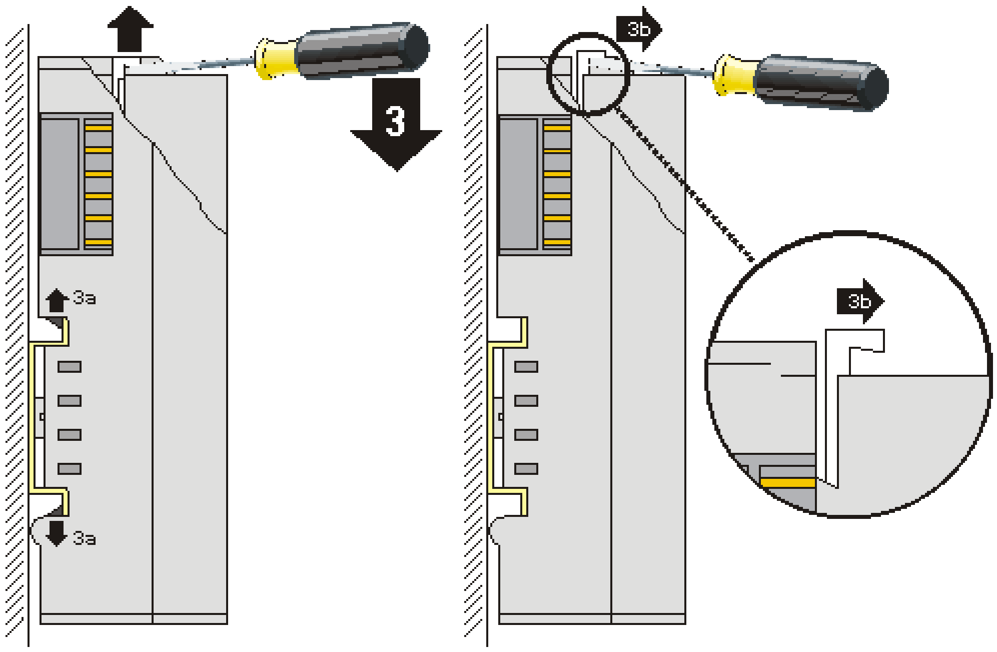


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

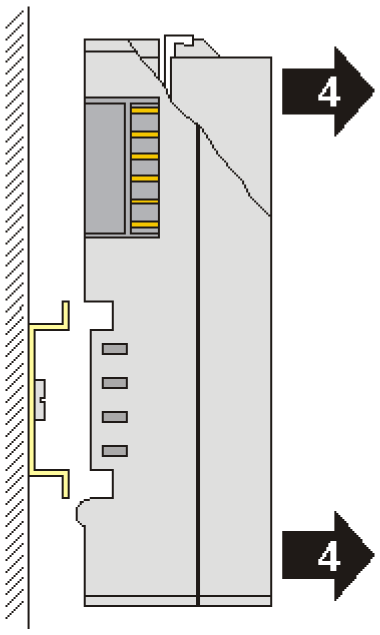
- Attach the cables.

Demounting

- Remove all the cables. Thanks to the KM/EM connector, it is not necessary to remove all the cables separately for this, but for each KM/EM connector simply undo 2 screws so that you can pull them off (fixed wiring)!
- Lever the unlatching hook on the left-hand side of the terminal module upwards with a screwdriver (3). As you do this
    - an internal mechanism pulls the two latching lugs (3a) from the top hat rail back into the terminal module,
    - the unlatching hook moves forwards (3b) and engages



- In the case 32 and 64 channel terminal modules (KMxxx4 and KMxxx8 or EMxxx4 and EMxxx8) you now lever the second unlatching hook on the right-hand side of the terminal module upwards in the same way.
- Pull (4) the terminal module away from the mounting surface.



3.4 Mounting and demounting - terminals with front unlocking

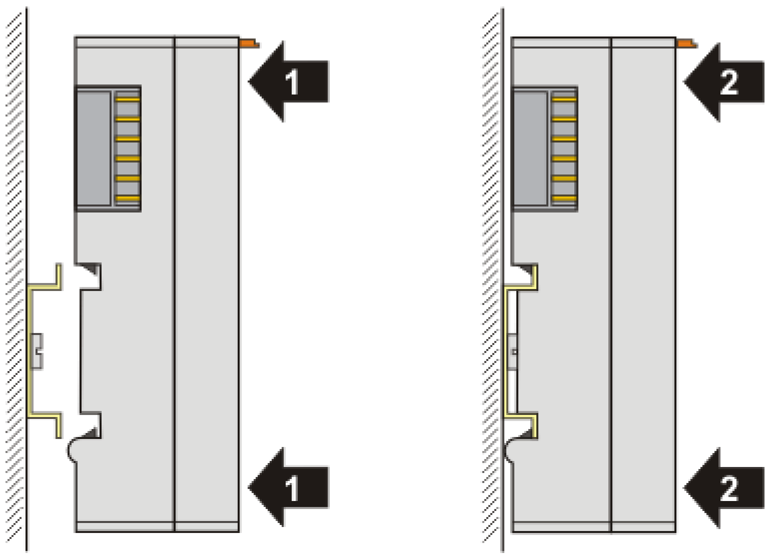
The terminal modules are fastened to the assembly surface with the aid of a 35 mm mounting rail (e.g. mounting rail TH 35-15).

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

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

Mounting

- Fit the mounting rail to the planned assembly location.

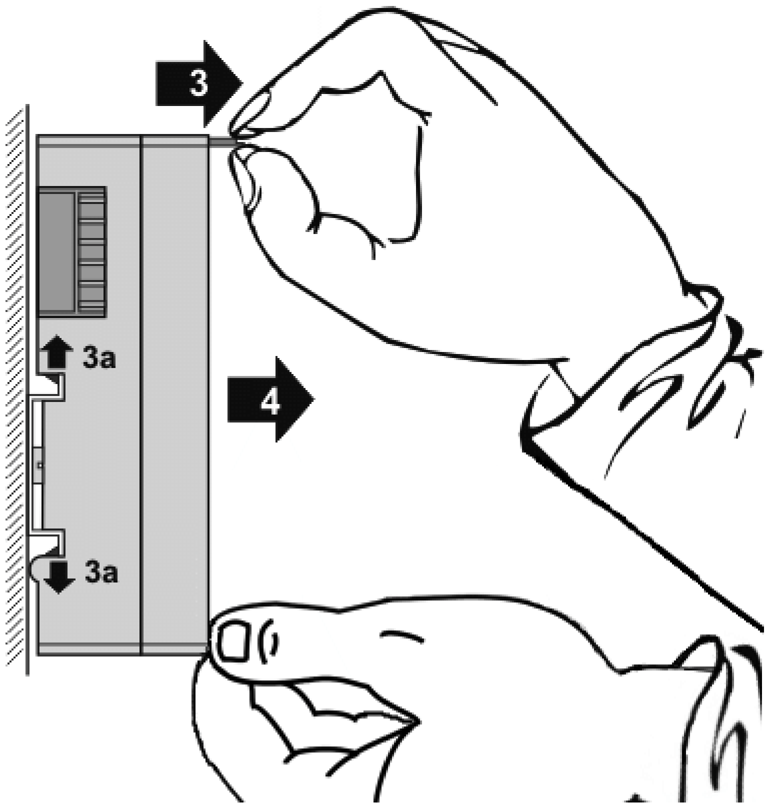


and press (1) the terminal module against the mounting rail until it latches in place on the mounting rail (2).

- Attach the cables.

Demounting

- Remove all the cables.
- Lever the unlatching hook back with thumb and forefinger (3). An internal mechanism pulls the two latching lugs (3a) from the top hat rail back into the terminal module.



- Pull (4) the terminal module away from the mounting surface.
  Avoid canting of the module; you should stabilize the module with the other hand, if required.

3.5 Installation positions

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

Optimum installation position (standard)

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

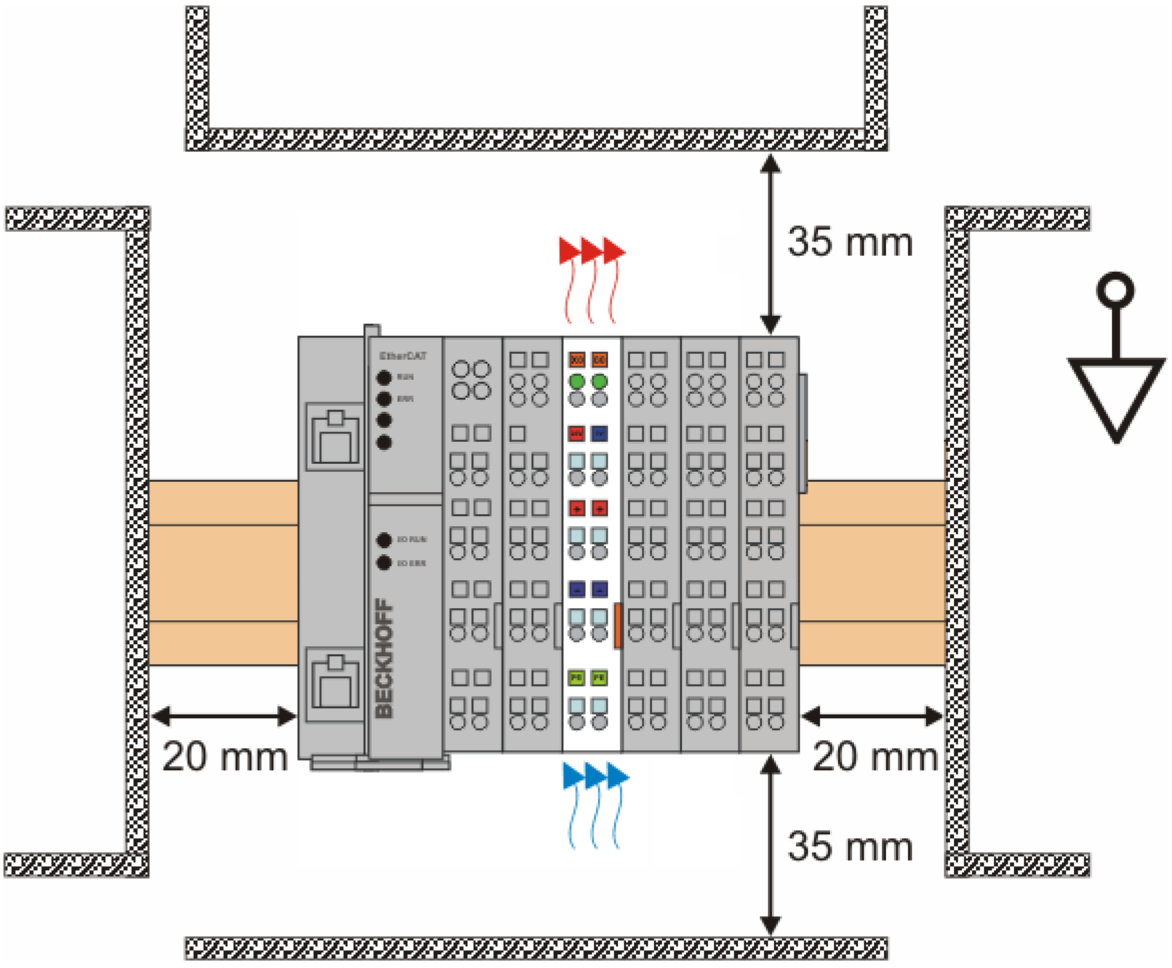


Fig. 20: *Recommended distances for standard installation position*

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

Other installation positions

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

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

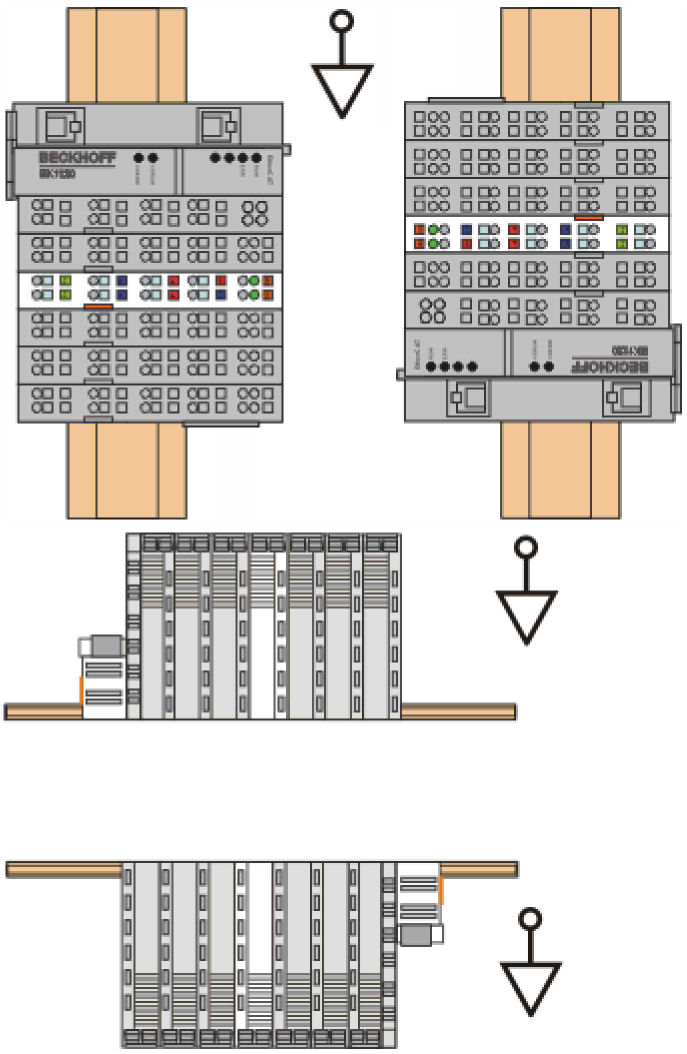


Fig. 21: *Other installation positions*

3.6 ATEX - Special conditions (standard temperature range)

WARNING

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

- The certified components are to be installed in a suitable housing that guarantees a protection class of at least IP54 in accordance with EN 60529! The environmental conditions during use are thereby to be taken into account!
- If the temperatures during rated operation are higher than 70°C at the feed-in points of cables, lines or pipes, or higher than 80°C at the wire branching points, then cables must be selected whose temperature data correspond to the actual measured temperature values!
- Observe the permissible ambient temperature range of 0 to 55°C for the use of Beckhoff fieldbus components standard temperature range in potentially explosive areas!
- Measures must be taken to protect against the rated operating voltage being exceeded by more than 40% due to short-term interference voltages!
- The individual terminals may only be unplugged or removed from the Bus Terminal system if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The connections of the certified components may only be connected or disconnected if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- The fuses of the KL92xx/EL92xx power feed terminals may only be exchanged if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!
- Address selectors and ID switches may only be adjusted if the supply voltage has been switched off or if a non-explosive atmosphere is ensured!

Standards

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

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

Marking

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

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

or

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

3.7 ATEX Documentation

Note	**Notes about operation of the Beckhoff terminal systems in potentially explosive areas (ATEX)** Pay also attention to the continuative documentation Notes about operation of the Beckhoff terminal systems in potentially explosive areas (ATEX) that is available in the download area of the Beckhoff homepage http:\\www.beckhoff.com!

3.8 UL notice

Application

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

Examination

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

For devices with Ethernet connectors

Not for connection to telecommunication circuits.

Basic principles

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

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

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

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

US LISTED
Ind. Cont. Eq.: 24TB

Use 4 Amp. fuse or
Class 2 power supply.
See instructions.

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

Application

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

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

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

3.9 EL6631 - LEDs

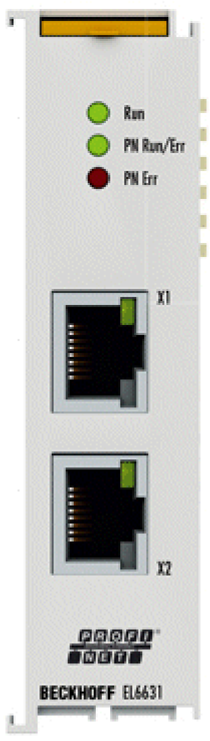


Fig. 22: LEDs EL6631

LEDs for EtherCAT diagnosis

LED		Display	Description
RUN	green	off	State of the EtherCAT State Machine: **INIT** = initialization of the terminal; **BOOTSTRAP** = function for firmware updates of the terminal
		flashing 200 ms	State of the EtherCAT State Machine: **PREOP** = function for mailbox communication and different standard-settings set
		off (1 s) on (200 ms)	State of the EtherCAT State Machine: **SAFEOP** = verification of the sync manager channels and the distributed clocks. Outputs remain in safe state
		on	State of the EtherCAT State Machine: **OP** = normal operating state; mailbox and process data communication is possible

LED diagnosis PROFINET RUN/Err

Color green	Color red	Meaning
off	flashing 200 ms	Terminal starts
flashing 200 ms	off	No name
1 s off, 200 ms on	off	No IP address
on	off	EL terminal is parameterized

LED diagnosis PROFINET Err

Colours green	Colours red	Meaning
off	flashing 500 ms	no AR established with any device
off	1 s off, 200 ms on	At least one device has not established an AR
1 s off, 200 ms on	off	At least one device has signaled an error, e.g. that there is a module difference or that the error bit for an IO-CR is set (Problem Indicator)
flashing 200 ms	off	At least one device is signaling that its status is ‘Stop’ (Provider State Stop)
on	off	All PROFINET devices are in data exchange mode

If there are several different errors, then the error that is located at the top of (or higher in) the table is always displayed.

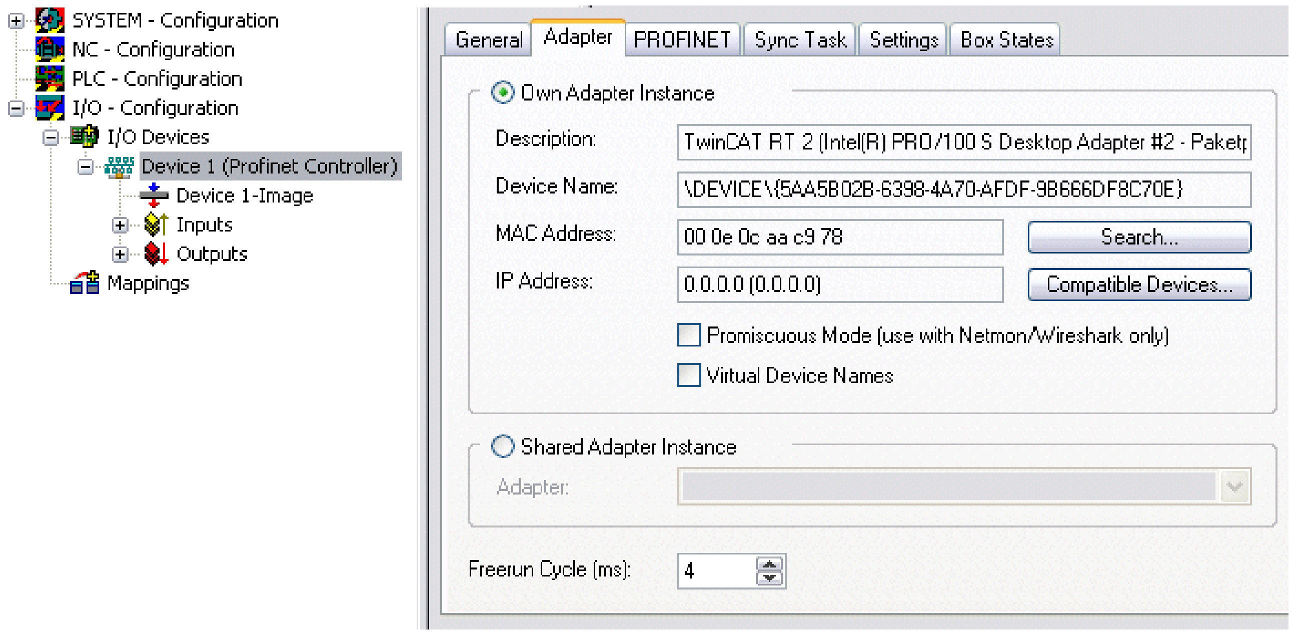
LEDs starting up

Run	PN Run/Err	PN Err	Meaning
off	off	off	No electrical voltage connected to E-bus. The EL6631 must be exchanged if EtherCAT terminals behind it function.
off	off	red on	EL terminal is starting up; after approx. 10 seconds, the LED should go out. If this does not happen, the EL6631 module must be exchanged.

4 PN controller protocol

4.1 Integration of the TwinCAT PROFINET controller protocol via a RealTimeEthernet interface

The controller protocol is appended directly to the I/O device. The available network interfaces are directly displayed when appending and are now available for selection.
If these are to be modified or checked afterwards, this can take place on the ‘Adapter’ tab.



(●) Own Adapter Instance

Description: TwinCAT RT 2 (Intel(R) PRO/100 S Desktop Adapter #2 - Paketp

Device Name: \DEVICE\{5AA5B02B-6398-4A70-AFDF-9B666DF8C70E}

MAC Address: 00 0e 0c aa c9 78

IP Address: 0.0.0.0 (0.0.0.0)

☐ Promiscuous Mode (use with Netmon/Wireshark only)

☐ Virtual Device Names

( ) Shared Adapter Instance

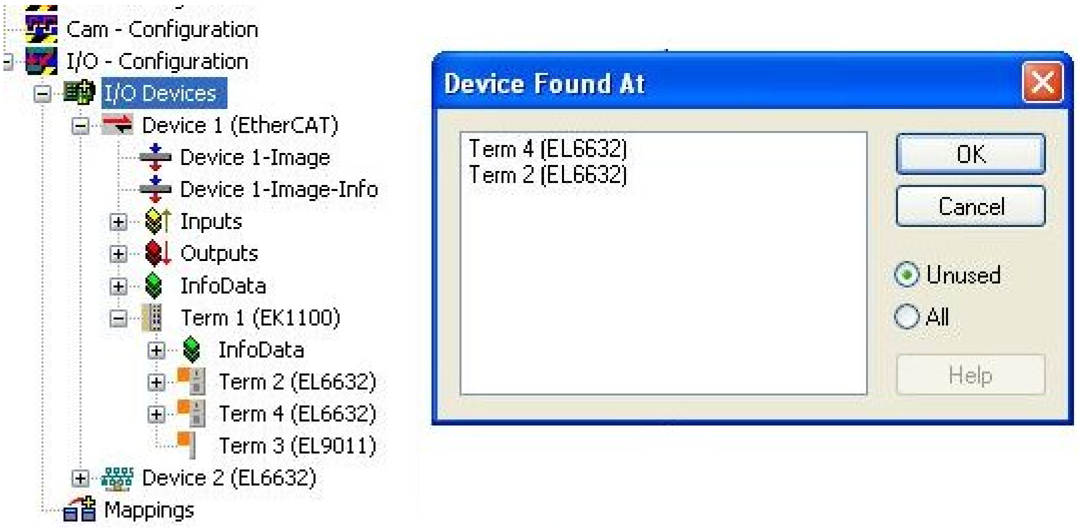
Adapter:

Freerun Cycle (ms): 4

Fig. 23: "Adapter" tab

4.2 Integration of the TwinCAT PROFINET controller protocol via an EL663x interface

The controller protocol is appended directly to the I/O device. The respective protocol must be selected according to the terminal that is to be used (EL6631 or EL6632). If such a terminal is present in the projected EtherCAT strand, the associated adaptor is directly displayed when appending the protocol. If there are several terminals the corresponding one can be selected.

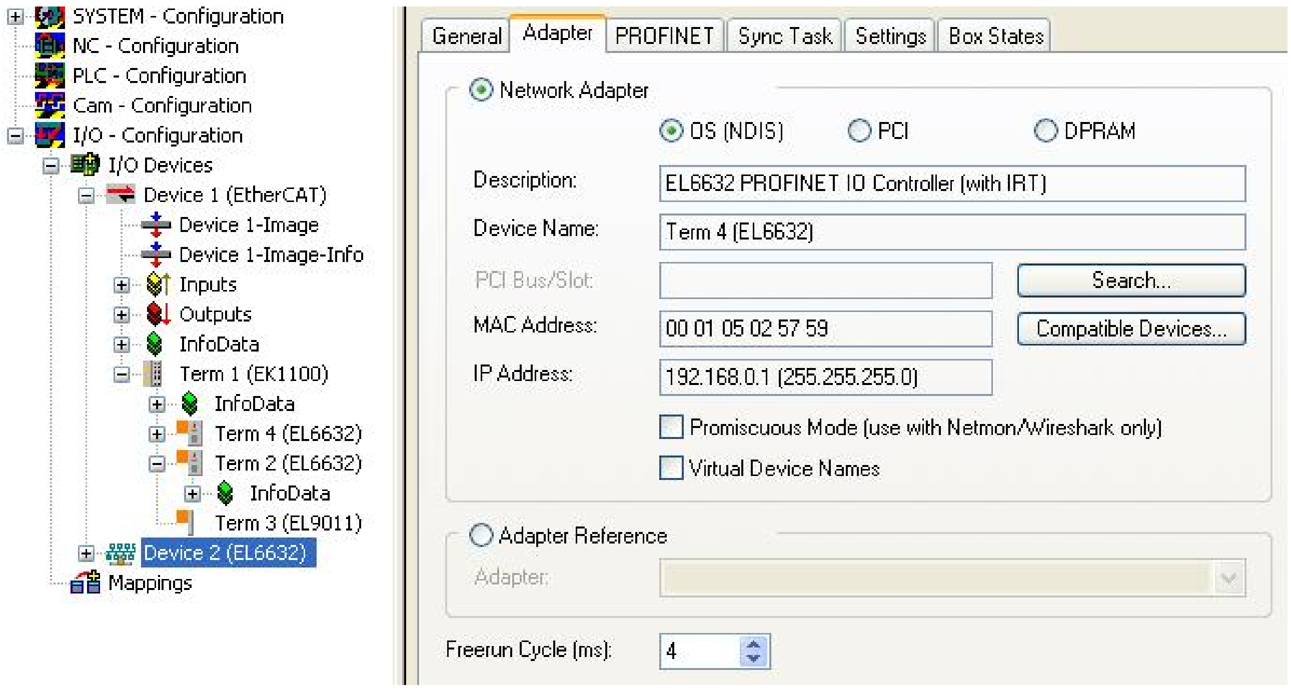


Term 4 (EL6632)
Term 2 (EL6632)

(●) Unused
( ) All

Fig. 24: Selection of the terminal for integrating the protocol

For the operation of several EL663x terminals the corresponding PROFINET protocol must be appended several times. If the terminal assignment is to be modified or checked afterwards, this can take place on the "Adapter" tab.



[✓ Network Adapter]

[✓ OS (NDIS)] PCI DPRAM

Description: EL6632 PROFINET IO Controller (with IRT)

Device Name: Term 4 (EL6632)

PCI Bus/Slot:

MAC Address: 00 01 05 02 57 59

IP Address: 192.168.0.1 (255.255.255.0)

☐ Promiscuous Mode (use with Netmon/Wireshark only)

☐ Virtual Device Names

Adapter Reference

Adapter:

Freerun Cycle (ms): 4

Fig. 25: "Adapter" tab, changing the terminal assignment

4.3 Settings / diagnostics

4.3.1 PROFINET

4.3.1.1 AMS Settings

Protocol AMS NetID text field

This is the NetID via which the PROFINET controller protocol can be reached via AMS.

Protocol AMS PortNo text field

This is the PortNo via which the PROFINET controller protocol can be reached via AMS. This is always fixed to 0xFFFF.

Server AMS NetID text field

This is the NetID to which certain AMS messages (e.g. PN records within the index range 0x1000 - 0x1FFF) are forwarded by the PROFINET driver. Currently this is always the SystemNetId.

Server AMS PortNo text field

This is the PortNo to which certain AMS messages (e.g. PN records within the index range 0x1000 - 0x1FFF) are forwarded by the PROFINET driver. By default this is the PLC Port 802 of runtime system 1.

4.3.1.2 PROFINET Key

Key text field

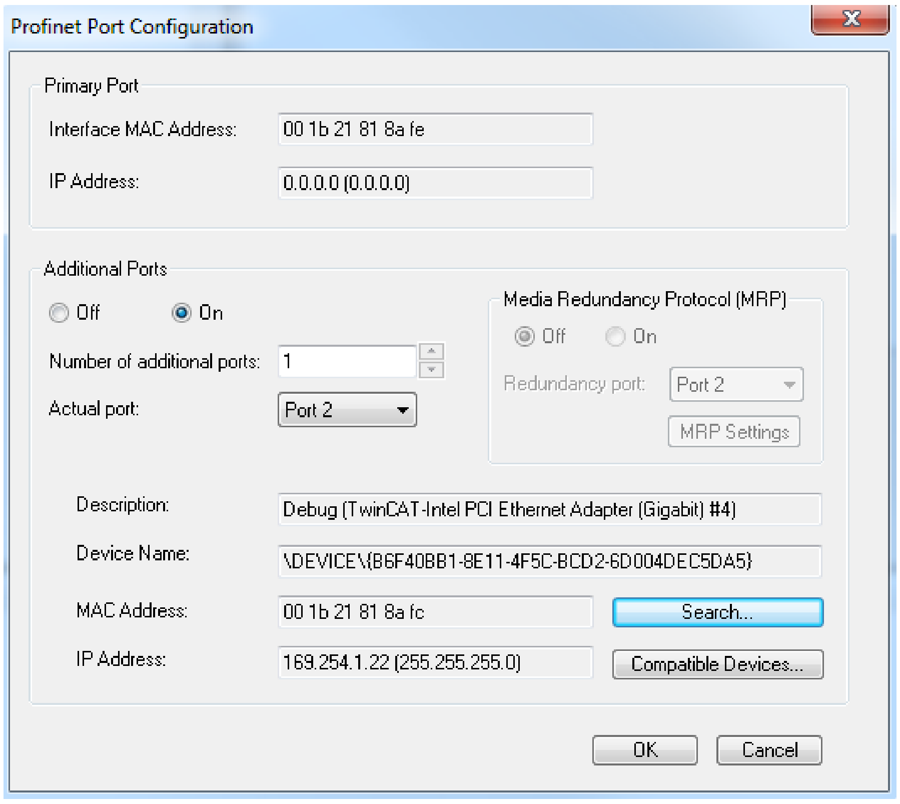
The key to enable the PROFINET controller protocol can be entered here. If a valid key has already been entered, a corresponding message appears in the window and no further entry can be made.

Button Insert Key...

The entered key is confirmed here. The key is thereby checked to ascertain whether or not it is valid. If not, a corresponding message appears. This button is inactive if a valid key has already been entered.

4.3.1.3 Button Port settings

This feature is presently only enabled for the real-time Ethernet protocol (no EL663x). With this a second PROFINET port and an intelligent switch can thus be realized with a second network card (Intel chipset) . It is intended to repeat this feature x times; however, it is presently limited to one additional port.



Profinet Port Configuration

Primary Port

Interface MAC Address:	00 1b 21 81 8a fe
IP Address:	0.0.0.0 (0.0.0.0)

Additional Ports

Off [✓ On]

Number of additional ports:	1
Actual port:	Port 2

Media Redundancy Protocol (MRP)

[✓ Off] On

Redundancy port: Port 2

Description:	Debug (TwinCAT-Intel PCI Ethernet Adapter (Gigabit) #4)
Device Name:	\DEVICE\{B6F40BB1-8E11-4F5C-BCD2-6D004DEC5DA5}
MAC Address:	00 1b 21 81 8a fc
IP Address:	169.254.1.22 (255.255.255.0)

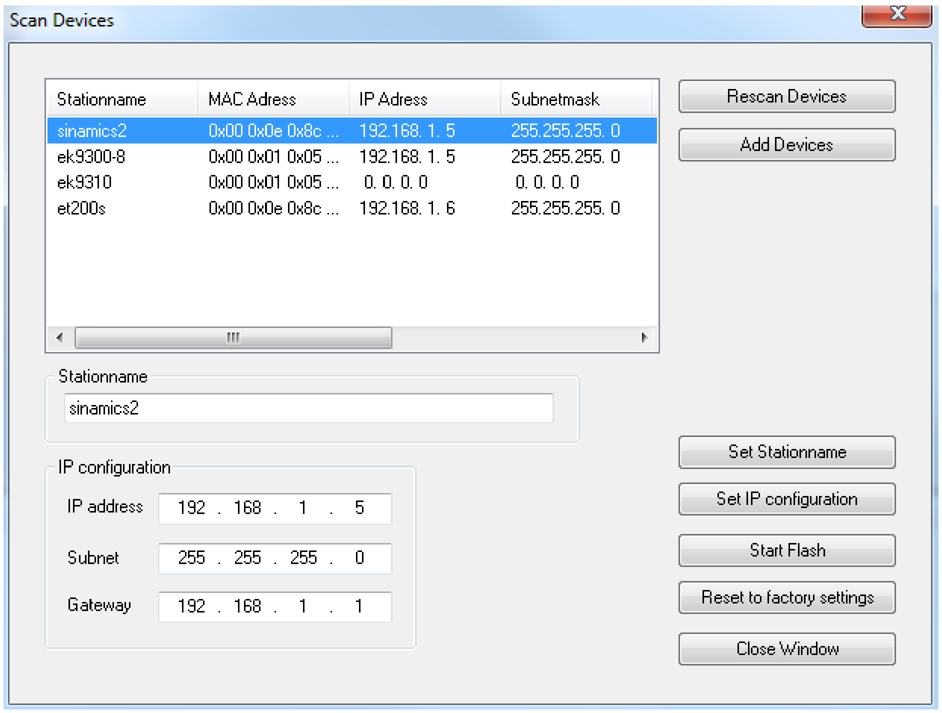
Fig. 26: "Profinet Port configuration" dialog

In future it will be possible to enable the MRP (Media Redundancy Protocol) function via this menu; various settings can be made for this.

4.3.1.4 Button Scan PNIO Devices

This feature is comparable with the ‘ScanBoxes’ feature which, however, is available only in CONFIG mode .

After successful scanning the following dialogue opens (if devices were found).



Stationname	MAC Adress	IP Adress	Subnetmask
sinamics2	0x00 0x0e 0x8c ...	192.168. 1. 5	255.255.255. 0
ek9300-8	0x00 0x01 0x05 ...	192.168. 1. 5	255.255.255. 0
ek9310	0x00 0x01 0x05 ...	0. 0. 0. 0	0. 0. 0. 0
et200s	0x00 0x0e 0x8c ...	192.168. 1. 6	255.255.255. 0

Stationname: sinamics2

IP configuration
- IP address: 192 . 168 . 1 . 5
- Subnet: 255 . 255 . 255 . 0
- Gateway: 192 . 168 . 1 . 1

Fig. 27: Dialog „Scan Devices“

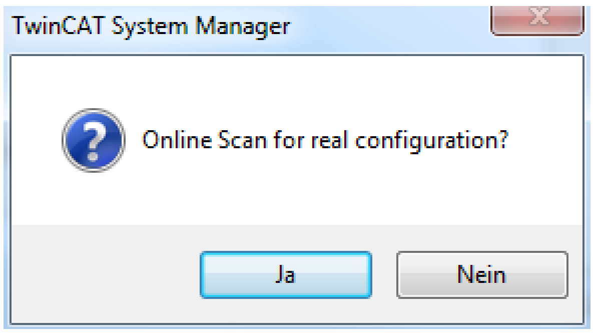
Various settings or project planning can be carried out for the devices here. These are adopted only when the corresponding button is explicitly pressed. When setting the name, care must be taken that only PROFINET-compliant characters are used. This also applies to the IP address; only valid combinations of IP and subnet are to be used. Name and IP are checked for correctness when setting PROFINET devices. DCP_SET is acknowledged with an error If this is not the case. Changes that have been made can be read back by pressing the Rescan button.

In addition the selected device can be signaled. This functionality is PROFINET-specific, but the method by which the signaling takes place is vendor-specific. As standard, however, the signal must arrive with a frequency of 2 Hz.
For example, the Beckhoff BK9103 Bus Coupler signals itself by the alternate flashing of two LEDs at a rate of 2 Hz. This function is very helpful for identifying the devices in this list. The flashing is stopped again by pressing the button once more. The flashing is stopped by closing the ‘Scan Devices’ window.

Subsequently, one or more devices can be marked with the Ctrl button. The selected device is adopted into the project by pressing ‘Add Devices’.

Note	**GSDML devices** The associated GSDML devices must be located in the ‘..\TwinCAT\Io\ProfiNet’ folder!

Upon pressing ‘Add Devices’, the following question box opens:



TwinCAT System Manager

Online Scan for real configuration?

Fig. 28: Confirmation "Add Devices"

'Yes' button:

An attempt is initially made to determine the ModuleIdentNumber of the DAP (Device Access Point) by an implicit read access. If this fails a corresponding dialog opens containing the possible DAPs, which must then be selected manually.
If all boxes have been appended, a 'Reload Devices' automatically takes place, i.e. the devices (adaptors) created are transmitted to the PROFINET driver. Subsequently, a distinction is made as to whether the box is a normal device or a drive with Profidrive support.
If the device is a normal device, the real module equipment (RealIdentificationData) is read out again by an implicit read access. If it is a Profidrive device, conversely, the required information is read out by a Profidrive access. A supervisor AR is established for this, within which the required write accesses can take place. The Submodule interface on the DAP is taken here as the Parameter Access Point. The parameter access takes place via data record 47, much like the case of the Profibus beforehand. When using Sinamics, however, it must be noted that such an access is only permitted from version 4.3 SP2. If an older version is used, a corresponding error message appears and the paramétrisation must take place manually.

Once the automatic module paramétrisation has been completed, a question box appears asking whether the port data should be read in automatically. Here again, the port connection for the individual devices is read out via an implicit read access.
The real port connection must be known for the various services. These may be simply diagnostic services, but the port connection is also required for the automatic device start (via alias) or for the creation of the IRT planning.
If this dialogue is acknowledged with 'no' or if the read access has failed, such a connection can also be made manually at the individual ports in the TwinCAT project.

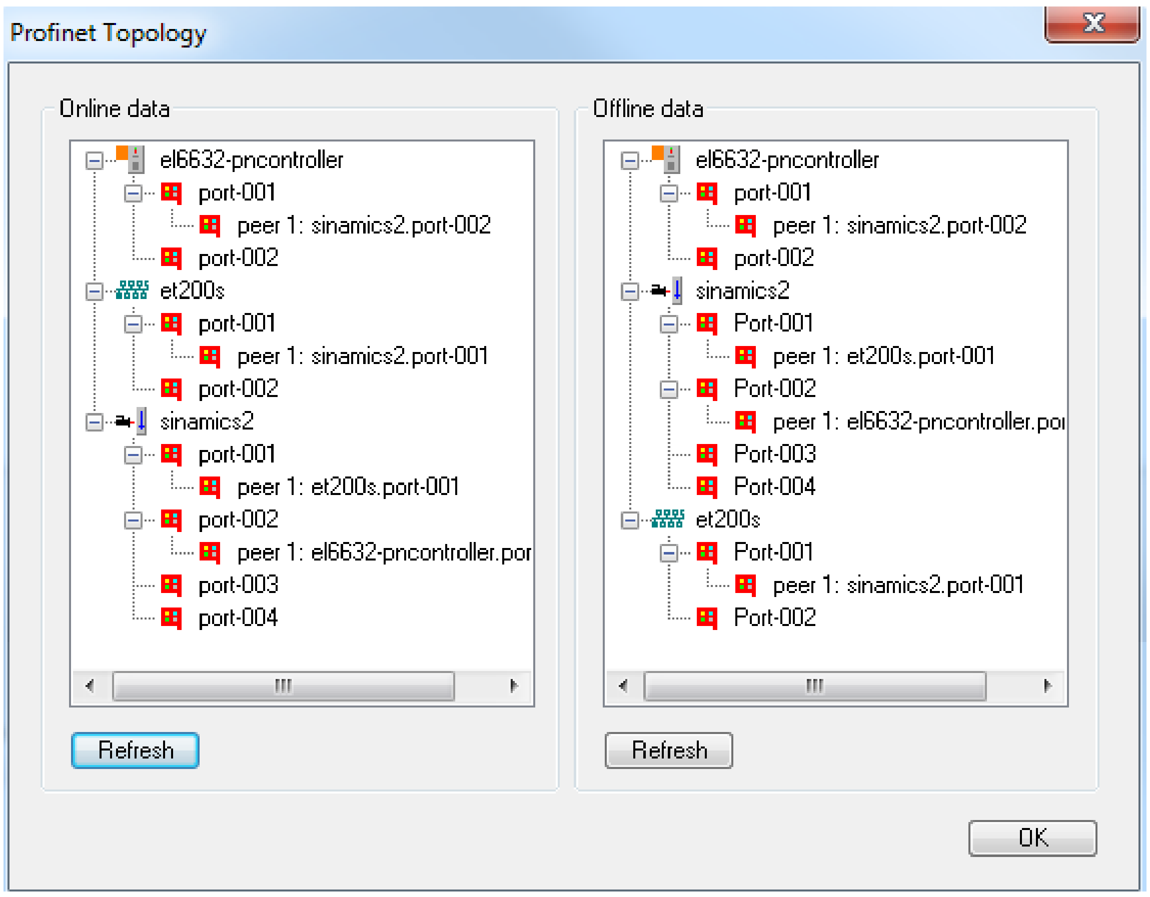
If the port connection has been successfully generated, a query also appears in the case of an IRT controller (e.g. project planning on an EL6632) asking whether all devices are to be automatically connected in IRT Mode (RTClass3) (provided they support it).
If this is affirmed the cable length is additionally set to 10 m copper cable on all projected ports. The IRT algorithm requires this information for the calculation of the signal propagation delays. The precise cable length is not so important here (approx. +/-10 m), because the propagation delays tend to be small at 100 Mbit/s (5 ns/m). If the automatic switchover is not to take place immediately, these points can also be modified later either on the protocol or on the individual devices (on the interface or port submodule).

'No' button:

For each device a check is performed to ascertain whether the GSDML is present in the respective folder ('..\TwinCAT\Io\ProfiNet'). If this is the case, the list of possible DAPs is read in. Subsequently a selection dialogue is opened so that the corresponding DAP can be selected.
Once the devices have been appended in the project, the API below the Box can be accessed and the modules and submodules can be manually appended via it.

4.3.1.5 Button Topology

The offline topology can be compared with the online topology via this dialogue.


Profinet Topology

Online data
- el6632-pncontroller
  - port-001
    - peer 1: sinamics2.port-002
  - port-002
- et200s
  - port-001
    - peer 1: sinamics2.port-001
  - port-002
- sinamics2
  - port-001
    - peer 1: et200s.port-001
  - port-002
    - peer 1: el6632-pncontroller.por
  - port-003
  - port-004

Offline data
- el6632-pncontroller
  - port-001
    - peer 1: sinamics2.port-002
  - port-002
- sinamics2
  - Port-001
    - peer 1: et200s.port-001
  - Port-002
    - peer 1: el6632-pncontroller.poı
  - Port-003
  - Port-004
- et200s
  - Port-001
    - peer 1: sinamics2.port-001
  - Port-002

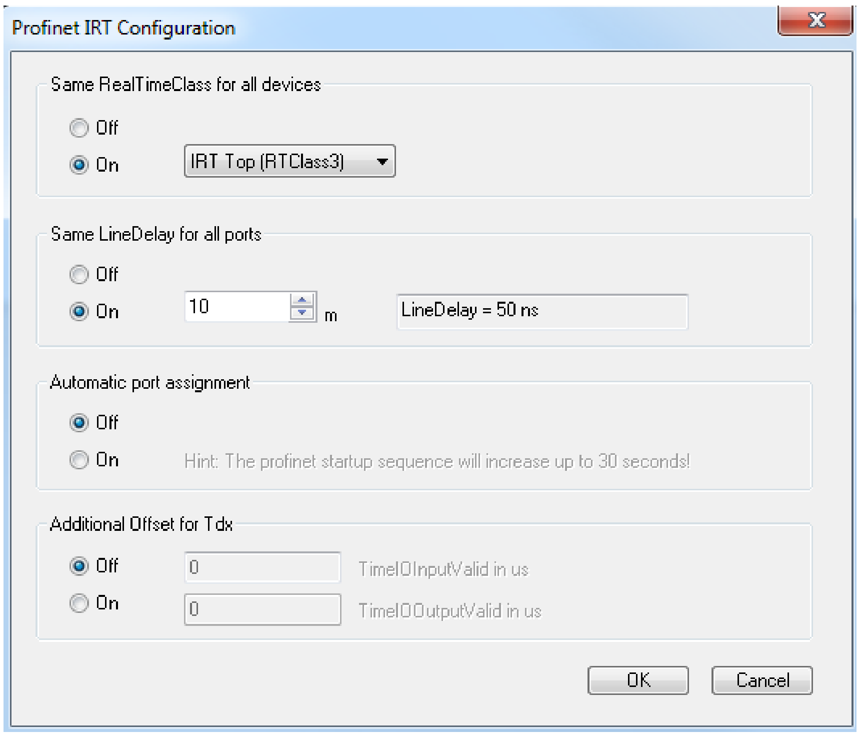
Fig. 29: "Profinet Topology" dialog

It is quite possible for a device to have several partners on a port in the online view. This is the case, for example, if a switch is used in PROFINET that does not support LLDP (protocol for neighborhood ID).

In the offline view, on the other hand, partners may have been assigned that don’t exist in the project. This takes place if the reading of the port properties was activated during automatic scanning and appending. In this case the device has a ‘neighbor’ that is adopted into the project, but the associated device box is missing from the *.tsm file. On activating this project the ‘neighbor’ which does not exist in the .tsm file is ignored in the driver.

4.3.1.6 Button IRT Config

This menu is enabled only for an IRT-capable controller (at present only EL6632). A global setting can be made for all projected devices via this menu.


Profinet IRT Configuration

Same RealTimeClass for all devices
Off [✓ On] IRT Top (RTClass3)

Same LineDelay for all ports
Off [✓ On] 10 m LineDelay = 50 ns

Automatic port assignment
[✓ Off] On
Hint: The profinet startup sequence will increase up to 30 seconds!

Additional Offset for Tdx
[✓ Off] On
0 TimeIOInputValid in us
0 TimeIOOutputValid in us

Fig. 30: "Profinet Port configuration" dialog

On the one hand the type of communication can be specified here. At present only RTClass1 (RT) and RTClass3 (IRT Top) are supported.

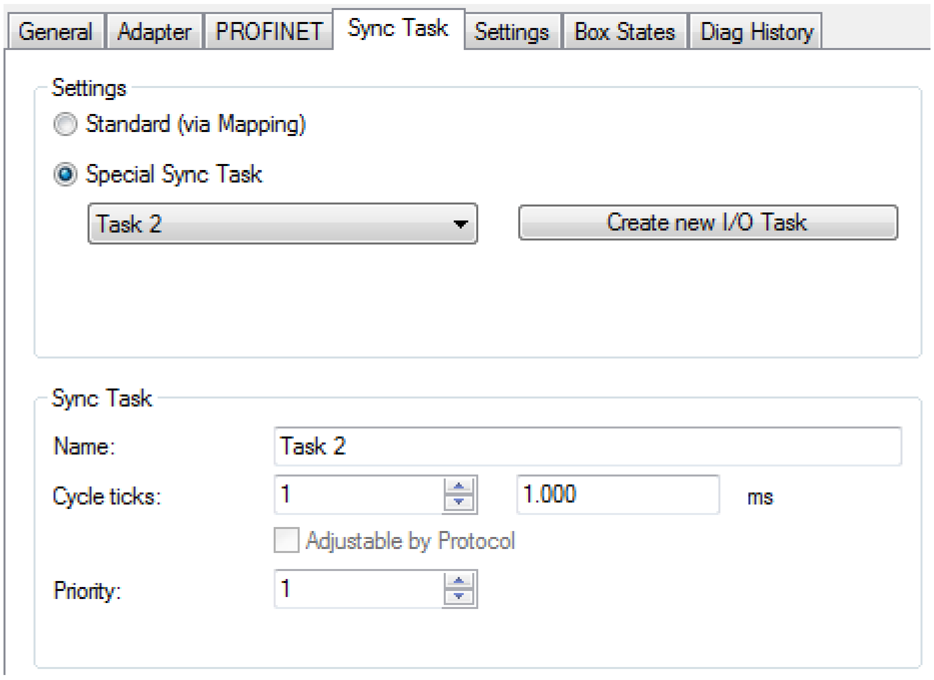
In addition there is an option in this dialogue to specify a general cable length (for IRT only). An approximate value or the max. cable length is sufficient here, because for the calculation of IRT communication this value tends to be lower (at 100 Mbaud and copper cable 5 ns/m). For optimization this feature can also be deactivated again later and the exact cable length can then be entered for each individual device (on the port submodules).

Furthermore there is an option here to activate an ‘automatic port assignment’. As a result of this the port connection set in the TwinCAT project is irrelevant. Before each restart of the PN communication the topology is read out and the IRT communication is calculated on the basis of this. The advantage of this is that possible cabling errors are minimized. In addition to that the ports can be simply replugged without having to change and reload the TwinCAT project. Only a restart of the PN communication is required (e.g. switch terminal to PREOP or disconnect the cable). As a result of this the bootup of the PROFINET communication can take up to 30 seconds longer. The reason for this is the TTL (TimeToLive) factor in the LLDP MIB. These are set by default to 20 seconds, i.e. only after this time can it be guaranteed that the port connection read is also the current one.

Also, an additional offset for all Ti / To values can be specified in this menu.

4.3.2 Task configuration

The PROFINET controller protocol must always be linked with a task. The protocol is also processed with the set task time. Theoretically the controller can also be jointly processed, for example, via a PLC or NC task. However, if a PLC project, for example, is stopped (e.g. by restart or debugging), this results in the PROFINET part also being stopped. In order to avoid such a side effect is it advisable to always create a free-running SyncTask.



Settings
Standard (via Mapping) [✓ Special Sync Task]
Task 2

Sync Task
Name: Task 2
Cycle ticks: 1 1.000 ms
Adjustable by Protocol
Priority: 1

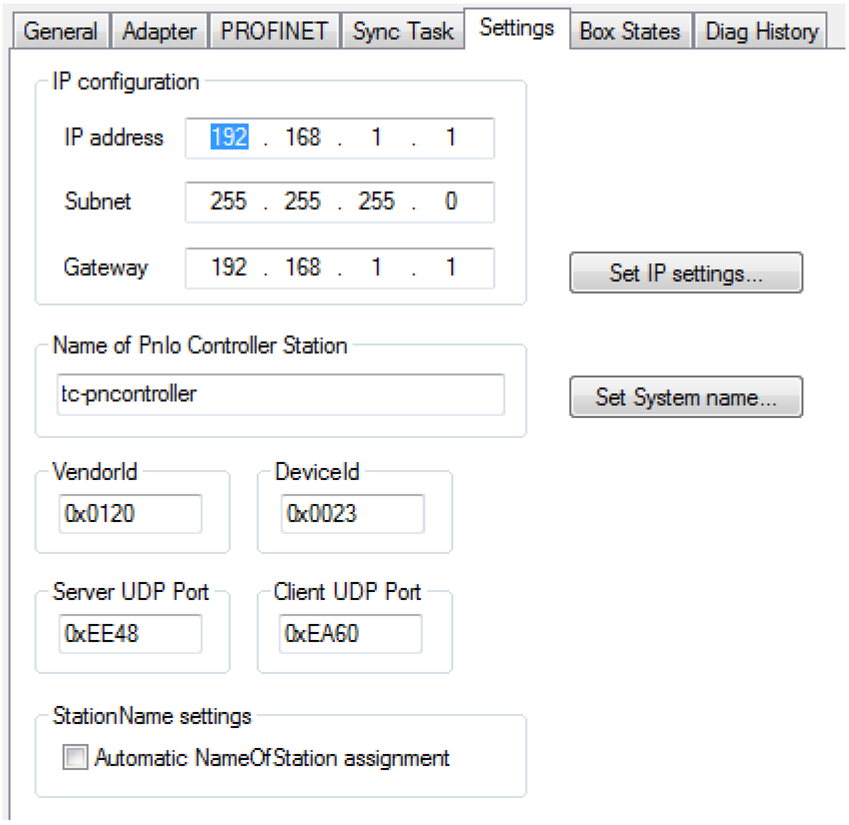
Fig. 31: "Sync Task" tab

It must be ensured that the task cycle lies in a PROFINET cycle, i.e. for PROFINET the basic cycle is 31.25 µs. This cycle is then always multiplied by the SendClockFactor (SCF) to obtain the basic cycle. The SendClockFactor is usually set to 32 for RTClass1. This is also the minimum PN cycle for RTClass1 for the Beckhoff PROFINET controller and results in the smallest cycle time of 1 ms. Further reductions take place using a ReductionRatioFactor. This always corresponds to a multiple of the minimum PN cycle. For RTClass1 the smallest cycle must always be doubled (permissible cycle times (for RTC1) with an SCF of 32 are 1, 2, 4, 8,…, 512).

The SCF can and must be reduced in order to achieve faster cycle times for RTClass3 as well. This is presently at least 16 for a Beckhoff IRT controller (EL6632), which corresponds in turn to a basic cycle of 500 µs. In the case of such a decrease of the PROFINET cycle, it must be noted that the time of the triggering task must also be adapted accordingly.

4.3.3 PROFINET controller-specific settings

Settings that directly concern the controller can made on the ‘Settings’ tab.



IP configuration

IP address 192 . 168 . 1 . 1

Subnet 255 . 255 . 255 . 0

Gateway 192 . 168 . 1 . 1

Name of PnIo Controller Station

tc-pncontroller

VendorId 0x0120

DeviceId 0x0023

Server UDP Port 0xEE48

Client UDP Port 0xEA60

StationName settings

☐ Automatic NameOfStation assignment

Fig. 32: "Settings" tab

An IP setting can take place here. The selection of the address range need not correspond to the network card settings. The PROFINET communication spreads its own net, which can be selected here. The IP settings shown in the above illustration are the default settings, i.e. if nothing is changed the controller uses these settings. The same applies to the controller name (system name). The corresponding button must be pressed to change either of these two settings. A check is made to ascertain that the input is correct (e.g. the format of the controller name must correspond to the PN spec.). These data are then permanently adopted. If the subnet or gateway is changed, the settings will also be adopted by possible projected devices. There is also a possibility to modify these settings via a supervisor tool.

In addition the VendorID and DeviceID of the controller can be read out in this dialogue. The server and client UDP port employed can also be set here. The default settings should, however, be adequate in most cases.

Furthermore there is a possibility in this dialogue to enable an automatic PROFINET start-up following a device exchange (including devices without removable media). For correct functioning the nominal topology must be specified once. On the basis of this information the controller can query the alias names of the individual devices. Every device that supports alias names generates such a name for each of its ports. This is composed of the neighborhood IDs (PortId.ChassisId). If this name is queried, the ‘new’ device answers. If VendorId and DeviceId are correct the device is named with the actual name and a normal PROFINET start-up can subsequently take place. With this mechanism a complete PROFINET system could also start up without having named an individual device beforehand.

4.3.4 Analysis of the box states

Directly below the PROFINET controller protocol there is a collective PROFINET error and a collective PROFINET status. Both indicate the number of devices in which a problem has occurred or in which a diagnosis is available, i.e. the error indicates possible problems with the establishment of a connection or reasons for an abort. The diagnosis provides status information about an existing connection.

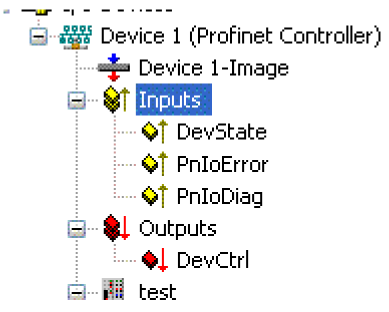
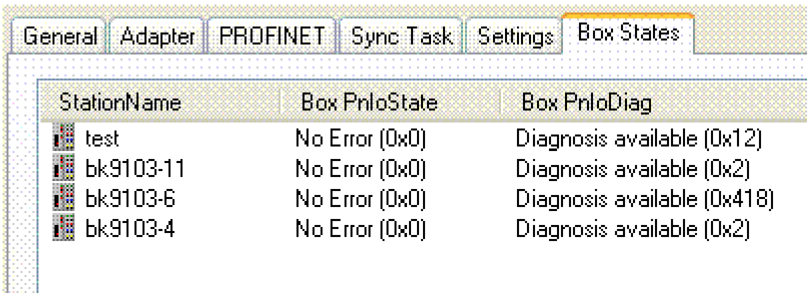


Fig. 33: TwinCAT tree, inputs for analysis

PnIoError - number of PROFINET IO devices that have an error
PnIoDiag - number of PROFINET IO devices where a diagnosis is available

It is possible to check at a glance which device or box has a problem in the protocol under Box States.



StationName	Box PnIoState	Box PnIoDiag
test	No Error (0x0)	Diagnosis available (0x12)
bk9103-11	No Error (0x0)	Diagnosis available (0x2)
bk9103-6	No Error (0x0)	Diagnosis available (0x418)
bk9103-4	No Error (0x0)	Diagnosis available (0x2)

Fig. 34: "Box states" tab

Presently the following error messages are displayed under the “PnIoState”.

Number	Text	Description	Remedial action / reason
0	No error	No error	No error
1	PROFINETDevice state machine is in boot mode	PROFINET Device State Machine is still in the start-up phase	Not an error, wait
2	Device not found	Device does not reply to the Identify Request	Check connection, device connected, was the device called by its correct name?
3	The stationname is not unique	The station name is not unique	There are two or more devices in the network with the same PROFINET name. A correct identification cannot take place.
4	IP could not set	IP address could not be set	The PROFINET device has rejected the IP settings for some reason. Check whether the IP settings are correct.
5	IP conflict	An IP conflict has occurred in the network	A possible cause is that several devices have the same IP address.
6	DCP set was not successful	There was no reply or an erroneous reply to a DCP Set	Check connection, device connected, was the device called by its correct name?
7	Watchdog error	The connection was broken off with a Watchdog error	Check the cycle time, check the connection, if necessary increase the Watchdog factor.
8	Datahold error	The connection was broken off with a Datahold error	Frame Data status was invalid for the length of the DataHoldTimer. Restart the device if necessary.
9	RTC3: Sync signal could not started	For IRT only: the Sync signal could not be started.	Is EtherCAT Sync signal correct or has Sync0 started?
10	PROFINET Controller has a link error	The PROFINET controller has no link	Check cable and connection.
11	The aliasname is not unique	The alias name is not unique	There are two or more devices in the network with the same alias name. This is made up of the neighbourhood information (PortId.ChassisId). A correct identification cannot take place.
12	The automatic name assignment isn't possible - wrong device type	The automatic name assignment is not possible	The expected PROFINET device is not in the projected position (VendorId or DeviceId does not correspond). Hence, no automatic naming and thus no device start is possible.
31	Only for EtherCAT gateways: WC-State of cyclic EtherCAT frame is 1	For EL6631 only: EtherCAT WC State is 1	Check the mode on the EtherCAT master & slave (OP?).

As opposed to the state, more than one status can be displayed in the “BoxPnIoDiag”, i.e. the whole thing is bit-coded and up to 16 pieces of information can be displayed. The following statuses are currently displayed.

0x0000 = No diagnosis
0xXXX1 = IOC-AR is not established
0xXXX2 = IOC-AR is established
0xXXX4 = IOC-AR is established but no ApplReady
0xXXX8 = IOC-AR is established but module difference
0xXX1X = At least one AlarmCR get diagnosis alarm
0xX1XX = At least one InputCR is invalid
0xX2XX = At least one InputCR Provider is in stop
0xX4XX = At least one InputCR Problemindicator is set
0x1XXX = At least one OutputCR is invalid
0x2XXX = At least one OutputCR Provider is in stop
0x4XXX = At least one OutputCR Problemindicator is set

On the one hand information about the status of the IO Controller Single AR is displayed here. In addition, collective statuses are formed from the Frame Data statuses of the individual CRs. This applies to the input and output CRs (currently only one CR is possible, in future several). Furthermore a PROFINET alarm is also displayed in “PnIoDiag”

Readout via ADS

The Box Status can be read out via an ADS Read.

ADS Read:
NetId = AMSNETID des PROFINET Controllers
Port = BoxPort (0x1000 + BoxId)
Indexgroup = 0xF829
IndexOffset = 0
Length = sizeof(TPnIoDeviceDiagData);

where:

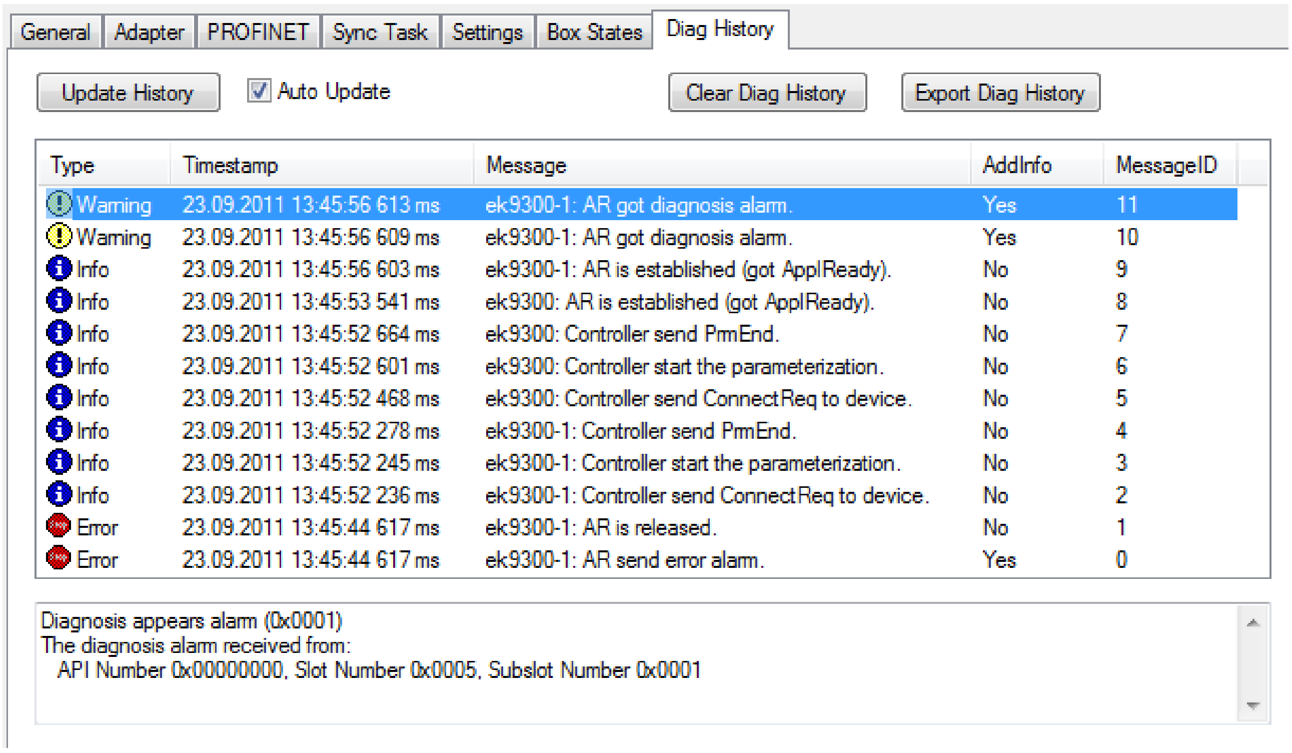
```
typedef struct
{
WORD pnioState;
WORD pnioDiag;
WORD NrOfInputCRs;
WORD NrOfOutputCRs;
WORD reserved[8];
} TPnIoDeviceDiagData, *PTPnIoDeviceDiagData;
```

Readout via CoE (for EL663x)

The Box Status can also be read out via CoE for the EL663x. The index 0xAyy0 (where yy is the Adaptor / Device number) and the subindex 0x001 must be taken for this.

4.3.5 Diagnosis history on the controller protocol

Logged diagnosis messages from the controller protocol can be read out on the “Diag History” tab. The diagnosis buffer operates as a ring buffer with a current maximum size of 1000 entries.



Type	Timestamp	Message	AddInfo	MessageID
Warning	23.09.2011 13:45:56 613 ms	ek9300-1: AR got diagnosis alarm.	Yes	11
Warning	23.09.2011 13:45:56 609 ms	ek9300-1: AR got diagnosis alarm.	Yes	10
Info	23.09.2011 13:45:56 603 ms	ek9300-1: AR is established (got ApplReady).	No	9
Info	23.09.2011 13:45:53 541 ms	ek9300: AR is established (got ApplReady).	No	8
Info	23.09.2011 13:45:52 664 ms	ek9300: Controller send PrmEnd.	No	7
Info	23.09.2011 13:45:52 601 ms	ek9300: Controller start the parameterization.	No	6
Info	23.09.2011 13:45:52 468 ms	ek9300: Controller send ConnectReq to device.	No	5
Info	23.09.2011 13:45:52 278 ms	ek9300-1: Controller send PrmEnd.	No	4
Info	23.09.2011 13:45:52 245 ms	ek9300-1: Controller start the parameterization.	No	3
Info	23.09.2011 13:45:52 236 ms	ek9300-1: Controller send ConnectReq to device.	No	2
Error	23.09.2011 13:45:44 617 ms	ek9300-1: AR is released.	No	1
Error	23.09.2011 13:45:44 617 ms	ek9300-1: AR send error alarm.	Yes	0

```
Diagnosis appears alarm (0x0001)
The diagnosis alarm received from:
  API Number 0x00000000, Slot Number 0x0005, Subslot Number 0x0001
```

Fig. 35: "Diag History" tab

The possible errors are grouped into three types:

- Info: e.g. information on the connection establishment
- Warning: e.g. PROFINET diagnostic alarms
- Error: e.g. Loss of connection

“AddInfo” indicates whether additional information about the event is available. If this is marked by “Yes”, the additional information can be fetched and displayed by clicking on the respective message. In the case of a diagnosis alarm (“Diagnosis” appears), the precise diagnosis information can be fetched at the corresponding level (device, API or module).

The complete diagnosis buffer is cleared by pressing the “Clear Diag History” button.

The displayed messages can be saved in a .TXT file by pressing the “Export Diag History” button.

4.3.6 Cyclic data

There are several cyclic process data directly below the PROFINET controller protocol. These data are only exchanged between the PROFINET driver and the system manager. They provide general information about the status of the PROFINET communication.

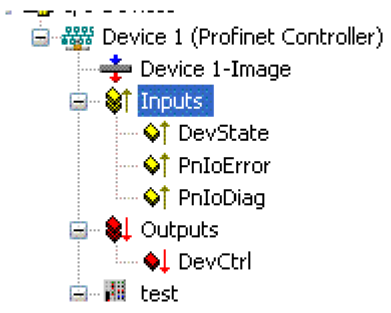


Fig. 36: TwinCAT tree, inputs for info

The ‘DevState’ variable contains information about the physical communication status of the controller, such as the link status or whether the transmission resources are still sufficient.

The other variables are the collective PROFINET error and the collective PROFINET status. Both indicate the number of devices in which a problem has occurred or in which a diagnosis is available, i.e. the error variable indicates possible problems with the establishment of a connection or reasons for an abort. The diagnostic variable provides status information about an existing connection.

The ‘DevCtrl’ output variable currently has no function.

For further information, please also read the chapter entitled ‘Box States [▸ 42]'.

4.3.7 Acyclic data

The ADS blocks are used to send acyclic data. These then access the PROFINET record data. So that acyclic data can be read or written, the PROFINET device must be in data exchange mode.

An *ADSReadWrite* is set.

ADS settings

AMSNetID: The AMSNetID of the Profinet controller

PORT: Port number of the device (take this from the system manager)

Index GROUP: 0x0000_F823

Index OFFSET: 0x0000_0000

DATA

```
typedef struct {
    WORD        RW;
        #define     PN_READ     0
        #define     PN_WRITE    1
    WORD        NrOfAR;
    DWORD        API;
    WORD        Slot;
    WORD        SubSlot;
    PNIO_RECORD    RecordData;
} PNIO_CONFIGRECORD
```

Table 1: Structure of the record data frame

nRW	nNr	nAPI	InSlot	SubSlot	nIndex	nLen	nTrans	nLenA
2 Byte	2 Byte	4 Byte	2 Byte	2 Byte	2 Byte	2 Byte	2 Byte	2 Byte

Table 2: Meaning of data from the record data frame

Designation	Values	Description
nRW	0 – READ 1 - WRITE	read- or write-access
nNr	normally „0“dec	it is possible to put multiple ARs (application releations) to one device (Controller, Supervisor, DeviceAccess). This indicates in which AR the data is exchanged -> normally there is only one AR, in this case zero.
nAPI	normally „0“dec	-> otherwise, the corresponding application profile should be placed here
nSlot	variable	Slot number
nSubSlot	variable	SubSlot number
nIndex	variable	Index number
nLen	variable	READ if nRW = 0: if the value „0“ is used when reading, the request is sent with the maximum buffer size, if nLen ≠ 0, the corresponding length is used. WRITE if nRW = 1: when writing: number of bytes, which follow from or after the “nReserved” word
nTrans	starts with „1“dec	If multiple records are downloaded at once, this transfer sequence number determines the order in which the data is processed.
nReserved	„0“dec	2 byte alignment
Data	variable	Data (from here the „nLen“ for the data length counts (writing only))

Sample:

Send a read request for I&M function 0

nRW	nNr	nAPI	InSlot	SubSlot	nIndex	nLen	nTrans	nLenA
00 00	00 00	00 00 00 00	00 00	01 00	F0 AF	00 00	01 00	00 00

Make sure that the receive data memory is large enough!

5 Devices at protocol

5.1 Appending PROFINET devices

Select “Append Box” by right-clicking on the protocol. The following dialogue then opens:

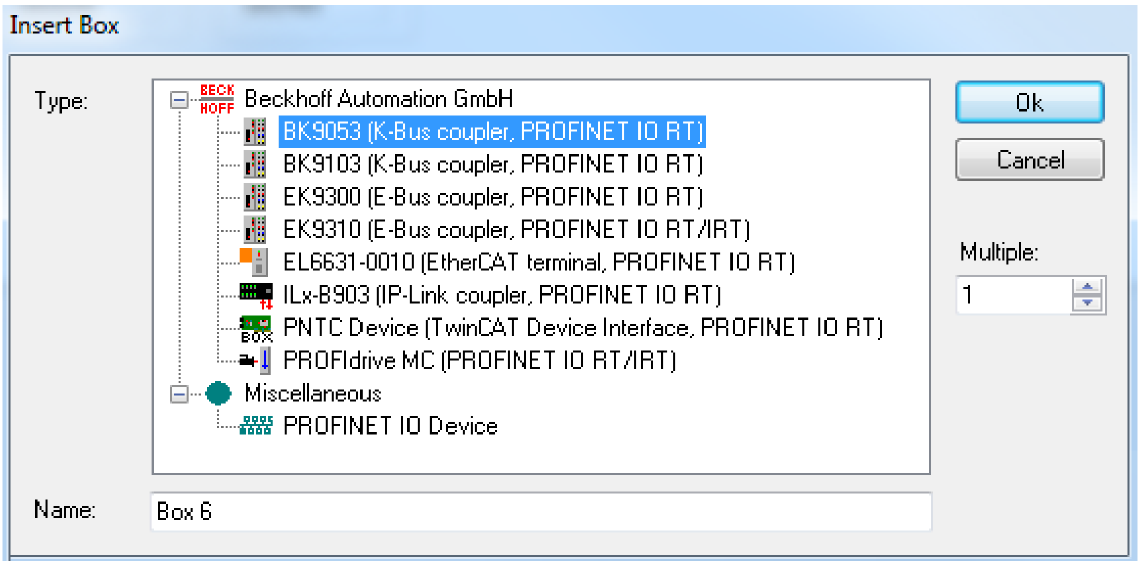

Insert Box
Type:
Beckhoff Automation GmbH
BK9053 (K-Bus coupler, PROFINET IO RT)
BK9103 (K-Bus coupler, PROFINET IO RT)
EK9300 (E-Bus coupler, PROFINET IO RT)
EK9310 (E-Bus coupler, PROFINET IO RT/IRT)
EL6631-0010 (EtherCAT terminal, PROFINET IO RT)
ILx-B903 (IP-Link coupler, PROFINET IO RT)
PNTC Device (TwinCAT Device Interface, PROFINET IO RT)
PROFIdrive MC (PROFINET IO RT/IRT)
Miscellaneous
PROFINET IO Device
Ok
Cancel
Multiple:
1
Name:
Box 6

Fig. 37: "Insert Box" dialog

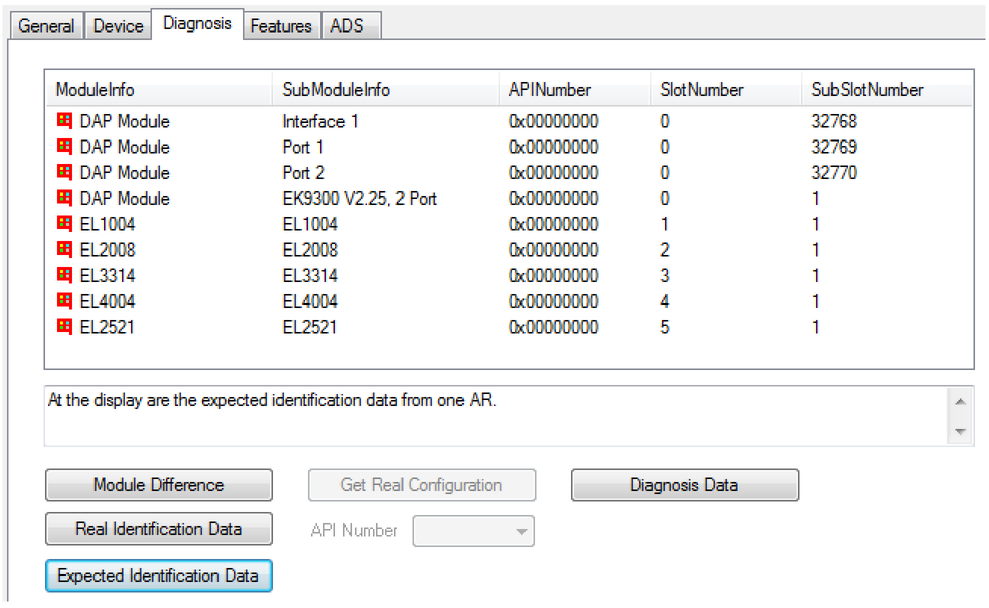
There is a possibility here to select various PROFINET devices. In Beckhoff devices a search is carried out via a defined path for the GSDML (..\TwinCAT\IO\PROFINET). These should be already present with the TwinCAT installation. If there are several GSDMLs for the same device here, the one with the latest date is taken. If no device description is found, a corresponding error message appears. Either the GSDML is copied into the folder and the menu is opened again, or the same procedure is selected as for the third-party devices. If you click on “PROFINET IO Device”, an option is offered to navigate to the corresponding GSDML in Windows Explorer. This is then integrated into the project.

The DNS name from the GSDML is taken as the default name. When appending several devices simultaneously, the default name is always supplemented by “-No.” (where No. = 1 to n). The name that was assigned (and with which the device also appears in the tree), is at the same time also the “PROFINET Station Name”, i.e. the name that must correspond to the name in the device. The device name can be checked by scanning.

The modules can be attached to the API (Application Profile Interface). The DAP (Device Access Point), which already brings along fixed properties from the GSDML (e.g. process data, interface and port submodules, etc.), is always on Slot 0.
This module is always there and cannot be deleted or shifted. Each further module is assigned to a certain API. The information regarding its identity comes from the GSDML. As standard this is always the API 0. Alternatively, an API e.g. for the PROFIDRIVE profile or a fieldbus API is also conceivable. By clicking in the API on “Append PROFINET modules…” a device catalogue is opened from which the corresponding modules can be selected and appended. If the modules support it (described in GSDML), the submodules can in turn be appended to them in the same way.

5.2 Comparison of nominal and actual configuration

If a connection exists, the project planning can be checked on the “Diagnosis” tab. At this level “Real Identification Data” returns the actually existing modules in an AR, “Expected Identification Data” indicates the expected modules (i.e. those projected in the controller) and “Module Difference” shows the differences to the device found in the nominal-actual comparison.

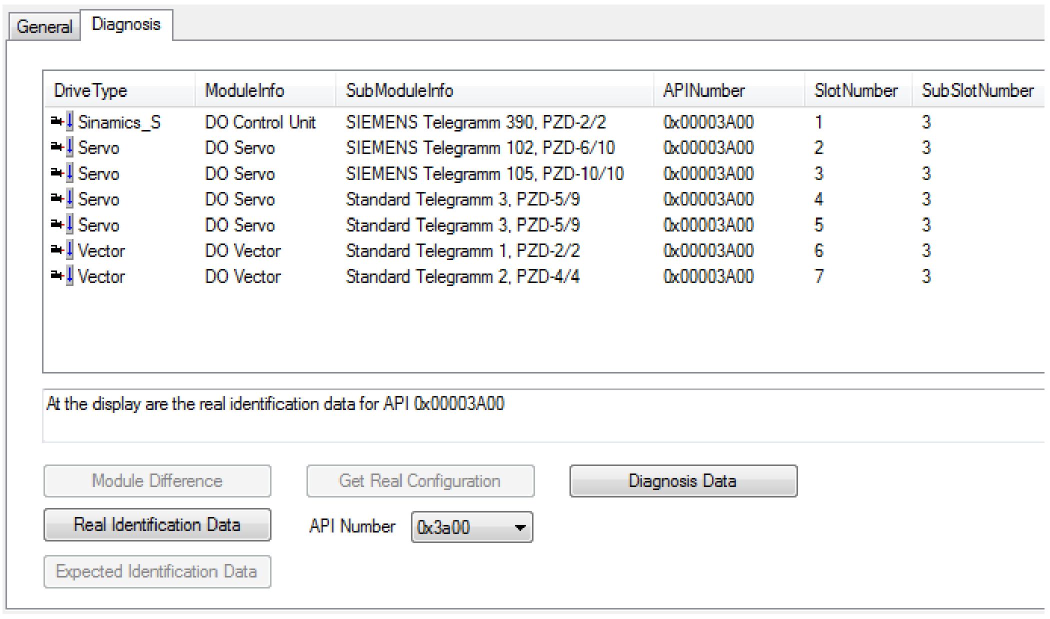


ModuleInfo	SubModuleInfo	APINumber	SlotNumber	SubSlotNumber
DAP Module	Interface 1	0x00000000	0	32768
DAP Module	Port 1	0x00000000	0	32769
DAP Module	Port 2	0x00000000	0	32770
DAP Module	EK9300 V2.25, 2 Port	0x00000000	0	1
EL1004	EL1004	0x00000000	1	1
EL2008	EL2008	0x00000000	2	1
EL3314	EL3314	0x00000000	3	1
EL4004	EL4004	0x00000000	4	1
EL2521	EL2521	0x00000000	5	1

At the display are the expected identification data from one AR.

Fig. 38: "Diagnosis" tab, checking the project planning

If you are on the “Diagnosis” tab within the API, you can select the corresponding API about which information is to be obtained. If, for example, the PROFINET device is a drive, then this usually supports the Profidrive profile, which is identified in turn via API 0x3A00. If the Real Identification Data is to be read from this API, for example, then this access takes place via the Profidrive profile.



DriveType	ModuleInfo	SubModuleInfo	APINumber	SlotNumber	SubSlotNumber
Sinamics_S	DO Control Unit	SIEMENS Telegramm 390, PZD-2/2	0x00003A00	1	3
Servo	DO Servo	SIEMENS Telegramm 102, PZD-6/10	0x00003A00	2	3
Servo	DO Servo	SIEMENS Telegramm 105, PZD-10/10	0x00003A00	3	3
Servo	DO Servo	Standard Telegramm 3, PZD-5/9	0x00003A00	4	3
Servo	DO Servo	Standard Telegramm 3, PZD-5/9	0x00003A00	5	3
Vector	DO Vector	Standard Telegramm 1, PZD-2/2	0x00003A00	6	3
Vector	DO Vector	Standard Telegramm 2, PZD-4/4	0x00003A00	7	3

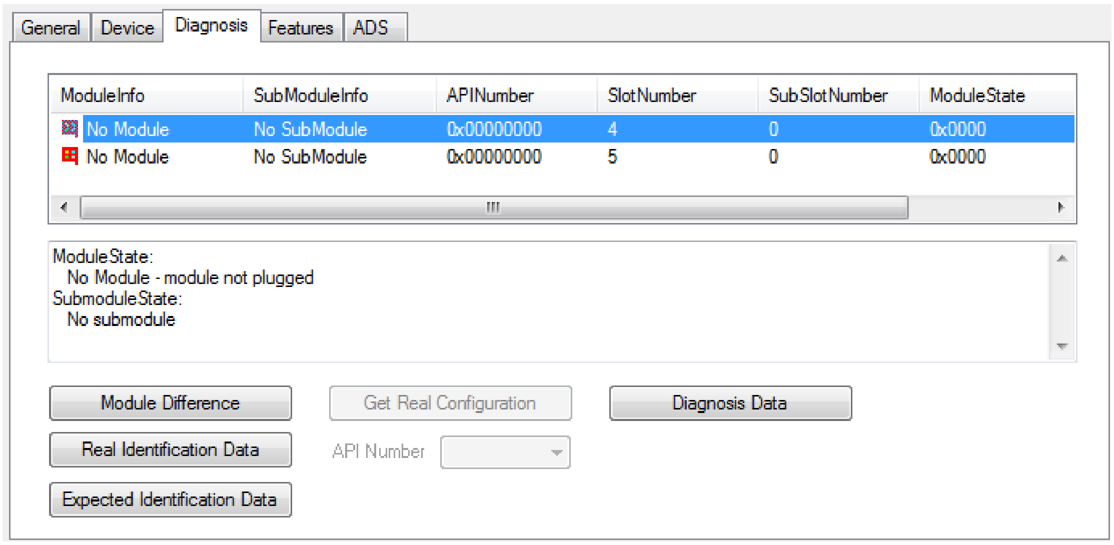
At the display are the real identification data for API 0x00003A00

API Number: 0x3a00

Fig. 39: "Diagnosis" tab, API selection

In addition, the “Get Real Configuration” button becomes active within an API (except for drives). It is possible via this to adopt the read-in data record into the current project. Note that modules that have already been created will be overwritten when doing this. This means that the links are lost, even in the case of previously correctly created modules.

When displaying the module differences, additional information is displayed by marking the message.



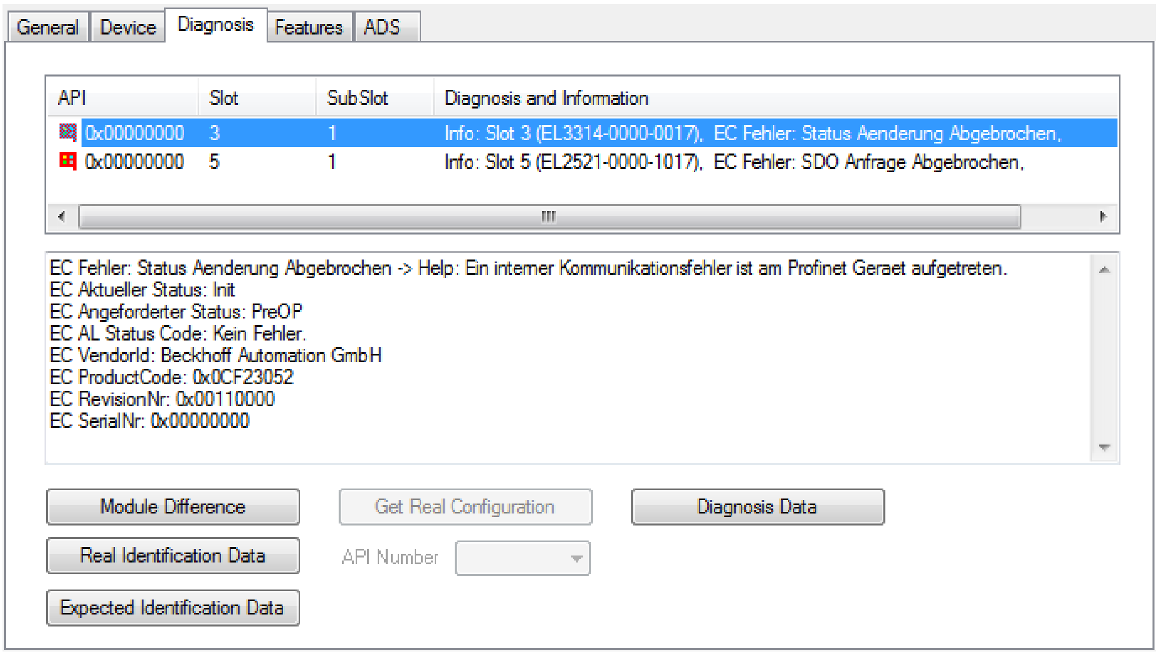
ModuleInfo	SubModuleInfo	APINumber	SlotNumber	SubSlotNumber	ModuleState
No Module	No SubModule	0x00000000	4	0	0x0000
No Module	No SubModule	0x00000000	5	0	0x0000

```
ModuleState:
  No Module - module not plugged
SubmoduleState:
  No submodule
```

Fig. 40: "Diagnosis" tab, adopt data set into project

Diagnosis Data

The available diagnosis can be read out by pressing the “Diagnosis Data” button. At device level all available diagnosis data for the existing AR is read out here.



API	Slot	SubSlot	Diagnosis and Information
0x00000000	3	1	Info: Slot 3 (EL3314-0000-0017), EC Fehler: Status Aenderung Abgebrochen,
0x00000000	5	1	Info: Slot 5 (EL2521-0000-1017), EC Fehler: SDO Anfrage Abgebrochen,

```
EC Fehler: Status Aenderung Abgebrochen -> Help: Ein interner Kommunikationsfehler ist am Profinet Geraet aufgetreten.
EC Aktueller Status: Init
EC Angeforderter Status: PreOP
EC AL Status Code: Kein Fehler.
EC VendorId: Beckhoff Automation GmbH
EC ProductCode: 0x0CF23052
EC RevisionNr: 0x00110000
EC SerialNr: 0x00000000
```

Fig. 41: "Diagnosis", Button Diagnosis Data

A maximum of two diagnosis parameters are displayed in the list; others are marked by “…”. If the individual message is clicked, all available diagnosis information is displayed in the window below.

Cyclic diagnosis via “PnIoBoxState” and “PnIoBoxDiag”

These variables are cyclically exchanged with the process image between the PROFINET driver and the System Manager.

Presently the following error messages are displayed under the PnIoBoxState.

Number	Text	Description	Remedial action / reason
0	No error	No error	No error
1	PROFINET Device state machine is in boot mode	PROFINET Device State Machine is still in the start-up phase	Not an error, wait
2	Device not found	Device does not reply to the Identify Request	Check connection, device connected, was the device called by its correct name?
3	The stationname is not unique	The station name is not unique	There are two or more devices in the network with the same PROFINET name. A correct identification cannot take place.
4	IP could not set	IP address could not be set.	The PROFINET device has rejected the IP settings for some reason. Check whether the IP settings are correct.
5	IP conflict	An IP conflict has occurred in the network.	A possible cause is that several devices have the same IP address.
6	DCP set was not successful	There was no reply or an erroneous reply to a DCP Set.	Check connection, device connected, was the device called by its correct name?
7	Watchdog error	The connection was broken off with a Watchdog error.	Check the cycle time, check the connection, if necessary increase the Watchdog factor.
8	Datahold error	The connection was broken off with a Datahold error.	Frame Data status was invalid for the length of the DataHoldTimer. Restart the device if necessary.
9	RTC3: Sync signal could not started	For IRT only: the Sync signal could not be started.	Is EtherCAT Sync signal correct or has Sync0 started?
10	PROFINET Controller has a link error	The PROFINET controller has no link.	Check cable and connection.
11	The aliasname is not unique	The alias name is not unique	There are two or more devices in the network with the same alias name. This is made up of the neighbourhood information (PortId.ChassisId). A correct identification cannot take place.
12	The automatic name assignment isn't possible - wrong device type	The automatic name assignment is not possible.	The expected PROFINET device is not in the projected position (VendorId or DeviceId does not correspond). Hence, no automatic naming and thus no device start is possible.
31	only for EtherCAT gateways: WC-State of cyclic EtherCAT frame is 1	For EL6631 only: EtherCAT WC State is 1	Check the mode on the EtherCAT master & slave (OP?).

As opposed to the state, more than one status can be displayed in the “PnIoBoxDiag”, i.e. the whole thing is bit-coded and up to 16 pieces of information can be displayed. The following statuses are currently displayed.

0x0000 = No diagnosis
0xXXX1 = IOC-AR is not established
0xXXX2 = IOC-AR is established
0xXXX4 = IOC-AR is established but no ApplReady
0xXXX8 = IOC-AR is established but module difference
0xXX1X = At least one AlarmCR get diagnosis alarm
0xX1XX = At least one InputCR is invalid
0xX2XX = At least one InputCR Provider is in stop
0xX4XX = At least one InputCR Problemindicator is set

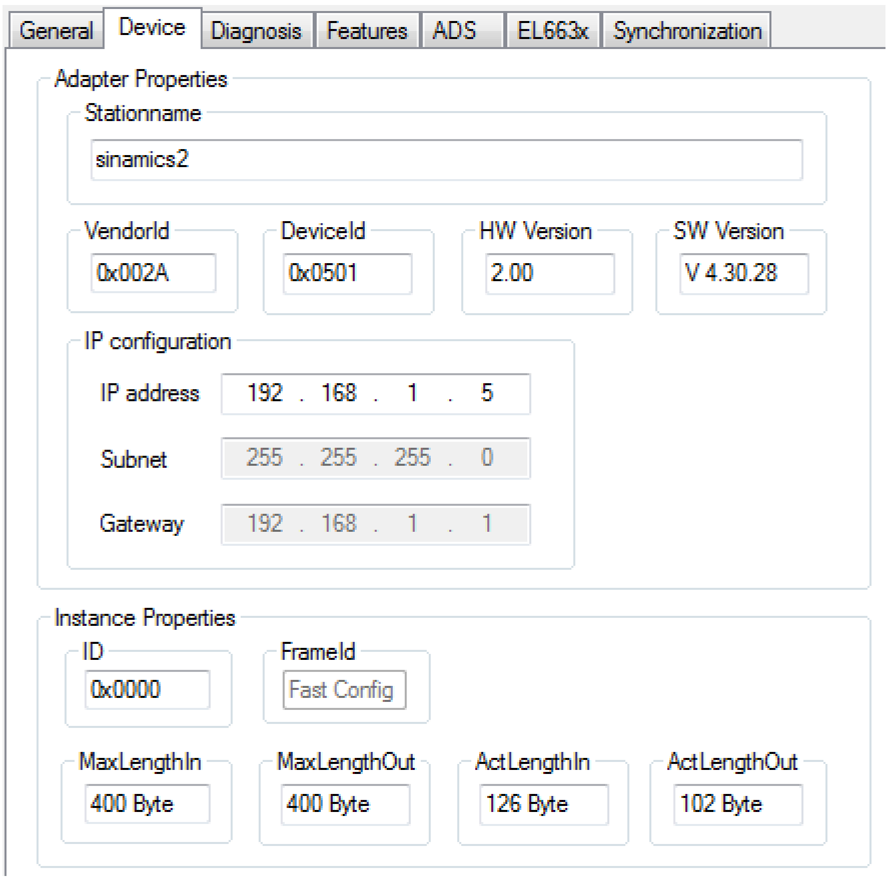
0x1XXX = At least one OutputCR is invalid
0x2XXX = At least one OutputCR Provider is in stop
0x4XXX = At least one OutputCR Problemindicator is set

On the one hand information about the status of the IO Controller Single AR is displayed here. In addition, collective statuses are formed from the Frame Data statuses of the individual CRs. The whole thing happens for the input and the output CRs (currently only one is possible; in future the controller will support several CRs). In addition a PROFINET alarm is also displayed in the “PnIoBoxDiag”

5.3 Settings

5.3.1 Project planning of the PROFINET device

When establishing a PROFINET connection the controller always assigns an IP address to the device from its own address space (if the device does not yet have one or if it has a different one). In TwinCAT the next higher address is taken for a device by default (starting from the controller adaptor class); the subnet and gateway are the same as those of the controller. Before the actual assignment of the IP address to the device by the controller, an ARP is used to test for a possible address conflict or to check whether the device already has this IP address. If there is a conflict, e.g. that the IP address is already assigned in the network, the IO driver determines this and outputs a corresponding message in the logger window. If there is no reply to the ARP, this means that no device (the projected device included) is using this IP configuration, which in turn results in the controller assigning the IP settings to the device via a DCP_SET. Setting is skipped if it is determined via the ARP that the device sought already has the projected IP address.



Stationname: sinamics2
VendorId: 0x002A
DeviceId: 0x0501
HW Version: 2.00
SW Version: V 4.30.28
IP address: 192 . 168 . 1 . 5
Subnet: 255 . 255 . 255 . 0
Gateway: 192 . 168 . 1 . 1
ID: 0x0000
FrameId: Fast Config
MaxLengthIn: 400 Byte
MaxLengthOut: 400 Byte
ActLengthIn: 126 Byte
ActLengthOut: 102 Byte

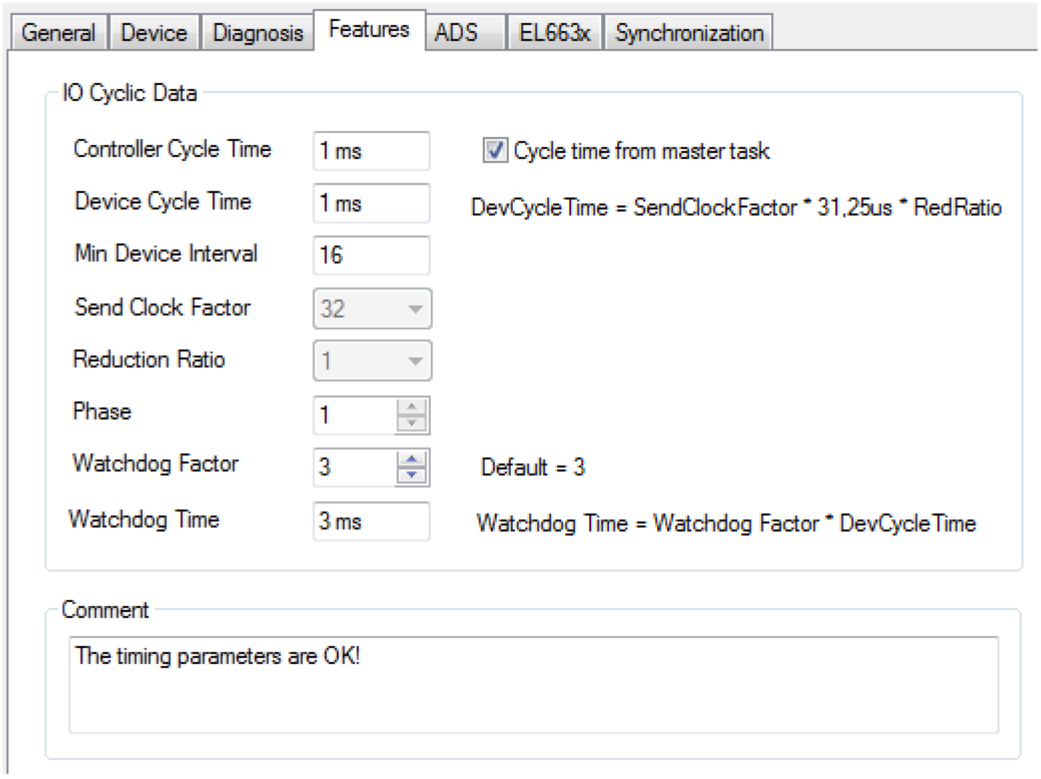
Fig. 42: "Device" tab

In addition the "InstanceID" and the "FrameID" can be changed in this window. However, the default settings are sufficient for most applications. The Instance ID is incorporated into the formation of the UUID object. A change should therefore be made only in exceptional cases. When changing the Frame ID the employed

RTClass must be taken into account (e.g. for RTClass1 unicast 0xC000 - 0xFAFF). If the device is on an IRT controller and all devices have been switched automatically to RTClass3, the FrameID is managed automatically and there is no input option (marked by “Fast Config”).

In addition the current process data length can be checked in this menu, i.e. the MaxLengths indicate which process data size is supported by the respective device, while the ActLengths indicate the current process data length (incl. IOPS and IOCS). The corresponding error message appears if the maximum lengths are exceeded on appending further modules/submodules.

Various settings for the cycle time can be made on the “Features” tab. The controller cycle time for RTCLass1 must always correspond to a power of two, starting from 1 ms (1, 2, 4, 8…). If an incorrect base time has been selected, this is indicated by a corresponding message. For RTClass3 the 1 ms base time can be divided again and again by two (down to min. 31.25 µs). The device cycle time can be changed via the exponents. The minimum is always the Controller Cycle Time, unless a larger minimum cycle time than that of the controller is defined in the GSDML. The maximum for RTClass1 is 512 ms. The “SendClockFactor” is fixed here as the time base with the value 32 (31.25 µs * 32 = 1 ms). The “ReductionRatioFactor” is also referenced to this, i.e. an RRFactor of 4 means a cycle time of 4 ms. The transmission point can be shifted again within a cycle via the phase; i.e. where RR = 4 the phase can be 1 - 4. However, this value is only of importance in the case of a synchronized transmission.



IO Cyclic Data

Controller Cycle Time	1 ms	☑ Cycle time from master task
Device Cycle Time	1 ms	DevCycleTime = SendClockFactor * 31,25us * RedRatio
Min Device Interval	16	
Send Clock Factor	32	
Reduction Ratio	1	
Phase	1	
Watchdog Factor	3	Default = 3
Watchdog Time	3 ms	Watchdog Time = Watchdog Factor * DevCycleTime

Comment

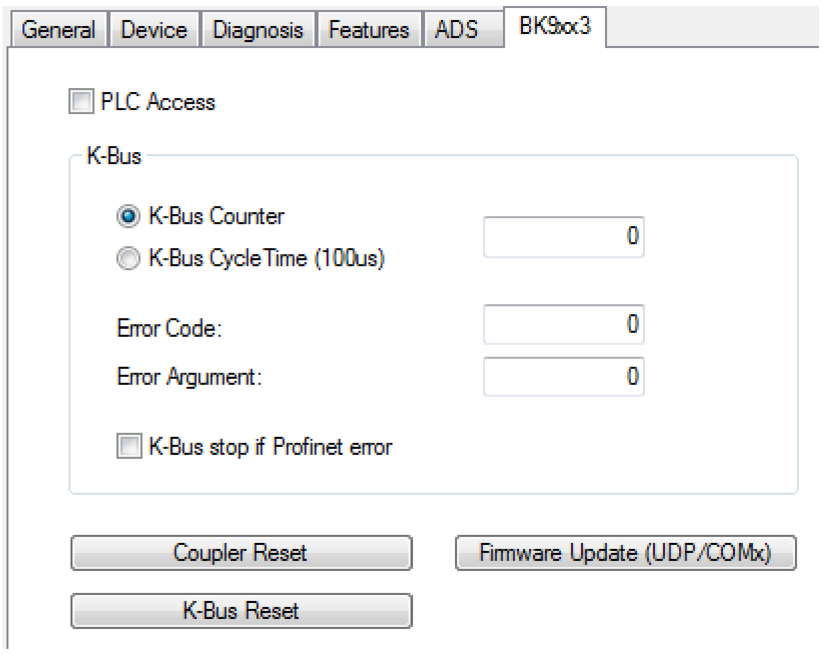
The timing parameters are OK!

Fig. 43: "Features" tab

In addition there is an option here to adjust the PROFINET Watchdog factor, i.e. each device monitors the input of the cyclic data on the basis of this factor. If the factor is set to the default value (3) this means that, with an RR of 4, three cycles require 12 ms. Hence, a device reacts after 12 ms to missing telegrams (e.g. with an alarm and/or disconnection of the AR). The limits and values are recalculated each time when adjusting the individual factors.

5.3.2 BK9xx3

In the case of the Beckhoff K-Bus Coupler (at present BK9103 or BK9053) that is not connected to an EL663x, an additional menu appears here.



☐ PLC Access

K-Bus

● K-Bus Counter: 0
○ K-Bus CycleTime (100us)

Error Code: 0

Error Argument: 0

☐ K-Bus stop if Profinet error

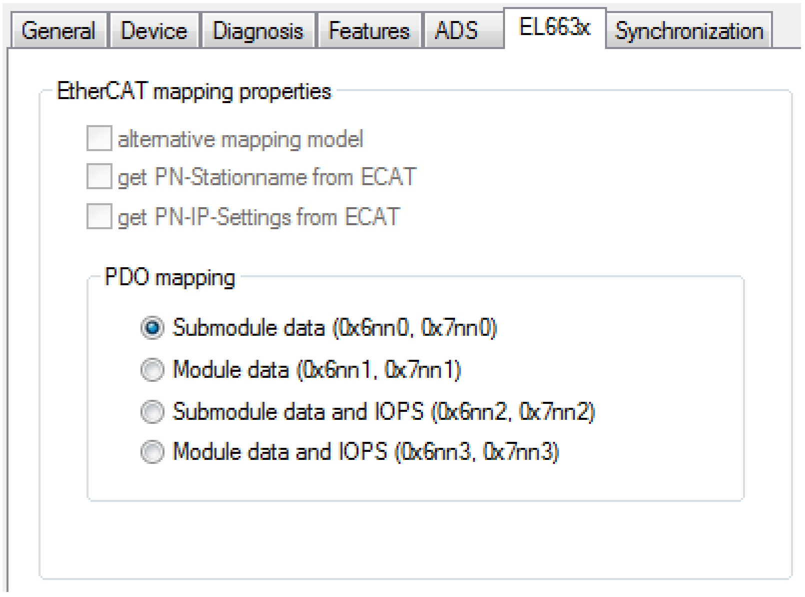
Fig. 44: "BK9xx3" tab

The cyclic process data in the DAP of the Bus Coupler can be accessed easily via this menu.

In addition, a firmware update from the System Manager to the Bus Coupler can be carried out via this menu. If the update takes place by IP it is important to ensure that the IP address is obtained via the DIP switches. If this is not the case the connection breaks off during the update, since the memory area of the IP settings is also formatted and rewritten.

5.3.3 EL663x

If the controller protocol is operated via an EL663x, then an additional menu appears on the devices.



EtherCAT mapping properties

☐ alternative mapping model
☐ get PN-Stationname from ECAT
☐ get PN-IP-Settings from ECAT

PDO mapping

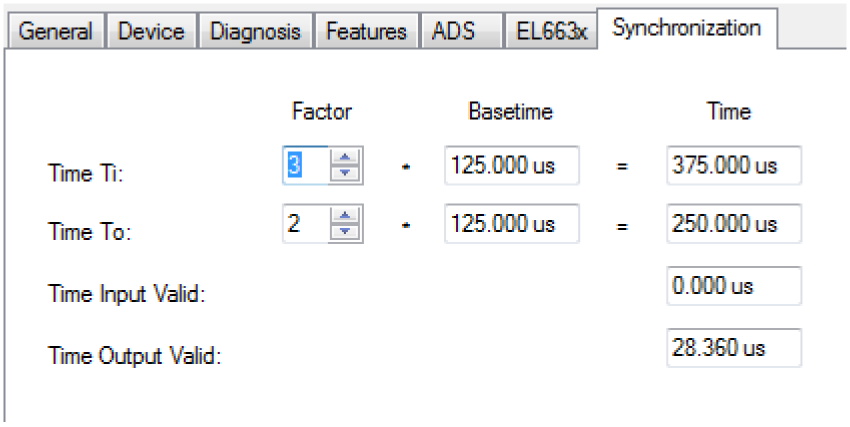
● Submodule data (0x6nn0, 0x7nn0)
○ Module data (0x6nn1, 0x7nn1)
○ Submodule data and IOPS (0x6nn2, 0x7nn2)
○ Module data and IOPS (0x6nn3, 0x7nn3)

Fig. 45: "EL663x" tab

At present only the choice of the PDO mapping is selectable for the controller; i.e. the form in which the PROFINET process data are mapped to the PDOs on the EtherCAT side is set here.

5.3.4 IRT Controller

If the device is operated on an IRT-capable controller, then an additional menu appears.



	Factor		Basetime		Time
Time Ti:	3	*	125.000 us	=	375.000 us
Time To:	2	*	125.000 us	=	250.000 us
Time Input Valid:					0.000 us
Time Output Valid:					28.360 us

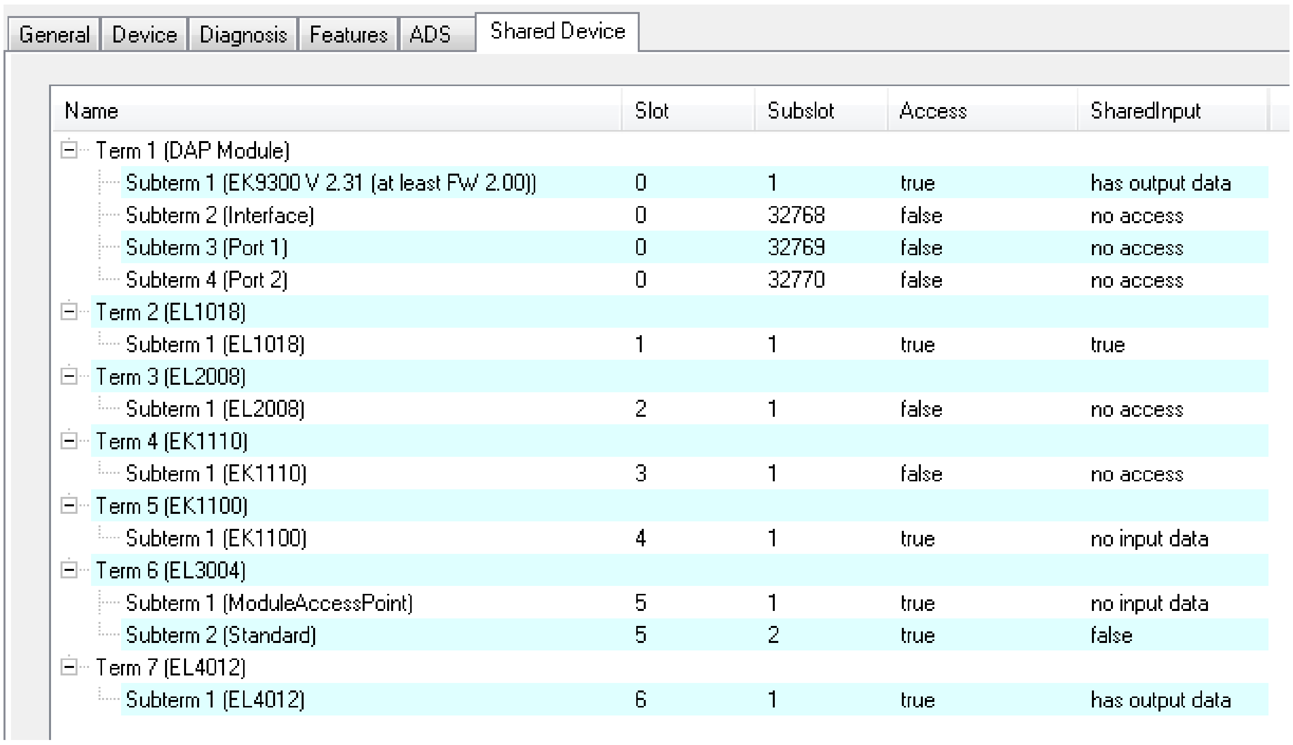
Fig. 46: "Synchronization" tab

It is possible to specify the Ti and To factors for IRT-capable devices via this menu. This means the time during which the data in the device are valid within a cycle, or should be set to valid. The prerequisite is that this feature is also supported. The GSMDL supplies the information about this. There is always a basic cycle here (base time). A statement about the minimum possible time comes via the GSDML on the basis of a minimum factor. The upper limit of the factor is limited by the cycle time employed. The shortest possible time in which the data could be valid over PROFINET (always in reference to the cycle) is displayed via the “Time Input Valid” or “Time Output Valid” parameter.

5.3.5 Shared Device

The SharedDevice feature is available from TwinCAT 2 build 22.50 or TwinCAT 3 build 4019.

The dialogue appears if the device supports "SharedDevice". The information for this comes from the GSDML.



Name	Slot	Subslot	Access	SharedInput
Term 1 (DAP Module)				
Subterm 1 (EK9300 V 2.31 (at least FW 2.00))	0	1	true	has output data
Subterm 2 (Interface)	0	32768	false	no access
Subterm 3 (Port 1)	0	32769	false	no access
Subterm 4 (Port 2)	0	32770	false	no access
Term 2 (EL1018)				
Subterm 1 (EL1018)	1	1	true	true
Term 3 (EL2008)				
Subterm 1 (EL2008)	2	1	false	no access
Term 4 (EK1110)				
Subterm 1 (EK1110)	3	1	false	no access
Term 5 (EK1100)				
Subterm 1 (EK1100)	4	1	true	no input data
Term 6 (EL3004)				
Subterm 1 (ModuleAccessPoint)	5	1	true	no input data
Subterm 2 (Standard)	5	2	true	false
Term 7 (EL4012)				
Subterm 1 (EL4012)	6	1	true	has output data

Fig. 47: "Shared Device" tab

There is an option here to allow or forbid the controller to access the individual sub-modules. By default the controller may access all sub-modules; if SharedInput is supported it is switched off.

The text messages for SharedInput have the following meanings:

- "not supported" - SharedInput is not supported by the device (info from the GSDML)
- "has output data" - the sub-module has outputs - activation of SharedInput not possible
- "no input data" - the sub-module has no inputs (and also no outputs)
- "no access" - access is blocked
- "true" or "false" - set value for SharedInput

The settings can be changed by double-clicking on the individual sub-modules. If the access to a port or interface sub-module is changed, then it is changed for all ports or interfaces.

5.4 Modules

5.4.1 Diagnosis at module level

The slot number of the modules always corresponds to the position in the tree, i.e. the DAP module always starts with slot number 0 and so on in order. At module level there is an option on the Diagnostic tab to compare the nominal and actual data for the respective module. In addition the existing diagnosis for the module can be read out.

5.4.2 Diagnosis at submodule level

PROFINET currently distinguishes between 4 types of submodule.

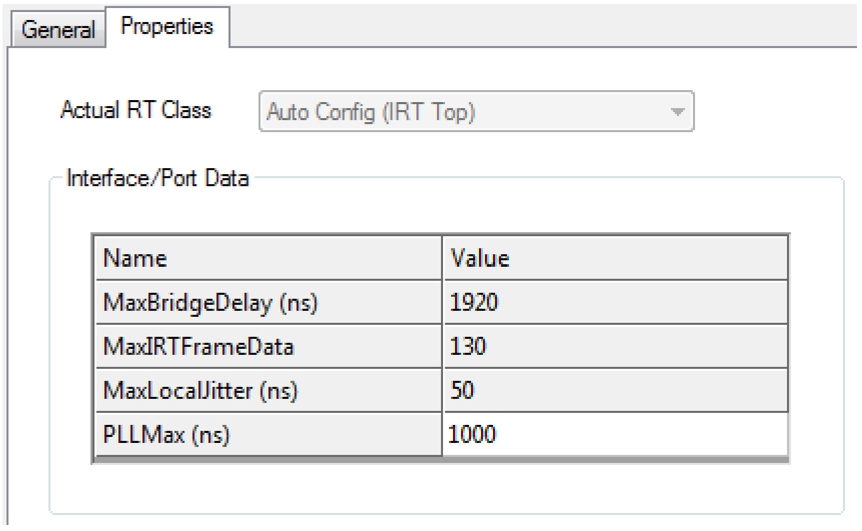
- Virtual submodules:
  The virtual submodules are always permanently connected to a module, i.e. when inserting a module the virtual submodules defined with it are also always inserted in the specified subslot. This kind of submodule is presently the commonest method.
- Real submodule:
  Here there is a possibility to select the pluggable submodules from a submodule list and to append them to the module. The necessary information is procured from the GSDML. In TwinCAT a module can be selected from such a list with the right mouse button (provided this is supported by the device).
- Port submodule:
  The physical properties of a network port are reproduced in such a submodule.
- Interface submodule:
  Device-specific properties are defined in the interfaces submodules. These can be, for example, additionally supported protocols, timing properties, supported MIBs etc.

In general the submodules have the same diagnostic properties as the modules, i.e. in this case also it is currently only possible to read out the nominal and actual configuration in TwinCAT. The order of the subslot numbers is not necessarily the same as the order in the TwinCAT project. Hence, for example, the order in DAP always starts with the interface submodule (ISM); however, the subslot number of the ISM is defined in the GSDML and starts at 0x8000. There are 16 possible interfaces (0x8x00), each with up to 256 ports (0x80xx). An ISM is followed by the Port submodule with the aforementioned subslot number.

5.4.3 Interface Submodule

The type of communication can always be set on the interface submodel (at present RTClass1 or RTClass3). The only exception is in the case that a generally valid RTClass was set via the ‘Auto Config...’ menu.

If communication takes place over RTClass3, then the PLL window can additionally be set at the interface.



Actual RT Class: Auto Config (IRT Top)

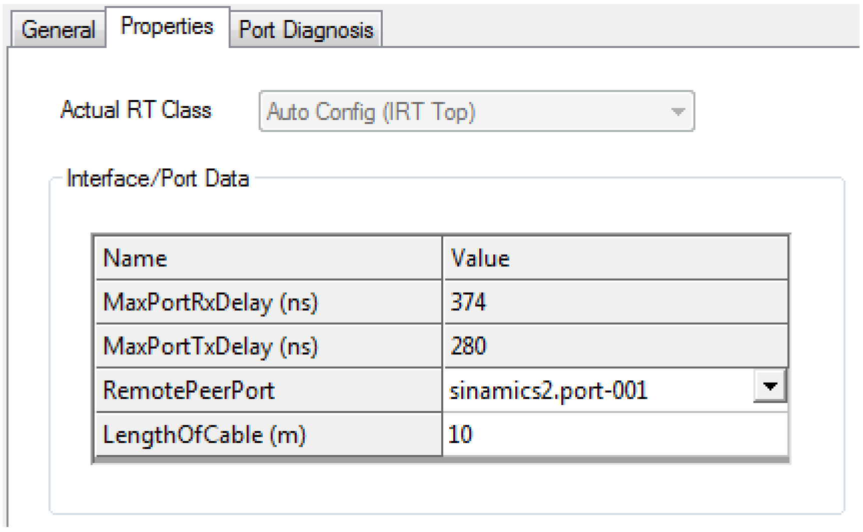
Interface/Port Data

Name	Value
MaxBridgeDelay (ns)	1920
MaxIRTFrameData	130
MaxLocalJitter (ns)	50
PLLMax (ns)	1000

Fig. 48: "Properties" tab, "PLL Window" setting

5.4.4 Port Submodule

Port-specific settings can be made on the ‘Properties’ tab. The menu of possible settings always depends on the RTClass employed.



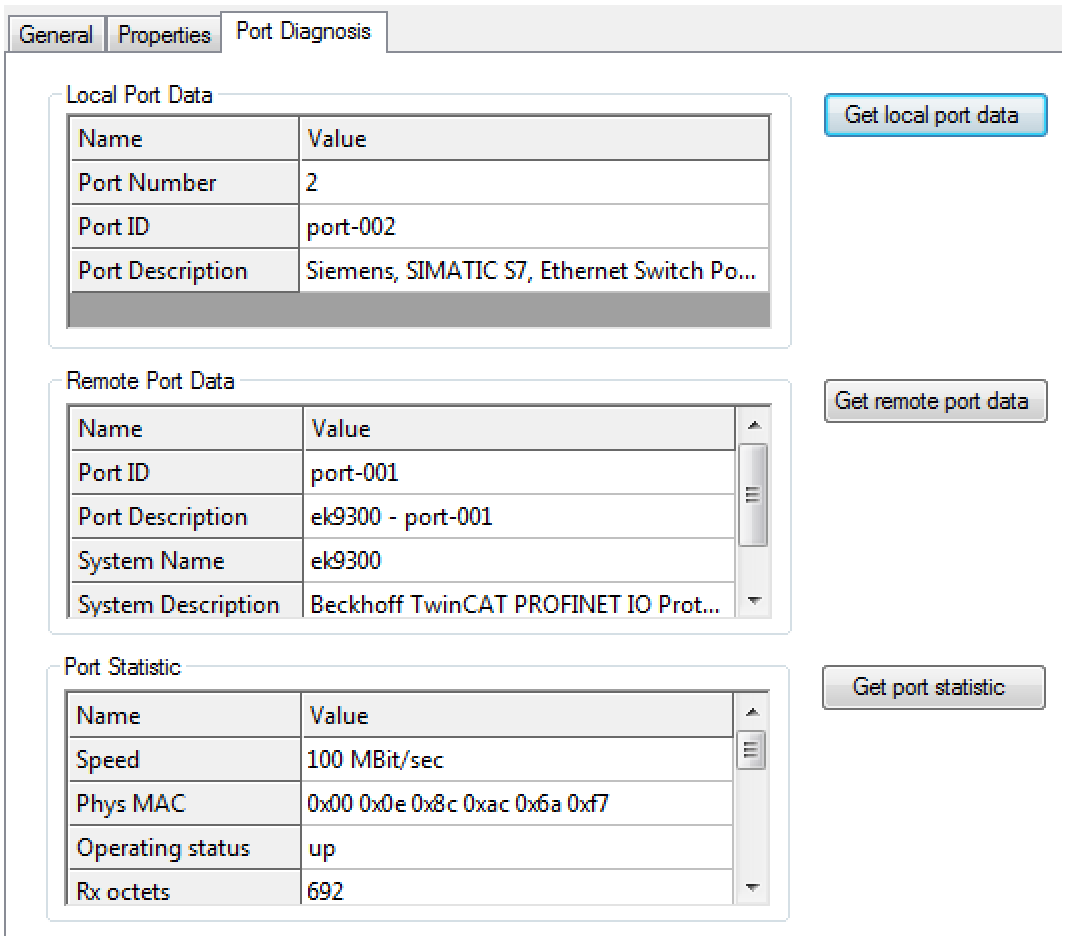
Actual RT Class: Auto Config (IRT Top)

Interface/Port Data

Name	Value
MaxPortRxDelay (ns)	374
MaxPortTxDelay (ns)	280
RemotePeerPort	sinamics2.port-001
LengthOfCable (m)	10

Fig. 49: "Properties" tab

In addition some port properties can be read out.



Local Port Data

Name	Value
Port Number	2
Port ID	port-002
Port Description	Siemens, SIMATIC S7, Ethernet Switch Po...

Remote Port Data

Name	Value
Port ID	port-001
Port Description	ek9300 - port-001
System Name	ek9300
System Description	Beckhoff TwinCAT PROFINET IO Prot...

Port Statistic

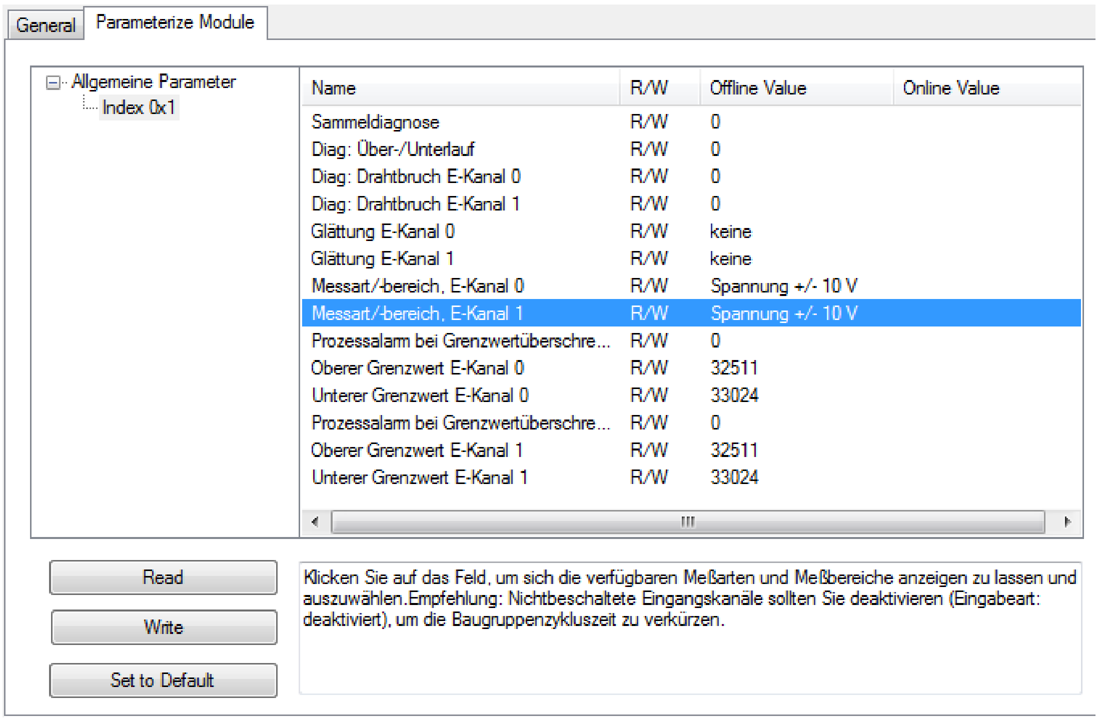
Name	Value
Speed	100 MBit/sec
Phys MAC	0x00 0x0e 0x8c 0xac 0x6a 0xf7
Operating status	up
Rx octets	692

Fig. 50: "Port Diagnosis" tab

The information here is subdivided into local port information and remote port properties. I.e. the LLDP protocol (IEEE Std 802.1AB) is prescribed in PROFINET from Conformance Class A (CCA). The devices exchange neighbourhood IDs via this protocol, so that each port is known to its neighbor. Furthermore, the Simple Network Management Protocol (SNMP) can be used as an aid at this point. On opening the ‘Port Diagnosis’ tab, TwinCAT acts as a Network Management Station (NMS) and collects the required device information via SNMP. In the above illustration, for example, it can be seen that the local Port 1 of the BK9053 is connected to Port 2 of the BK9103. For correct topology recognition it is important that only devices are present in the strand that also support the LLDP protocol (this is also applies to switches!).

5.4.5 Real / virtual submodules

If these submodules have paramétrisation data they will be displayed as shown in the illustration below.



Name	R/W	Offline Value	Online Value
Sammeldiagnose	R/W	0	
Diag: Über-/Unterlauf	R/W	0	
Diag: Drahtbruch E-Kanal 0	R/W	0	
Diag: Drahtbruch E-Kanal 1	R/W	0	
Glättung E-Kanal 0	R/W	keine	
Glättung E-Kanal 1	R/W	keine	
Messart/-bereich, E-Kanal 0	R/W	Spannung +/- 10 V	
Messart/-bereich, E-Kanal 1	R/W	Spannung +/- 10 V	
Prozessalarm bei Grenzwertüberschre...	R/W	0	
Oberer Grenzwert E-Kanal 0	R/W	32511	
Unterer Grenzwert E-Kanal 0	R/W	33024	
Prozessalarm bei Grenzwertüberschre...	R/W	0	
Oberer Grenzwert E-Kanal 1	R/W	32511	
Unterer Grenzwert E-Kanal 1	R/W	33024	

Klicken Sie auf das Feld, um sich die verfügbaren Meßarten und Meßbereiche anzeigen zu lassen und auszuwählen.Empfehlung: Nichtbeschaltete Eingangskanäle sollten Sie deaktivieren (Eingabeart: deaktiviert), um die Baugruppenzykluszeit zu verkürzen.

Fig. 51: "Parameterize Module" tab, parameter display

Selection can be made here between the individual indices. The data can be read and/or written depending on the access method. The online values are updated when reading back. If an individual index is marked, then all values within an index will be set to default; if individual values are marked, only these will be reset. Writable values are changed by double-clicking on the respective line.

6 TwinCAT library and programming

6.1 Overview

There are ready-to-use function blocks for the use of the Profinet controller. The library contains further function blocks for the EL6631-0010 PROFINET Device Terminal, but these are not part of this documentation.

(http://infosys.beckhoff.com/content/1033/el6631_el6632/Resources/zip/2595517963.zip)

I&M functions

Block I&M functions	Meaning	Description
FB_PN_IM0_READ [▸ 62]	Read the I&M function 0	Supplement and EL663x
FB_PN_IM1_READ [▸ 63]	Read the I&M function 1	Supplement and EL663x
FB_PN_IM2_READ [▸ 65]	Read the I&M function 2	Supplement and EL663x
FB_PN_IM3_READ [▸ 67]	Read the I&M function 3	Supplement and EL663x
FB_PN_IM4_READ [▸ 70]	Read the I&M function 4	Supplement and EL663x
FB_PN_IM1_WRITE [▸ 64]	Write the I&M function 1	Supplement and EL663x
FB_PN_IM2_WRITE [▸ 66]	Write the I&M function 2	Supplement and EL663x
FB_PN_IM3_WRITE [▸ 68]	Write the I&M function 3	Supplement and EL663x
FB_PN_IM4_WRITE [▸ 70]	Write the I&M function 4	Supplement and EL663x

Statistics and diagnostic information

Block I&M functions	Meaning	Description
FB_PN_GET_PORT_STATISTIC [▸ 71]	Read the port statistics	Supplement and EL663x
FB_PN_READ_PORT_DIAG [▸ 72]	Read the port diagnosis	Supplement and EL663x

Development environment	Target platform	PLC libraries to include
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcEtherCAT.lib TcPlcIoFunction.lib TcUtilities.lib TcSystem.lib TcBase.lib

6.2 Functions

6.2.1 I&M

6.2.1.1 FUNCTION_BLOCK FB_PN_IM0_READ

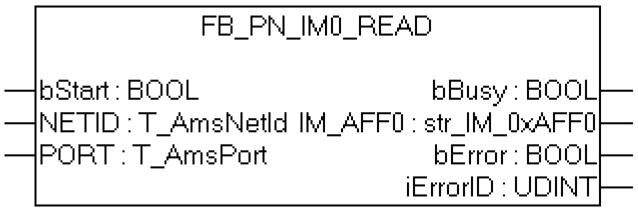


Fig. 52: FUNCTION_BLOCK FB_PN_IM0_READ

Using this function block the PROFINET controller reads all I&M 0 data (Identification & Maintenance) from a device referenced via the *Port* input.
The frame structure of the I&M0 function corresponds to the index 0xAFF0 [▶ 75] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart  : BOOL;
    NETID   : T_AmsNetId; (* AMS Net ID from Controller *)
    PORT    : T_AmsPort;  (* Port used by Controller to communicate with Device *)
END_VAR
```

bStart: The block is activated by a rising edge on this input

NETID: AMS Net ID des Controllers

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy   : BOOL;
    IM_AFF0 : str_IM_0xAFF0;
    bError  : BOOL;
    iErrorID    : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

IM_AFF0: Output of the I&M0 frame supplied by the device in a structure. str_IM_0xAFF0.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.1.2 FUNCTION_BLOCK FB_PN_IM1_READ

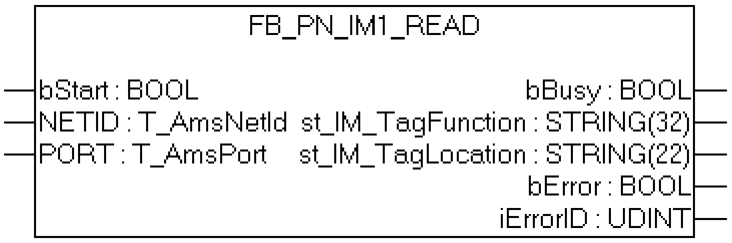


Fig. 53: FUNCTION_BLOCK FB_PN_IM1_READ

Using this function block the PROFINET controller reads all I&M1 data (Information & Maintenance) from a device referenced via the *Port* input.
The frame structure of the I&M1 function corresponds to the index 0xAFF1 [▸ 75] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart   : BOOL;
    NETID    : T_AmsNetId; (* AMS Net ID from Controller *)
    PORT     : T_AmsPort;  (* Port used by Controller to communicate with Device *)
END_VAR
```

bStart: The block is activated by a rising edge on this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy        : BOOL;
    st_IM_TagFunction : STRING(32);
    st_IM_TagLocation : STRING(22);
    bError       : BOOL;
    iErrorID         : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

st_IM_TagFunction: label read out for the function of the device.

st_IM_TagLocation: label read out for the installation site of the device.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.1.3 FUNCTION_BLOCK FB_PN_IM1_WRITE

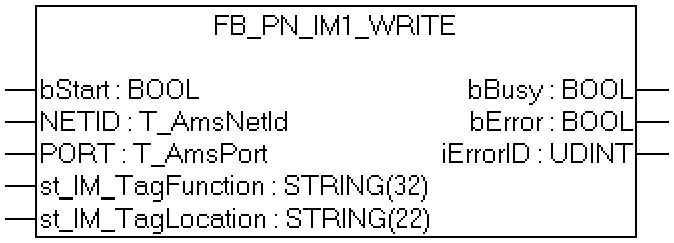


Fig. 54: FUNCTION_BLOCK FB_PN_IM1_WRITE

Using this function block the PROFINET controller writes all I&M1 data (Identification & Maintenance) data to a device referenced via the *Port* input.
The frame structure of the I&M1 function corresponds to the index 0xAFF1 [▶ 75] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart      : BOOL;
    NETID   : T_AmsNetId; (* AMS Net ID from Controller *)
    PORT    : T_AmsPort;  (* Port used by Controller to communicate with Device *)
    st_IM_TagFunction    : STRING(32);
    st_IM_TagLocation    : STRING(22);
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

st_IM_TagFunction: With this string the functional description is saved to the device.

st_IM_TagLocation: With this string the installation site is saved to the device.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    iErrorID        : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set..

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.1.4 FUNCTION_BLOCK FB_PN_IM2_READ

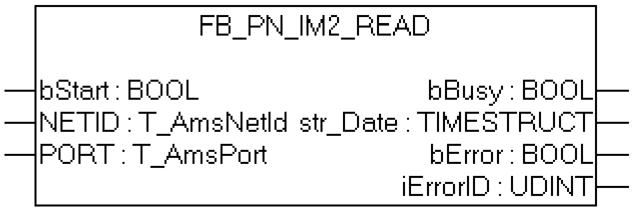


Fig. 55: FUNCTION_BLOCK FB_PN_IM2_READ

Using this function block the PROFINET controller reads all I&M 2 data (Identification & Maintenance) from a device referenced via the *Port* input.
The frame structure of the I&M2 function corresponds to the index 0xAFF2 [▶ 76] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart   : BOOL;
    NETID    : T_AmsNetId;(* AMS Net ID from Controller *)
    PORT     : T_AmsPort;  (* Port used by Controller to communicate with Device *)
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    str_Date        : TIMESTRUCT;  (*YYYY-MM-DD HH:MM*)
    bError      : BOOL;
    iErrorID        : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

str_Date: Returns the date of installation of the device in the format < YYYY-MM-DD HH:MM >.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.1.5 FUNCTION_BLOCK FB_PN_IM2_WRITE

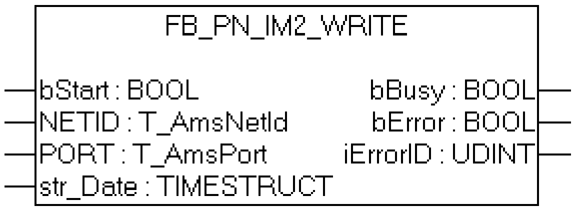


Fig. 56: FUNCTION_BLOCK FB_PN_IM2_WRITE

Using this function block the PROFINET controller writes all I&M 2 data (Identification & Maintenance) data to a device referenced via the *Port* input.
The frame structure of the I&M2 function corresponds to the index 0xAFF2 [▸ 76] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart   : BOOL;
    NETID    : T_AmsNetId;(* AMS Net ID from Controller *)
    PORT     : T_AmsPort; (* Port used by Controller to communicate with Device *)
    str_Date : TIMESTRUCT;(*YYYY-MM-DD HH:MM*)
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000hex)

str_Date : Writes a date (e.g. date of installation of the device) to the device in the format < YYYY-MM-DD HH:MM >.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    iErrorID        : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

bError: If an error should occur during the transmission of the command, this output is set after the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.1.6 FUNCTION_BLOCK FB_PN_IM3_READ

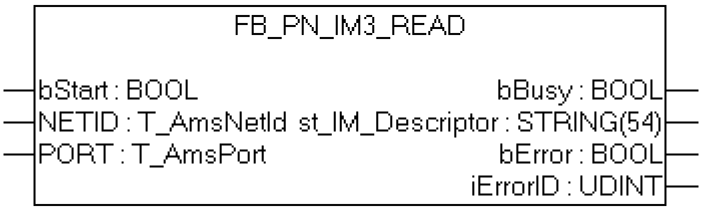


Fig. 57: FUNCTION_BLOCK FB_PN_IM3_READ

Using this function block the PROFINET controller reads all I&M3 data (Identification & Maintenance) from a device referenced via the *Port* input.
The frame structure of the I&M3 function corresponds to the index 0xAFF3 [▸ 76] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart  :
    BOOL; NETID   : T_AmsNetId;(* AMS Net ID from Controller *)
    PORT    : T_AmsPort;  (* Port used by Controller to communicate with Device *)
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    st_IM_Descriptor    : STRING(54);
    bError      : BOOL;
    iErrorID        : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

st_IM_Descriptor: Returns the manufacturer description stored for the device.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.1.7 FUNCTION_BLOCK FB_PN_IM3_WRITE

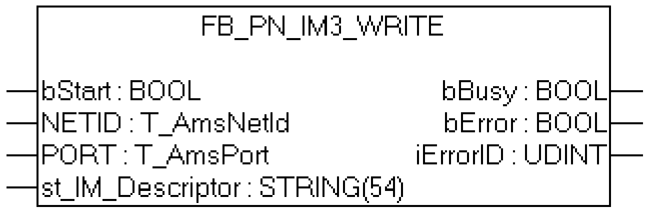


Fig. 58: FUNCTION_BLOCK FB_PN_IM3_WRITE

Using this function block the PROFINET controller writes all I&M3 data (Identification & Maintenance) data to a device referenced via the *Port* input.
The frame structure of the I&M3 function corresponds to the index 0xAFF3 [▶ 76] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart      : BOOL;
    NETID   : T_AmsNetId;(* AMS Net ID from Controller *)
    PORT    : T_AmsPort; (* Port used by Controller to communicate with Device *)
    st_IM_Descriptor     : STRING(54);
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

st_IM_Descriptor: Returns the manufacturer description stored for the device.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    iErrorID        : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.1.8 FUNCTION_BLOCK FB_PN_IM4_READ

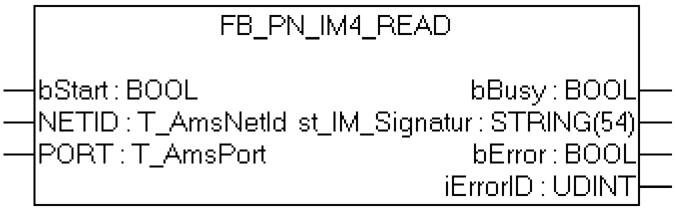


Fig. 59: FUNCTION_BLOCK FB_PN_IM4_READ

Using this function block the PROFINET controller reads all I&M4 data (Identification & Maintenance) from a device referenced via the *Port* input.
The frame structure of the I&M4 function corresponds to the index 0xAFF4 [▶ 76] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart   : BOOL;
    NETID    : T_AmsNetId; (* AMS Net ID from Controller *)
    PORT     : T_AmsPort;  (* Port used by Controller to communicate with Device *)
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    st_IM_Signatur   : STRING(54);
    bError      : BOOL;
    iErrorID        : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

st_IM_Signatur: Returns the manufacturer signature stored for the device.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.1.9 FUNCTION_BLOCK FB_PN_IM4_WRITE

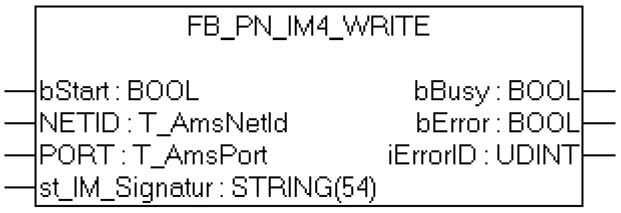


Fig. 60: FUNCTION_BLOCK FB_PN_IM4_WRITE

Using this function block the PROFINET controller writes all I&M4 data (Identification & Maintenance) data to a device referenced via the *Port* input.
The frame structure of the I&M4 function corresponds to the index 0xAFF4 [▶ 76] according to PROFINET standard.

VAR_INPUT

```
VAR_INPUT
    bStart      : BOOL;
    NETID   : T_AmsNetId;(* AMS Net ID from Controller *)
    PORT    : T_AmsPort; (* Port used by Controller to communicate with Device *)
    st_IM_Signatur   : STRING(54);
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

st_IM_Signatur: Signature of the manufacturer is written to the device.

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    bError      : BOOL;
    iErrorID        : UDINT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

bError: In the event of an error during the command transfer, this output is set once the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.2 Port

6.2.2.1 FUNCTION_BLOCK FB_PN_GET_PORT_STATISTIC

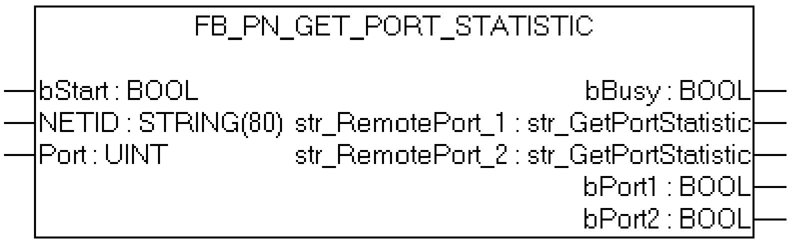


Fig. 61: FUNCTION_BLOCK FB_PN_GET_PORT_STATISTIC

When called, this module supplies the statistical data for the ports of a PROFINET device.

VAR_INPUT

```
VAR_INPUT
    bStart      : BOOL;
    NETID   : T_AmsNetId; (* AMS Net ID from Controller *)
    PORT    : T_AmsPort;  (* Port used by Controller to communicate with Device *)
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    str_RemotePort_1     :  str_GetPortStatistic [▸ 77];
    str_RemotePort_2     :  str_GetPortStatistic [▸ 77];
    bPort1      : BOOL;
    bPort2      : BOOL;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

str_RemotePort_1: This structure contains the statistical data for Port 1.

str_RemotePort_2: This structure contains the statistical data for Port 2.

bPort1: Is TRUE if the port has a link.

bPort2: Is TRUE if the port has a link.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.2.2.2 FUNCTION_BLOCK FB_PN_READ_PORT_DIAG

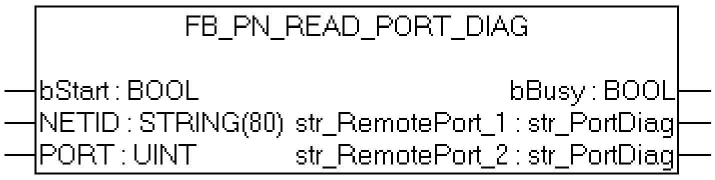


Fig. 62: FUNCTION_BLOCK FB_PN_READ_PORT_DIAG

This block calls the port diagnostic information of a PROFINET device.

VAR_INPUT

```
VAR_INPUT
    bStart      : BOOL;
    NETID   : T_AmsNetId;(* AMS Net ID from Controller *)
    PORT    : T_AmsPort; (* Port used by Controller to communicate with Device *)
END_VAR
```

bStart: The function block is activated by a positive edge at this input

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    str_RemotePort_1    : str_PortDiag [▶ 78];
    str_RemotePort_2    : str_PortDiag [▶ 78];
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

str_RemotePort_1: This structure contains the diagnostic information for Port 1.

str_RemotePort_2: This structure contains the diagnostic information for Port 2.

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcPROFINETDiag.Lib

6.2.3 AlarmDiag

6.2.3.1 FUNCTION_BLOCK FB_PN_ALARM_DIAG

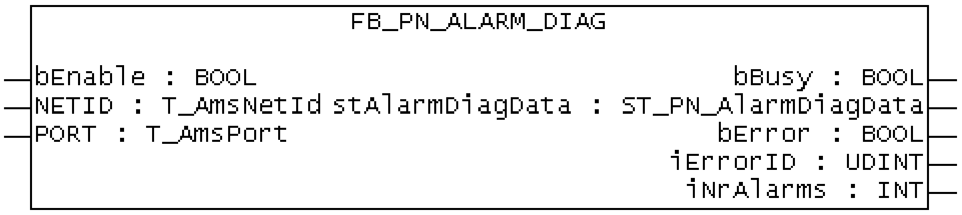


Fig. 63: FUNCTION_BLOCK FB_PN_ALARM_DIAG

Diagnosis alarms can be read out using this function block. Each instance of this function block makes a PLC input available ("PnIoBoxDiag"). This input must be linked to the “PnIoBoxDiag” input of the device that is to be evaluated. After successful reading of the diagnosis alarms/warnings, the alarm status of the device is reset. The function block must be called once for each PROFINET device. A running index (iNrAlarms) indicates how many diagnosis alarms have been read from the buffer.

VAR_INPUT

```
VAR_INPUT
    bEnable : BOOL;
    NETID   : T_AmsNetId; (* AMS Net ID from Controller *)
    PORT    : T_AmsPort;  (* Port used by Controller to communicate with Device *)
END_VAR
```

bEnable: Activation of the function block

NETID: AMS Net ID of the controller

PORT: Port via which the controller communicates with the device (port = Device ID + 1000 hex)

VAR_OUTPUT

```
VAR_OUTPUT
    bBusy       : BOOL;
    stAlarmDiagData : ST_PN_AlarmDiagData;
    bError      : BOOL;
    iErrorID        : UDINT;
    iNrAlarms           : INT;
END_VAR
```

bBusy: When the function block is activated this output is set. It remains set until a feedback is received. While Busy = TRUE, no new command will be accepted at the inputs.

stAlarmDiagData: Diagnosis messages are output via this structure. An alarm is output via the structure in each cycle as long as the status bit [0x0010 = at least one AlarmCR got a diagnosis alarm] is present at the PLC input.

bError: If an error should occur during the transmission of the command, this output is set after the *bBusy* output has been reset.

iErrorID: Supplies an ADS error number when the output *bError* is set.

iNrAlarms: Number of alarms last read out.

VAR

```
VAR
        PnIoBoxDiag AT %I* : WORD; (*Hardware Input*)
END_VAR
```

PnIoBoxDiag : Hardware input: this variable must be linked to the PROFINET device. A change of state of this variable indicates to the PLC program that there are new diagnosis alarms in the linked PROFINET device.

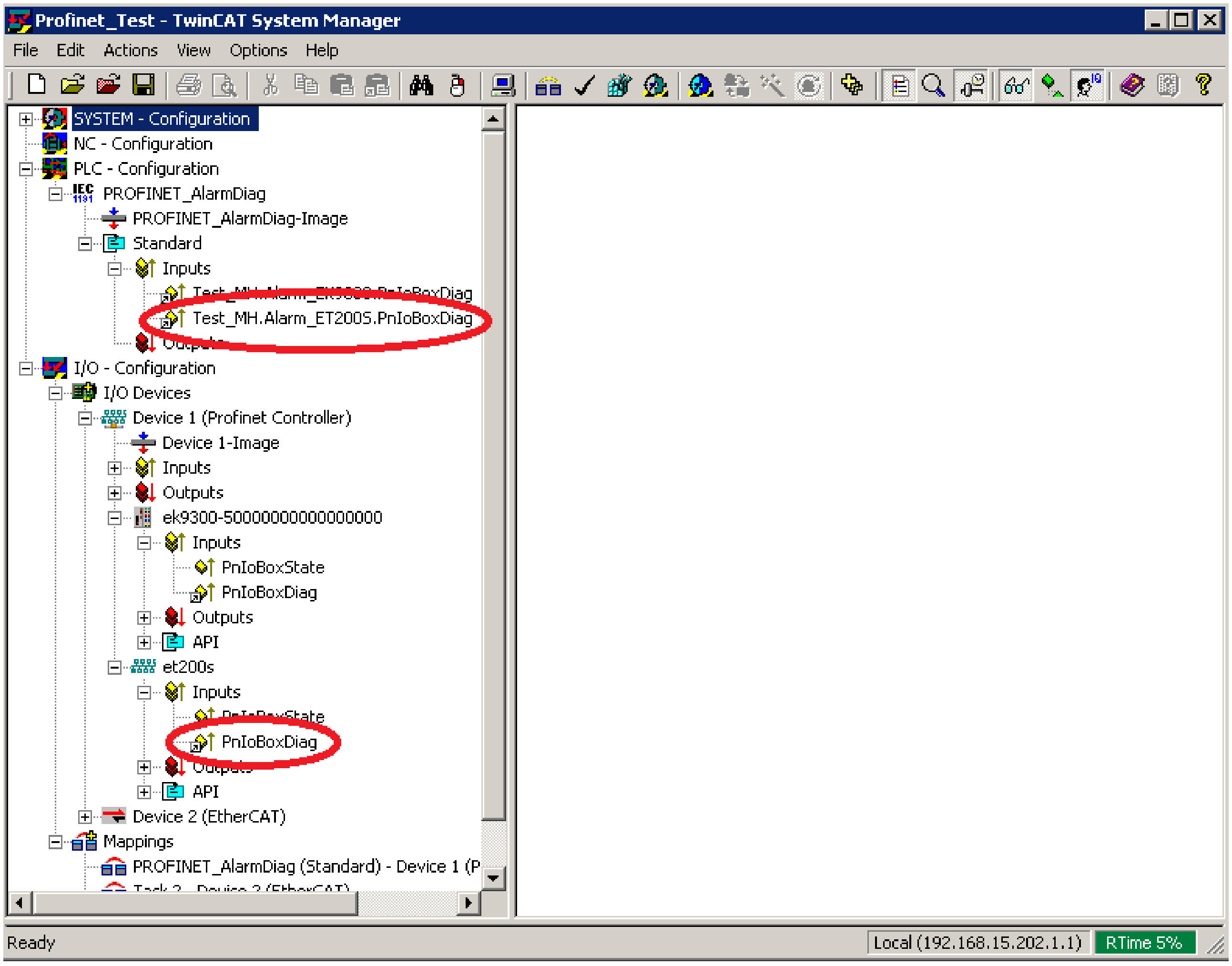


Fig. 64: Linking of the variables in the TwinCAT tree

Development environment	Target platform	PLC libraries to be linked
TwinCAT v2.11.0 R3	PC or CX (x86, ARM)	TcProfinetDiag.Lib

6.3 Data Structures

6.3.1 I&M

6.3.1.1 str_SW_Rec

The data structure **str_IM_0xAFF0** maps the structure of the I&M0 frame in the PLC. Which contains information that is permanently stored in PROFINET devices.

```
TYPE str_IM_0xAFF0 :

STRUCT
    nBlockTyp        : WORD;
    nBlockLen        : WORD;
    nBlockVersion    : WORD;
    nVendorID        : WORD;
    cOrderID         : STRING(21);
    cSerialNumber    : STRING(17);
    nHW_Rev          : WORD;
    strSW_Rev        : str_SW_Rec;
    nRevCount        : WORD;
    nProfileID       : WORD;
    nProfileSpecType     : WORD;
    arIM_Version     : ARRAY[0..1] OF BYTE;
    nSupport         : WORD;
END_STRUCT

END_TYPE
```

The data structure **str_SW_REC** contains the software version of the PROFINET device.

```
TYPE str_SW_Rec :

STRUCT
    cSWRevPrefix    :STRING(2);
    nSWRevFuncEnhance       :BYTE;
    nSWRevBugFix    :BYTE;
    nSWRevIntCha    :BYTE;
END_STRUCT

END_TYPE
```

6.3.1.2 str_IM_0xAFF1

The data structure **str_IM_0xAFF1** maps the structure of the I&M1 frame in the PLC. This structure is used both for writing to, and for reading from a PROFINET device.

```
TYPE str_IM_0xAFF1 :

STRUCT
    nBlockTyp        : WORD;
    nBlockLen        : WORD;
    nBlockVersion    : WORD;
    st_IM_TagFunction : STRING(32);
    st_IM_TagLocation : STRING(22);
END_STRUCT

END_TYPE
```

6.3.1.3 str_IM_0xAFF2

The data structure **str_IM_0xAFF2** maps the structure of the I&M2 frame in the PLC. This structure is used both for writing to, and for reading from a PROFINET device.

```
TYPE str_IM_0xAFF1 :

STRUCT
    nBlockTyp       : WORD;
    nBlockLen       : WORD;
    nBlockVersion   : WORD;
    st_IM_Date      : STRING(16);
END_STRUCT

END_TYPE
```

6.3.1.4 str_IM_0xAFF3

The data structure **str_IM_0xAFF3** maps the structure of the I&M3 frame in the PLC. This structure is used both for writing to, and for reading from a PROFINET device.

```
TYPE str_IM_0xAFF3 :

STRUCT
    nBlockTyp       : WORD;
    nBlockLen       : WORD;
    nBlockVersion   : WORD;
    st_IM_Descriptor    : STRING(54)
END_STRUCT

END_TYPE
```

6.3.1.5 str_IM_0xAFF4

The data structure **str_IM_0xAFF4** maps the structure of the I&M4 frame in the PLC. This structure is used both for writing to, and for reading from a PROFINET device.

```
TYPE str_IM_0xAFF3 :

STRUCT
    nBlockTyp       : WORD;
    nBlockLen       : WORD;
    nBlockVersion   : WORD;
    st_IM_Signatur  : STRING(54)
END_STRUCT

END_TYPE
```

6.3.2 Port

6.3.2.1 str_GetPortStatistic

All statistical information of a device is represented in the data structure **str_GetPortStatistic**.

```
TYPE str_GetPortStatisitc :

STRUCT
    Speed              : DWORD;
    PhyMAC             : STRING(50);
    OperatingStatus    : STRING(16);
    RxOctets           : DWORD;
    RxUniCastPackets   : DWORD;
    RxBadPackets       : DWORD;
    RxDroppedFrames    : DWORD;
    RxUnknownProtocol  : DWORD;
    TxOctets           : DWORD;
    TxUniCastPackets   : DWORD;
    TxBadPackets       : DWORD;
    TxDroppedPackets   : DWORD;
END_STRUCT

END_TYPE
```

6.3.2.2 str_PortDiag

All port diagnostic information is represented in the data structure **str_PortDiag**.

```
TYPE str_PortDiag :

STRUCT
    PortId              : STRING(128);
    PortDescription     : STRING(128);
    SystemName          : STRING(128);
    SystemDescription   : STRING(128);
    ChassisId           : STRING(128);
END_STRUCT

END_TYPE
```

6.3.3 AlarmDiag

6.3.3.1 ST_PN_DiagMessage

The data structure **ST_PN_DiagMessage** contains the complete data stream of a diagnostic message that is sent by a PROFINET on request. This data stream is evaluated in the FB_PN_ALARM_DIAG function block and is copied to a readable structure.

```
TYPE ST_PN_DiagMessage :

STRUCT
    nFlags    : WORD;
    nTextID   : WORD;
    TimeStamp : ARRAY[0..7] OF BYTE;
    nData     : ARRAY[0..299] OF BYTE;
END_STRUCT

END_TYPE
```

6.3.3.2 ST_PN_Diag

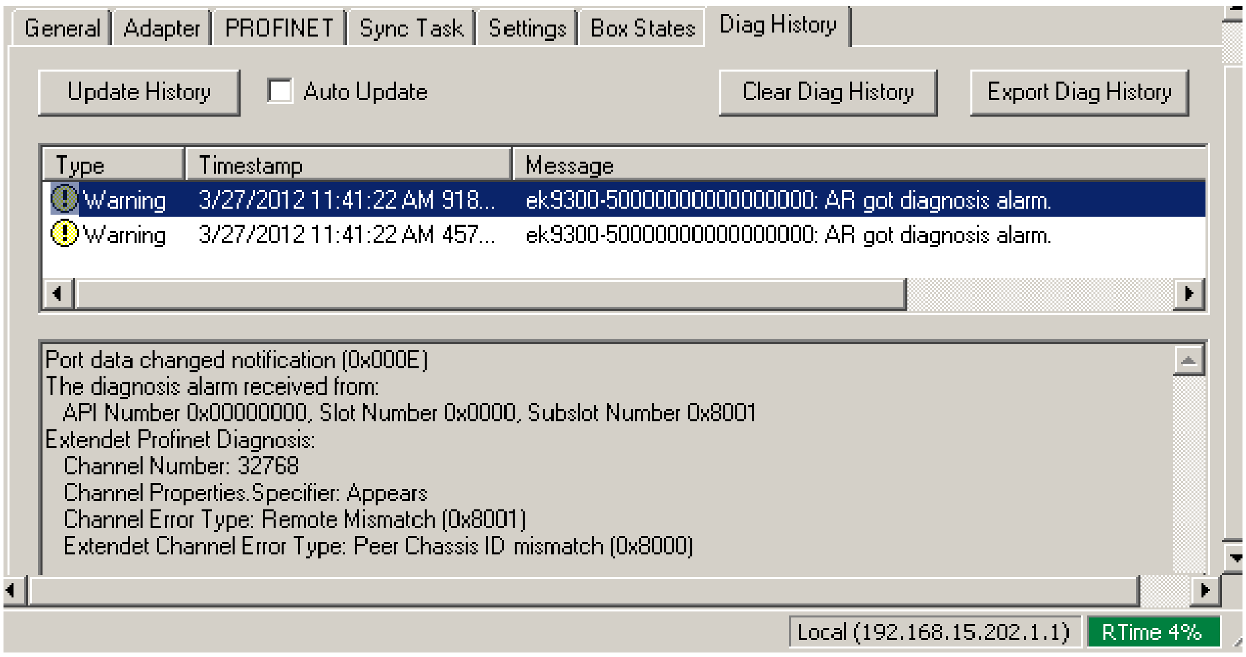
The data structure **ST_PN_Diag** contains a diagnostic message from a terminal that is connected via a PN device and a controller.

```
TYPE str_PortDiag :

STRUCT
    strTimeStamp            : ARRAY[0..7] OF BYTE;
    nAPI                    : DWORD;
    nSlot                   : WORD;
    nSubSlot                : WORD;
    nAlarmType              : WORD;
    nAlarmSpecifier         : WORD;
    nUserStructIdentifier   : WORD;
    nChannelNumber          : WORD;
    nChannelErrorTyp        : WORD;
    nChannelProperties      : WORD;
    nExtChannelErrorTyp     : WORD;
    arSpare                 : ARRAY [1..9] OF WORD;
    arUserSpecificData      : ARRAY [0..19] OF BYTE;
END_STRUCT

END_TYPE
```

The information content of the structure corresponds to that of the Diag History, which is displayed in the system manager.



Type	Timestamp	Message
Warning	3/27/2012 11:41:22 AM 918...	ek9300-50000000000000000: AR got diagnosis alarm.
Warning	3/27/2012 11:41:22 AM 457...	ek9300-50000000000000000: AR got diagnosis alarm.

```
Port data changed notification (0x000E)
The diagnosis alarm received from:
  API Number 0x00000000, Slot Number 0x0000, Subslot Number 0x8001
Extendet Profinet Diagnosis:
  Channel Number: 32768
  Channel Properties.Specifier: Appears
  Channel Error Type: Remote Mismatch (0x8001)
  Extendet Channel Error Type: Peer Chassis ID mismatch (0x8000)
```

Local (192.168.15.202.1.1) RTime 4%

Fig. 65: "Diag History" tab

6.3.3.3 ST_PN_AlarmDiagData

The data structure **ST_PN_AlarmDiagData** contains the alarm diagnosis data record read from a device, including a time stamp that indicates when the event occurred and a flag that indicates that ‘user-specific’ data are present.

```
TYPE ST_PN_AlarmDiagData :

STRUCT
    ST_TimeStamp    : TIMESTRUCT;
    sNameOfStation  : STRING(20);
    ST_Diag         : ST_PN_Diag [▶ 78];
    bUserSpecData   : BOOL;
END_STRUCT

END_TYPE
```

6.3.4 Enumeration types for PROFINET alarms

E_PN_ALARM_TYP

The enumeration type **E_PN_ALARM_TYP** lists all PROFINET communication alarms.

```
TYPE E_PN_ALARM_TYP :
(
    PN_ALARM_RESERVE            :=0,
    PN_ALARM_DIAGNOSE_APPEARS   :=1,
    PN_ALARM_PROCESS            :=2,
    PN_ALARM_PULL               :=3,
    PN_ALARM_PLUG               :=4,
    PN_ALARM_STATUS             :=5,
    PN_ALARM_UPDATE             :=6,
    PN_ALARM_REDUNDANCY         :=7,
    PN_ALARM_Controlled_by_Supervisor           :=8,
    PN_ALARM_Released                           :=9,
    PN_ALARM_Plug_Wrong_Submodule               :=16#A,
    PN_ALARM_Diagnosis_Disappears               :=16#B,
    PN_ALARM_Multicast_Communication_Mismatch   :=16#C,
    PN_ALARM_Multicast                          :=16#D,
    PN_ALARM_STATUS_                            :=16#E,
    PN_ALARM_Sync_                              :=16#F,
    PN_ALARM_Isochronous_Mode_Problem_Notification :=16#10
);
END_TYPE
```

7 Appendix

7.1 EtherCAT AL Status Codes

For detailed information please refer to the EtherCAT system description.

7.2 Firmware compatibility

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

Note

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

Attention

Risk of damage to the device!

Pay attention to the instructions for firmware updates on the separate page [▶ 81]. 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!

EL6631			
Hardware (HW)	**Firmware**	**Revision no.**	**Release date**
02-13*	01 (V00.08)	EL6631-0000-0016	2012/04
		EL6631-0000-0017	2012/10
	02	EL6631-0000-0018	2013/05
	03		2014/12
	04		2015/10
	05*		2017/01

EL6632			
Hardware (HW)	**Firmware**	**Revision no.**	**Release date**
02-07	00 (V00.26)	EL6632-0000-0016	BETA Version
	01	EL6632-0000-0017	2014/11
08-13*	02		2015/10
	03		2016/03
	04*		2017/03

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

7.3 Firmware Update EL/ES/EM/EPxxxx

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

Storage locations

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

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

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

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

Simplified update by bundle firmware

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

- for the firmware to be in a packed format: recognizable by the file name, which also contains the revision number, e.g. 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

Attention

Risk of damage to the device!

Note the following when downloading new device files

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

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

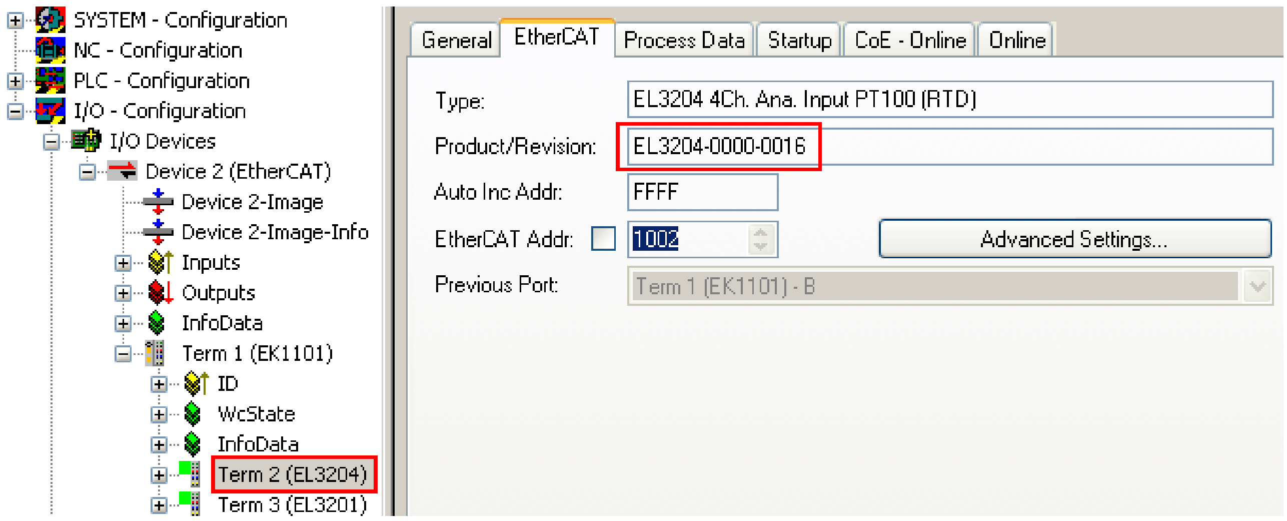
Device description ESI file/XML

Attention

Notice regarding update of the ESI description/EEPROM

Some slaves have stored calibration and configuration data from the production in the EEPROM. These are irretrievably overwritten during an update.

The ESI device description is stored locally on the slave and loaded on start-up. Each device description has a unique identifier consisting of slave name (9 characters/digits) and a revision number (4 digits). Each slave configured in the System Manager shows its identifier in the EtherCAT tab:



Type: EL3204 4Ch. Ana. Input PT100 (RTD)
Product/Revision: EL3204-0000-0016
Auto Inc Addr: FFFF
EtherCAT Addr: 1002
Previous Port: Term 1 (EK1101) - B

Fig. 66: *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.

Note	**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:

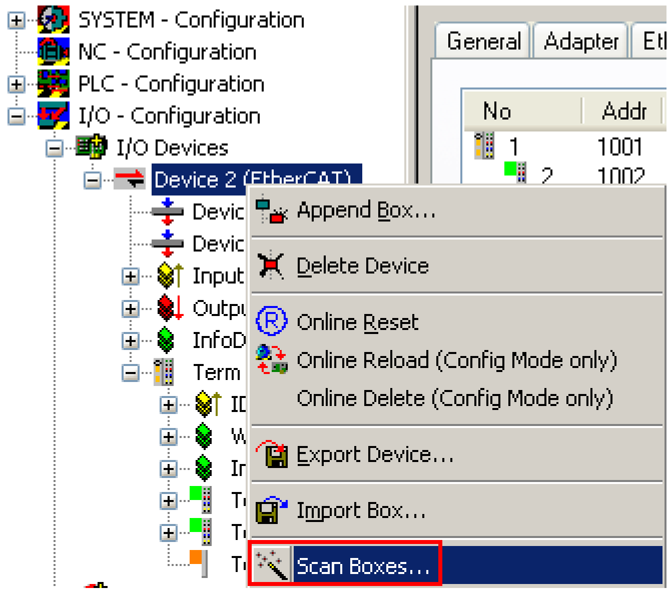
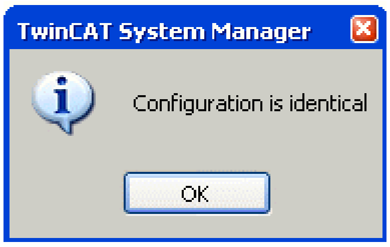


Fig. 67: *Scan the subordinate field by right-clicking on the EtherCAT device in Config/FreeRun mode*

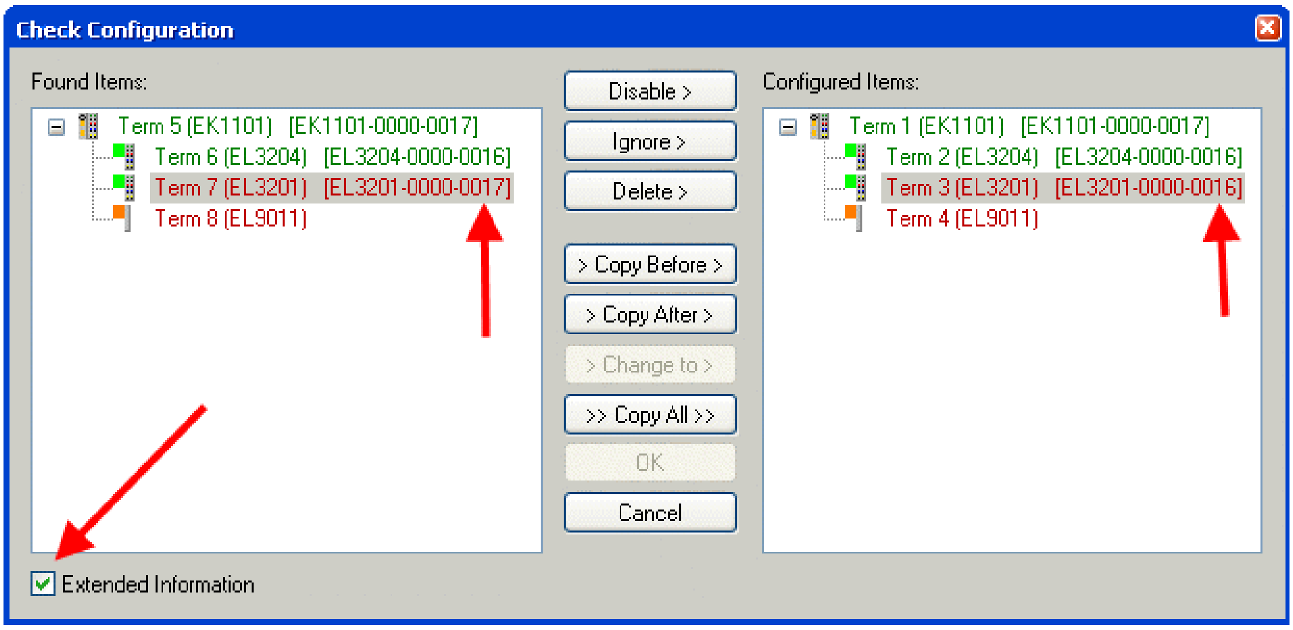
If the found field matches the configured field, the display shows


TwinCAT System Manager

Configuration is identical

Fig. 68: *Configuration is identical*

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


Check Configuration

Found Items:

Term 5 (EK1101) [EK1101-0000-0017]
Term 6 (EL3204) [EL3204-0000-0016]
Term 7 (EL3201) [EL3201-0000-0017]
Term 8 (EL9011)

Configured Items:

Term 1 (EK1101) [EK1101-0000-0017]
Term 2 (EL3204) [EL3204-0000-0016]
Term 3 (EL3201) [EL3201-0000-0016]
Term 4 (EL9011)

Extended Information ✓

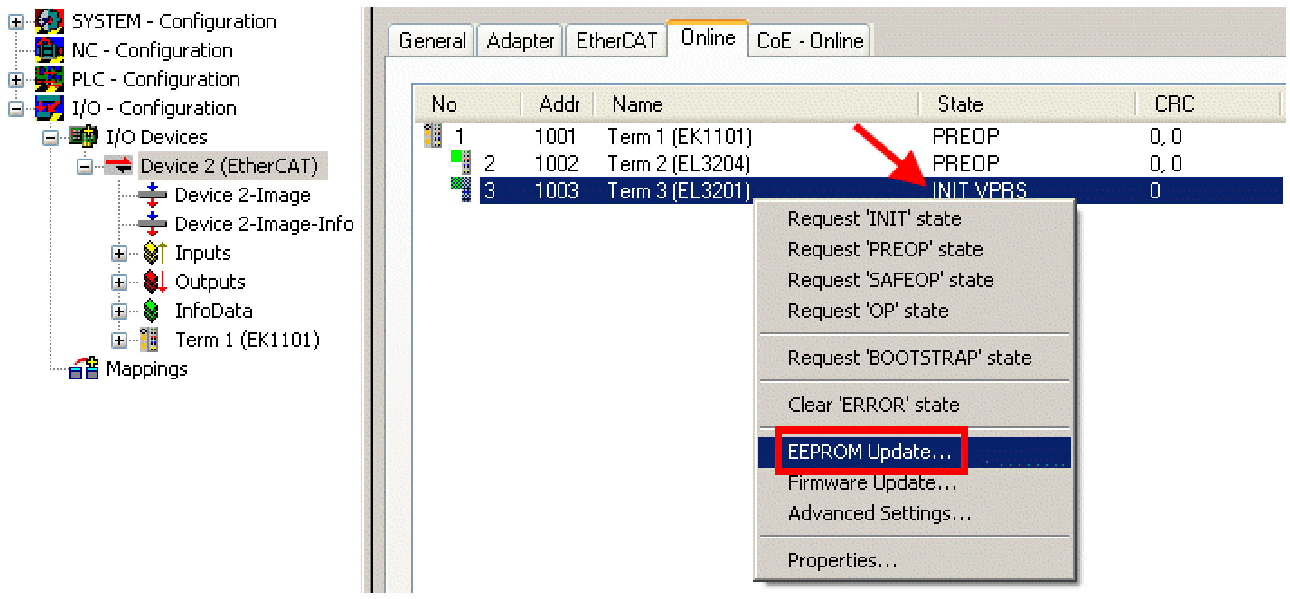
Fig. 69: *Change dialog*

In this example in Fig. *"Change dialog"*, an EL3201-0000-**0017** was found, while an EL3201-0000-**0016** was configured. In this case the configuration can be adapted with the *Copy Before* button. The *Extended Information* checkbox must be set in order to display the revision.

Changing the ESI slave identifier

The ESI/EEPROM identifier can be updated as follows under TwinCAT:

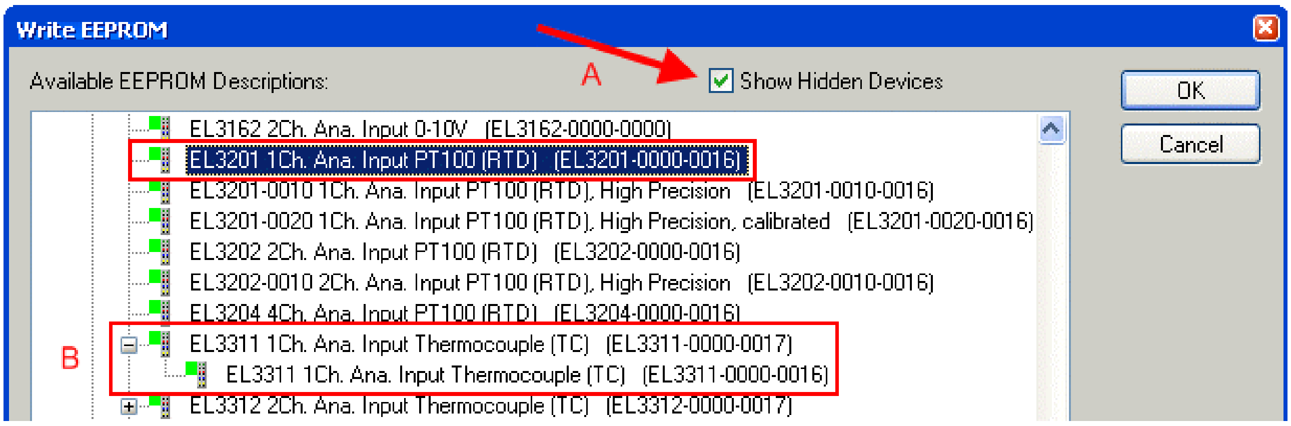
- Trouble-free EtherCAT communication must be established with the slave.
- The state of the slave is irrelevant.
- Right-clicking on the slave in the online display opens the *EEPROM Update* dialog, Fig. *"EEPROM Update"*



No	Addr	Name	State	CRC
1	1001	Term 1 (EK1101)	PREOP	0, 0
2	1002	Term 2 (EL3204)	PREOP	0, 0
3	1003	Term 3 (EL3201)	INIT VPRS	0

Fig. 70: *EEPROM Update*

The new ESI description is selected in the following dialog, see Fig. *"Selecting the new ESI"*. The checkbox *Show Hidden Devices* also displays older, normally hidden versions of a slave.



EL3162 2Ch. Ana. Input 0-10V (EL3162-0000-0000)
EL3201 1Ch. Ana. Input PT100 (RTD) (EL3201-0000-0016)
EL3201-0010 1Ch. Ana. Input PT100 (RTD), High Precision (EL3201-0010-0016)
EL3201-0020 1Ch. Ana. Input PT100 (RTD), High Precision, calibrated (EL3201-0020-0016)
EL3202 2Ch. Ana. Input PT100 (RTD) (EL3202-0000-0016)
EL3202-0010 2Ch. Ana. Input PT100 (RTD), High Precision (EL3202-0010-0016)
EL3204 4Ch. Ana. Input PT100 (RTD) (EL3204-0000-0016)
EL3311 1Ch. Ana. Input Thermocouple (TC) (EL3311-0000-0017)
EL3311 1Ch. Ana. Input Thermocouple (TC) (EL3311-0000-0016)
EL3312 2Ch. Ana. Input Thermocouple (TC) (EL3312-0000-0017)

Fig. 71: *Selecting the new ESI*

A progress bar in the System Manager shows the progress. Data are first written, then verified.

Note	**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.

Determining the firmware version

Determining the version on laser inscription

Beckhoff EtherCAT slaves feature serial numbers applied by laser. The serial number has the following structure: **KK YY FF HH**

KK - week of production (CW, calendar week)
YY - year of production
FF - firmware version
HH - hardware version

Example with ser. no.: 12 10 03 02:

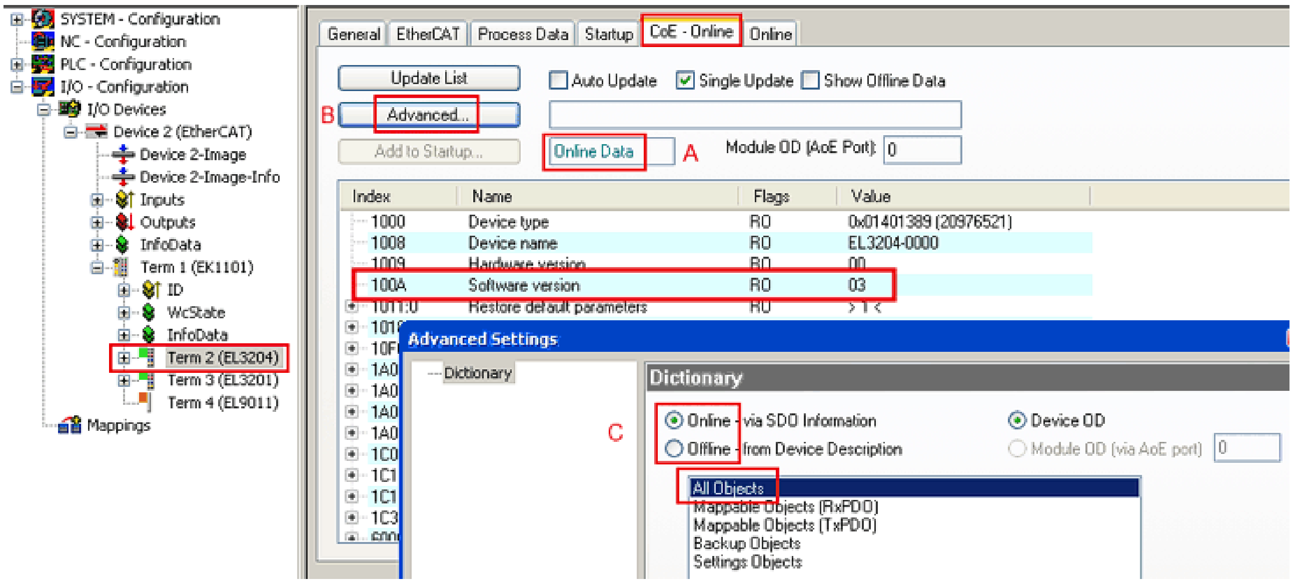
12 - week of production 12
10 - year of production 2010
03 - firmware version 03
02 - hardware version 02

Determining the version via the System Manager

The TwinCAT System Manager shows the version of the controller firmware if the master can access the slave online. Click on the E-Bus Terminal whose controller firmware you want to check (in the example terminal 2 (EL3204)) and select the tab *CoE Online* (CAN over EtherCAT).

Note	**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 does supported it. This CoE directory can only be displayed if a slave is connected and operational. • **offline**: The EtherCAT Slave Information ESI/XML may contain the default content of the CoE. This CoE directory can only be displayed if it is included in the ESI (e.g. "Beckhoff EL5xxx.xml"). The Advanced button must be used for switching between the two views.

In Fig. *"Display of EL3204 firmware version"* the firmware version of the selected EL3204 is shown as 03 in CoE entry 0x100A.



Auto Update [✓ Single Update] Show Offline Data

Module OD (AoE Port): 0

Index	Name	Flags	Value
1000	Device type	RO	0x01401389 (20976521)
1008	Device name	RO	EL3204-0000
1009	Hardware version	RO	00
100A	Software version	RO	03
1011:0	Restore default parameters	RO	> 1 <

Dictionary: [✓ Online - via SDO Information] Offline - from Device Description; [✓ Device OD] Module OD (via AoE port) 0

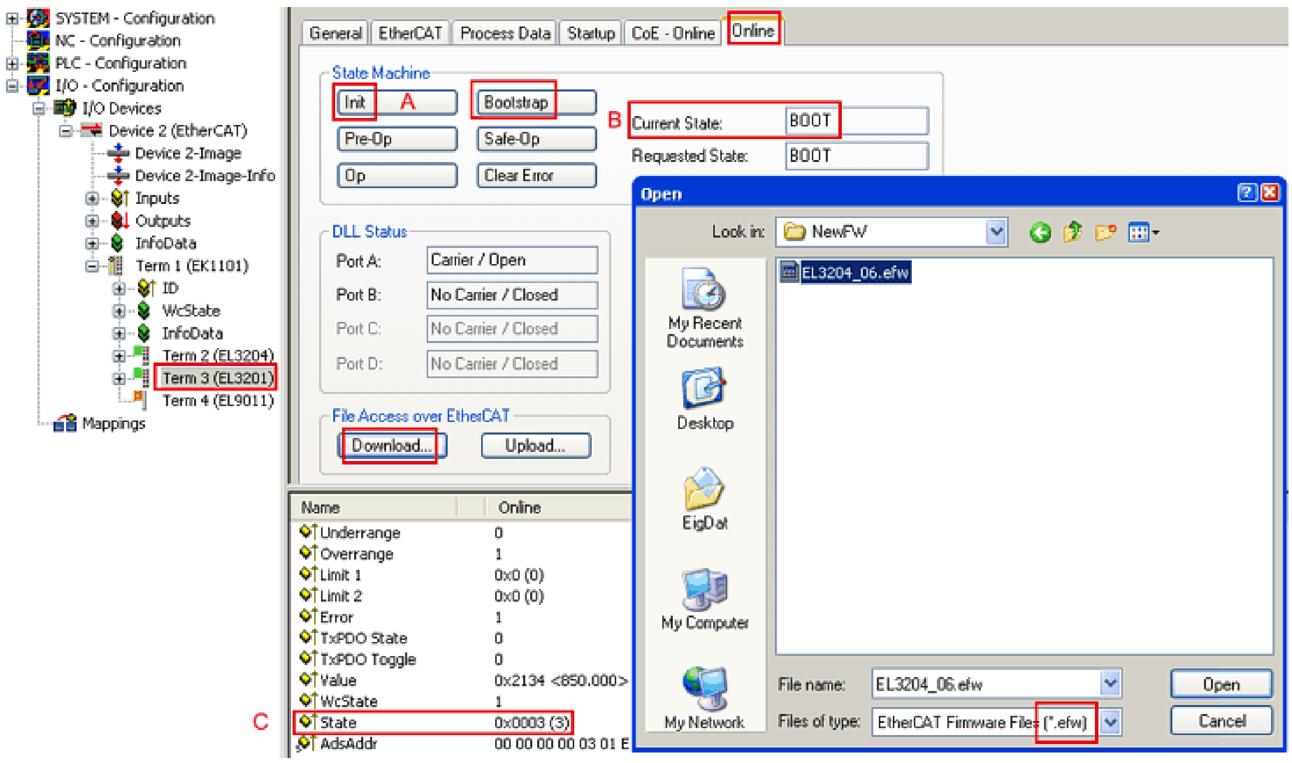
Fig. 72: *Display of EL3204 firmware version*

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*.

Updating controller firmware *.efw

Note	**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".*



Current State: BOOT
Requested State: BOOT

Port A: Carrier / Open
Port B: No Carrier / Closed
Port C: No Carrier / Closed
Port D: No Carrier / Closed

Name	Online
Underrange	0
Overrange	1
Limit 1	0x0 (0)
Limit 2	0x0 (0)
Error	1
TxPDO State	0
TxPDO Toggle	0
Value	0x2134 <850.000>
WcState	1
State	0x0003 (3)
AdsAddr	00 00 00 00 03 01 E

Look in: NewFW
EL3204_06.efw
File name: EL3204_06.efw
Files of type: EtherCAT Firmware Files (*.efw)

Fig. 73: *Firmware Update*

Proceed as follows, unless instructed otherwise by Beckhoff support.

- Switch slave to INIT (A)
- Switch slave to BOOTSTRAP
- Check the current status (B, C)
- Download the new *efw file
- After the download switch to INIT, then OP
- Switch off the slave briefly

FPGA firmware *.rbf

If an FPGA chip deals with the EtherCAT communication an update may be accomplished via an *.rbf file.

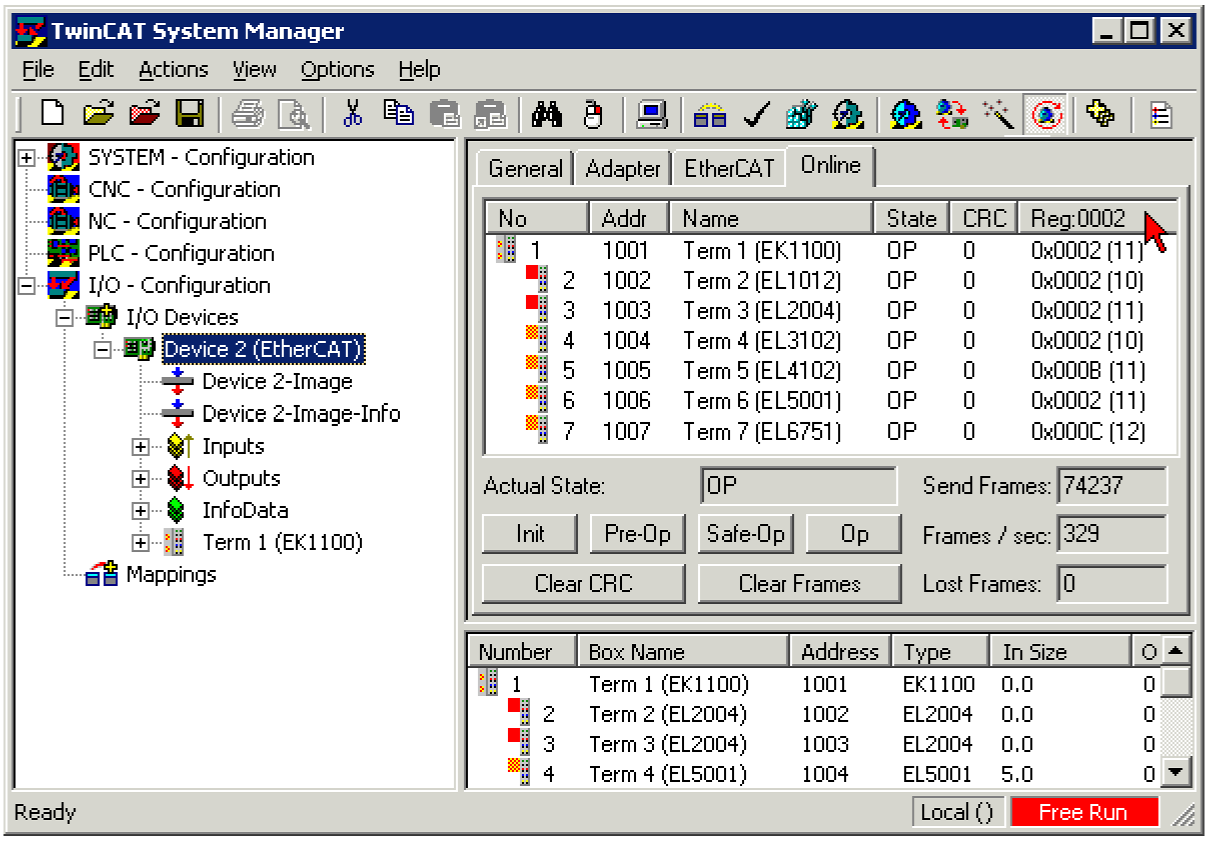
- Controller firmware for processing I/O signals
- FPGA firmware for EtherCAT communication (only for terminals with FPGA)

The firmware version number included in the terminal serial number contains both firmware components. If one of these firmware components is modified this version number is updated.

Determining the version via the System Manager

The TwinCAT System Manager indicates the FPGA firmware version. Click on the Ethernet card of your EtherCAT strand (Device 2 in the example) and select the *Online* tab.

The *Reg:0002* column indicates the firmware version of the individual EtherCAT devices in hexadecimal and decimal representation.



No	Addr	Name	State	CRC	Reg:0002
1	1001	Term 1 (EK1100)	OP	0	0x0002 (11)
2	1002	Term 2 (EL1012)	OP	0	0x0002 (10)
3	1003	Term 3 (EL2004)	OP	0	0x0002 (11)
4	1004	Term 4 (EL3102)	OP	0	0x0002 (10)
5	1005	Term 5 (EL4102)	OP	0	0x000B (11)
6	1006	Term 6 (EL5001)	OP	0	0x0002 (11)
7	1007	Term 7 (EL6751)	OP	0	0x000C (12)

Actual State: OP
Send Frames: 74237
Frames / sec: 329
Lost Frames: 0

Number	Box Name	Address	Type	In Size	O
1	Term 1 (EK1100)	1001	EK1100	0.0	0
2	Term 2 (EL2004)	1002	EL2004	0.0	0
3	Term 3 (EL2004)	1003	EL2004	0.0	0
4	Term 4 (EL5001)	1004	EL5001	5.0	0

Fig. 74: *FPGA firmware version definition*

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

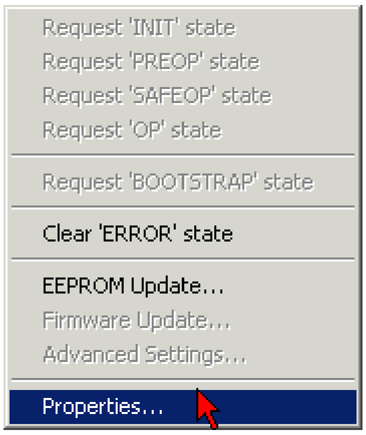


Fig. 75: *Context menu Properties*

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.

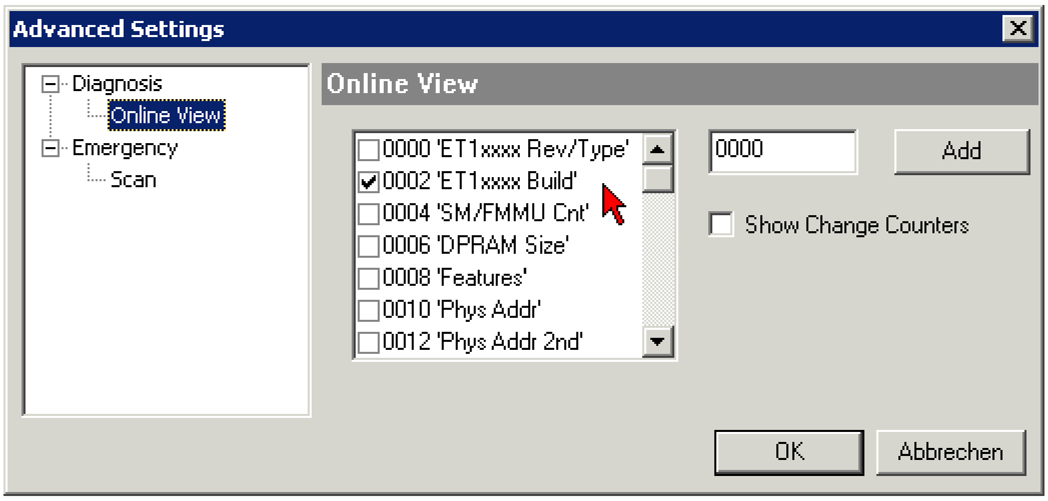


Fig. 76: *Dialog Advanced Settings*

Update

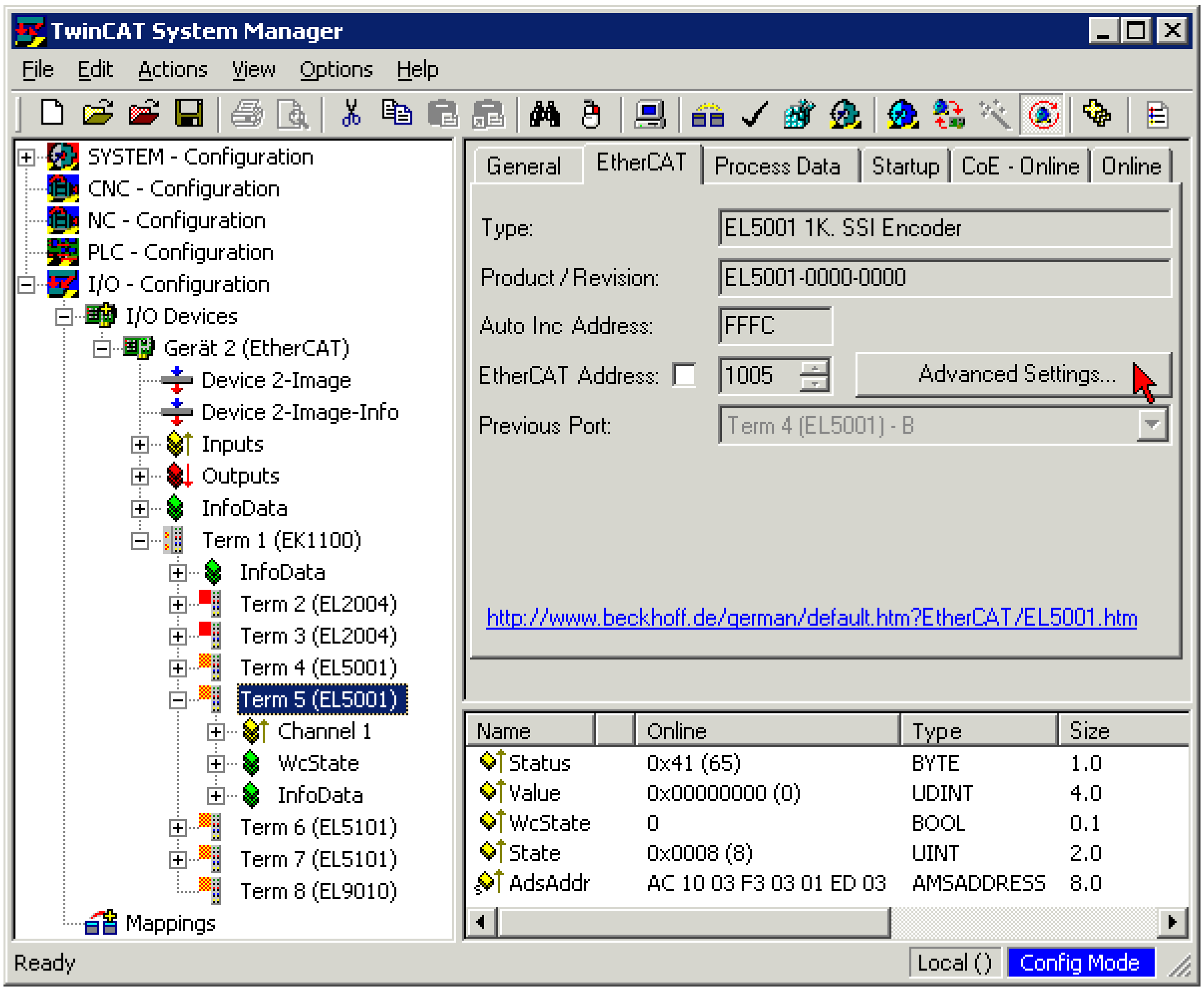
For updating the FPGA firmware

- 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

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.


TwinCAT System Manager
File Edit Actions View Options Help
General EtherCAT Process Data Startup CoE - Online Online
Advanced Settings...
Ready
Local ()
Config Mode

SYSTEM - Configuration
CNC - Configuration
NC - Configuration
PLC - Configuration
I/O - Configuration
- I/O Devices
  - Gerät 2 (EtherCAT)
    - Device 2-Image
    - Device 2-Image-Info
    - Inputs
    - Outputs
    - InfoData
    - Term 1 (EK1100)
      - InfoData
      - Term 2 (EL2004)
      - Term 3 (EL2004)
      - Term 4 (EL5001)
      - Term 5 (EL5001)
        - Channel 1
        - WcState
        - InfoData
      - Term 6 (EL5101)
      - Term 7 (EL5101)
      - Term 8 (EL9010)
- Mappings

Type:	EL5001 1K. SSI Encoder
Product / Revision:	EL5001-0000-0000
Auto Inc Address:	FFFC
EtherCAT Address:	1005
Previous Port:	Term 4 (EL5001) - B

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

Name	Online	Type	Size
Status	0x41 (65)	BYTE	1.0
Value	0x00000000 (0)	UDINT	4.0
WcState	0	BOOL	0.1
State	0x0008 (8)	UINT	2.0
AdsAddr	AC 10 03 F3 03 01 ED 03	AMSADDRESS	8.0

Fig. 77: *Select dialog Advanced Settings*

The *Advanced Settings* dialog appears. Under *ESC Access/E²PROM/***FPGA** click on *Write FPGA* button,

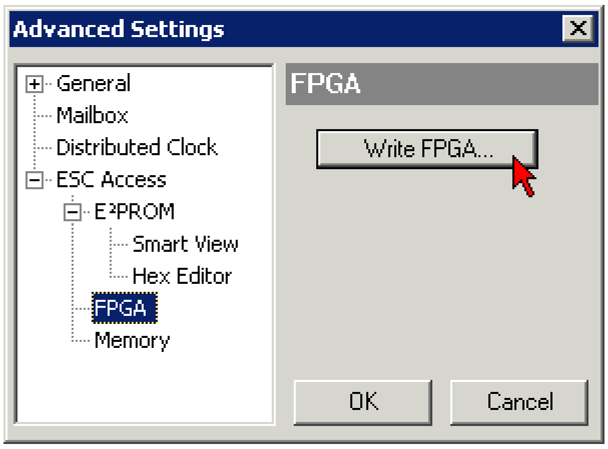
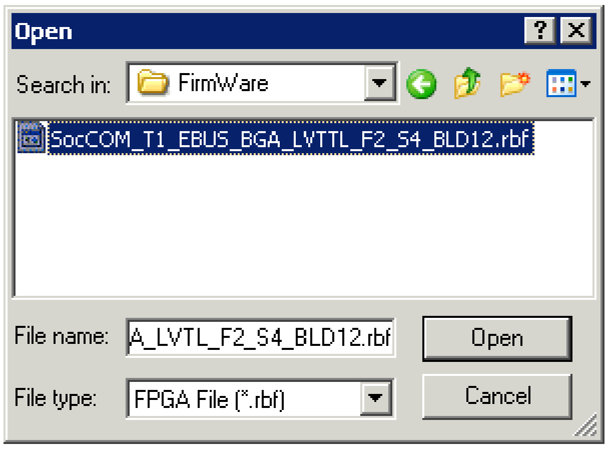


Fig. 78: *Select dialog Write FPGA*


Open

Search in: FirmWare

SocCOM_T1_EBUS_BGA_LVTTL_F2_S4_BLD12.rbf

File name: A_LVTL_F2_S4_BLD12.rbf

File type: FPGA File (*.rbf)

Fig. 79: *Select file*

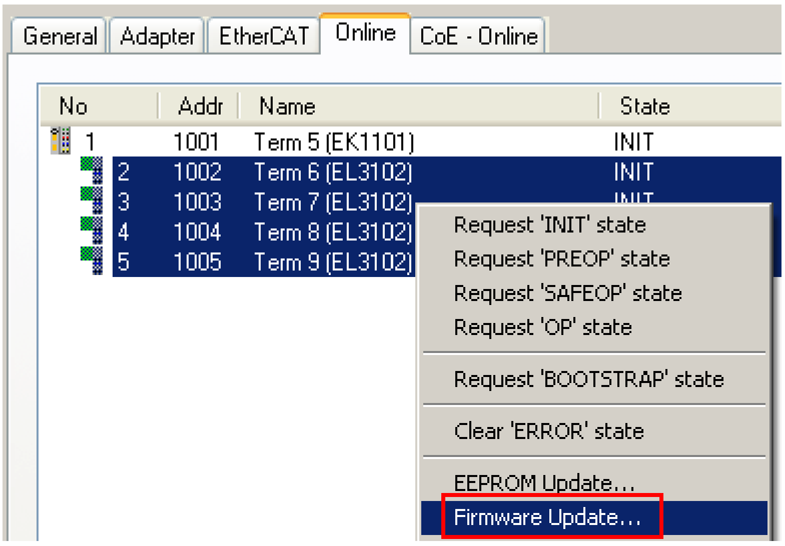
Select the file (*.rbf) with the new FPGA firmware, and transfer it to the EtherCAT device.

Attention	**Risk of damage to the device!** A firmware download to an EtherCAT device must never be interrupted! If this process is cancelled, the supply voltage switched off or the Ethernet connection interrupted, the Ether-CAT device can only be recommissioned by the manufacturer!

In order to activate the new FPGA firmware a restart (switching the power supply off and on again) of the EtherCAT device is required.

Simultaneous updating of several EtherCAT devices

The firmware and ESI descriptions of several devices can be updated simultaneously, provided the devices have the same firmware file/ESI.



No	Addr	Name	State
1	1001	Term 5 (EK1101)	INIT
2	1002	Term 6 (EL3102)	INIT
3	1003	Term 7 (EL3102)	INIT
4	1004	Term 8 (EL3102)	
5	1005	Term 9 (EL3102)	

Fig. 80: *Multiple selection and firmware update*

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

List of illustrations